

Received May 8, 2020, accepted May 21, 2020, date of publication June 1, 2020, date of current version June 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2999036

A Novel Case Base Reasoning and Frequent Pattern Based Decision Support System for Mitigating Software Risk Factors

MUHAMMAD ASIF^{1,2} AND JAMIL AHMED³, (Senior Member, IEEE)

¹Department of Computing and Technology, Abasyn University–Peshawar, Peshawar 25000, Pakistan

²Department of Software Engineering, Foundation University Islamabad–Rawalpindi, Islamabad 44000, Pakistan

³Department of Information Technology, Hazara University Mansehra, Dhodial 21300, Pakistan

Corresponding author: Muhammad Asif (muhammad.asif83@hotmail.com)

ABSTRACT Software risk management is crucial for the success of software project development. The existing literature has models for risk management, but is too complex to be used in practice. The information in the existing studies is scattered over different articles which makes it difficult to find relevant knowledge to establish relationship between risk factors and mitigations. This paper presents a novel model which identifies the relationship between risk factors and mitigations automatically by using intelligent Decision Support System (DSS). The proposed model has four steps. Firstly, the input of the system has been designed where risk factors and mitigations have been inputted into it. Secondly, rule based machine learning approach has been used for mining of associations between risks and mitigations. Thirdly, Case Based Reasoning (CBR) approach has been used to determine the previous cases as rules. Finally, automated rules have been generated to develop an intelligent DSS to mitigate the software risks. The proposed technique copes with the highly cited existing limitations of risk handling like, lack of generic DSS and intelligent relationship between software risks and mitigations. Automated rules have been discovered with a novel idea of CBR and frequent pattern. The proposed model is capable of mitigating upcoming risks in future. Star schema has been implemented to support our proposed DSS. Moreover, from highly cited literature 40 studies were identified from which 26 risk factors, 57 mitigations, 14 questions and 26 automated rules have been extracted. According to the validation of IT industry experts, the average of the effectiveness of DSS is 51-55%. The novelty of the proposed research is that it uses two state of the art methods (Rule Based Machine Learning and CBR) to identify software risk mitigations. The results of the proposed model show that the chances of risks in software development have been reduced significantly.

INDEX TERMS Case based reasoning, decision support system, machine learning, rule based system, software risks, and software mitigations.

I. INTRODUCTION

Software projects fail due to various different types of risk factors, namely certain, uncertain, dependent and independent [1], [2]. Risk factors force project team to compromise on the software project scope, objectives, planning, scheduling, budget and execution. Numerous factors affect the whole software development process, such as poor decision of project managers, inappropriate hiring of employees by a human resource department, etc. [1].

Software risk management is very crucial for the success of project development [3]–[5]. The main problem for the software industry is the risks which cannot be ignored during

The associate editor coordinating the review of this manuscript and approving it for publication was Justin Zhang.

the software development phases [6]. Software projects are very complex by nature and therefore, liable to failure due to this reason. However, many researchers have made efforts to minimize the failure of the software projects. In 1995, United State of America (USA) spends around \$250B on 175,000 projects, but a good of them were failed or did not achieve the required results. One of the major reasons for these failures was the wrong assessment and mismanagement by the project managers [7]. There are many other statistics, where software projects are reported to fail due wrong decision at different level of the project development. It can be observed from the current literature that there is no formal DSS available that can be used in software development to support software developers to avoid such kind of failures. Incorporating intelligence DSS to the process

becomes imperative. Therefore, this research proposes an intelligent system to bridge a gap between project manager's decisions and software risks. This DSS will work on the basis of four steps. First, Domain Experts or literature supplies their knowledge in the shape of software risk factors, along with mitigations into the Knowledge Base (KB) [8], [21]. In this regard an exploratory survey has also been included in this research [24]. Second, Rule Based Machine Learning approach known as the Association Rule Learning is used for mining associations. Eclat algorithm is used to find associations between these risks and their mitigations in the form of frequent patterns and rules. Third, CBR approach is used to determine the previous cases in the form of rules [9], [10]. Conclusively, the intelligent decision net is established for risk factors and risk mitigation.

The scope of this research work is, risk factors and risk mitigations of Risk Mitigation, Risk Monitoring and Risk Management (RMMM) plan i.e. Risk Factors \rightarrow Risk Mitigation.

This paper is divided into various sections. Section II presents the contribution. Section III discusses the review of related work. Section IV describes the proposed design of DSS. Section V shows the prototype development. Section VI explains the validation of the proposed model. The conclusion is presented in the final section VII.

II. CONTRIBUTION

The contributions of this research work are as follows:

1. The proposed technique copes with the highly cited existing limitations of risk handling like, lack of generic DSS and intelligent relationship between software risk factors and mitigations [1], [2]–[10], [19], [25]–[29], [36], [37].
2. The intelligent relationship between risks and mitigations, in the form of rules, has been discovered by using a novel idea of CBR and frequent pattern base [8], [9].
3. The proposed model is capable of mitigating upcoming risks in the future, based on knowledge discovery which also learns the new cases [37], [25].
4. A total no. of 26 highly cited risk factors and 57 risk mitigations have been identified from the existing literature and our proposed DSS mitigate these risk factors [8], [24].
5. A star schema structure has been implemented to support our proposed DSS for knowledge discovery.

III. RELATED WORK

This section discusses the latest literature on risk factors, risk mitigations and related issues.

In [11], a Knowledge Management (KM) in software engineering has been discussed. This research paper describes the levels of knowledge, i.e. Data (raw facts and figures), Information (processed form of data), Knowledge (Meta Information) and the importance of KM in software engineering. This paper focuses on KM to make better decisions, decrease time and cost, increase quality of the product and prevent

human errors. It also reveals that KM is a prevention and mitigation strategy for risks. It represents the flow of information through Experience Factory Organization (EFO). It has three phases. a) Package experience by analysis, evaluating and synthesizing raw facts and model building. b) Making an experience base or KB of data, models and experience. c) Identifying and using the relevant experience of previous projects for the current project.

In [12], a DSS for software project management introduces the hybrid approach to the problem because not a single process model is best suitable for all processes. It combines three different models, i.e. analytical model, discrete model and continuous model. This hybrid structure provides both qualitative and quantitative suggestions for better results of software processes. But this research work has been unable to propose a single generic DSS model.

In [13], a risk assessment of software projects using fuzzy inference system introduces the fuzzy rule based system. Approximately 17 million fuzzy rules have been introduced for better comparison of different projects or risk mitigations, but instead of creating the whole rule based or KB, this research work uses a heuristic approach for inference.

In [6], Software Engineering Body of Knowledge (SWEBOK) explains the risk management guidelines. As per a Delphi study [14], 53 risk factors, e.g., project size, team size, scope creeps, etc. from managers were collected from Hong Kong, Finland and United States. This study adopted three phases, i.e. brain storming, narrowing down and ranking of risks. Software risk management methodologies, discussed in the technical report of the Software Engineering Institute to manage the risks during development and risk mitigation, can be achieved through the risk mitigation strategy and risk planning in [15].

In [16], a risk-based decision support has been based on space programs with risks. The main focus of this research is reduction of technical, cost and schedule risks. Total risk architecture has been developed to improve the quality of decision making for mitigating risks. This risk architecture collects risk drivers into project risks, supportability risks and mission risks with decisions to accept initialization the feedback loop. If risks are not acceptable, then it reconsiders the re-plan of mitigation, avoidance strategies and control paradigm. However, if risks are acceptable, then it evaluates the fault tree and processes the risks.

In [17], a goal-oriented framework of risk analysis has been proposed. It has proposed the optimal solution in terms of multiple objects, for example, risk factors, cost of treatment, rewards on goals, etc. This model or framework helps in risk analysis and reduction of chances of risks. A three layer approach has been followed in this work. 1) Asset Layer. 2) Event Layer and 3) Treatment layer.

In [18], a risk oriented model to assess strategic decisions on new product development projects has been proposed. This work makes use of a decision tree for handling the risks, such as, increasing the budget and delays in project delivery.

In [19], a conceptual framework for Knowledge Based Risk Management and processes has been proposed. It proposed elements to build a Knowledge Base Risk Management (KBRM) framework for Information Technology (IT) projects. It also suggests the merger of knowledge management and risk management process for the improvement of Risk Response Planning (RRP).

In [20], the major focus is on the phases of risk identifications and risk assessment. It follows three steps of flow: a) risk identification, b) risk assessment and c) risk treatment or controlling. This research conducted a case study. And the extensively explained through this case study.

In [25], risk management in Distributed Software Development (DSD) has been presented. A DSS has been developed for practitioners in order to assess risks and choose suitable control strategies. For the construction of a knowledge base, a systematic literature review has been conducted.

In [26], risks and opportunities of software-as-a-service have been explored from a survey of IT executives. They have introduced a research model on the basis of the opportunity - risk framework. Data from a survey of 349 IT executives at various German companies has been collected and analyzed. Some prominent factors have been found such as security threats, cost, performance, economic risks, etc.

In [27], Wallace's work has focused on five risk factors to measure software risks by using fuzzy logic. The factors are Team, Planning, Complexity, Requirements and User. This work proposes a framework for the risk assessment and management. The total no. of 243 fuzzy rules have been generated using this framework.

In [28], a study has been conducted with automated search for risks and risk mitigations in global software development. They have extracted 85 risks and 77 risk mitigations and then, categorized them into four parts, i.e. software development, outsourcing rationale, project management and human resource.

In [19], an empirical study has been conducted for the software development risk management. A total number of 145 software projects have been investigated and a model for software risk management has been proposed. The final survey conducted in this study has 78 questions.

In [29], problems and solutions of global software development and collaboration have been discussed. The important barriers are cultural, temporal, geographic and linguistic distances. These barriers are overcome by building knowledge sharing infrastructure, synchronous communication and frequent site visits.

In [30], a tertiary study has been conducted. The main objective is to conduct a Systematic Literature Review (SLR) in distributed software development. Out of fourteen SLRs, three are related to software design, software engineering education and requirements. Four topics are related to the engineering process. Seven SLRs are related to management of distributed development. The main aim of this study is to identify the challenges of distributed software development and finding their solutions.

In [31], the use of Global Software Engineering (GSE) jargon has been investigated. For identification of the problem, a Delphi-inspired study has been conducted with ten researchers in global software engineering. They developed an empirically based glossary for the important concepts in global software engineering. Then they developed a taxonomy for global software engineering based on generalization-specialization relationships. It is used to map and categorize the existing knowledge.

In [32], a study of the use of agile practices in global software engineering has been conducted. This study focuses on ten years' research papers from 1999-2009. Then, these papers have been classified into research type and contribution.

In [33], a systematic literature review has been conducted to identify challenges and solutions in distributed software development projects. An evidence-based distributed software development model has been proposed in it. A total no. of 54 related works have been studied. These were published from 1998 to 2009.

In [34], a global software engineering knowledge management approach for intensive risk mitigation has been proposed. The four step approach is focused, i.e. a knowledge management, satisfiability of goal, requirement and maintenance of the requirement. An agile management system using knowledge management was discussed in this work. They have proposed architecture based and algorithmic development based approach for the prediction of software risks and their mitigation. The articles of [25], [35]–[37] have also discussed about decision support system in risk Assessment, risk management, challenges and risk analytics respectively.

After review of the related work, it has been concluded that there should be a compact rule based DSS for creating a link between software risk factors and mitigations using RBS and CBR with data mining techniques.

IV. PROPOSED DECISION SUPPORT SYSTEM ENGINE DESIGN

A DSS Engine algorithm and mathematical proof are proposed given below respectively. Domain Experts enter software risk factors and mitigations into the KB. It then searches for existing rules stored in the KB. If rule(s) are found, then DSS engine executes these rule(s) and an association among risk factors and risk mitigations is generated. Otherwise, a) It prioritizes the risk factors in relation with risk mitigations which are already stored in the KB. b) It makes new relations of the entered risk factors with risk mitigations. These relations are created through input from the domain experts as well as from the existing literature. c) An association rule mining technique, Eclat algorithm is used to find frequent patterns and then CBR approach is used to determine new or updated rules. It also matches with previous cases of existing rules. d) These relationships are then added as rules in the KB. e) The new rules extracted from the KB are then executed.

f) It then generates the intelligent risk mitigation decision rules.

In CBR, a new case of risk factor and its mitigation comes as a problem statement. Case Base contains learned cases or solved cases. If a case exists, then it is used as a proposed solution in the form of a solved case. Otherwise, this rule will be revised and tested to be retained in Case of risks and mitigations in Step I. Generation of automated rules in step II. In step III a star schema for data warehouse. Eclat algorithm is implemented in step IV. And CBR in step V. The terms used in the below mathematical algorithm are as under.

X = Risk factors and mitigations in the form of knowledge from the domain expert

Y = Rule(s) in the form of risk factors with mitigation from knowledge base

A = Rule(s) with frequent patterns

B = Risk mitigation decision

C = Risk factors and store in working memory

D = relationship of risk factors with risk mitigation

E = The relationships in the form of rules into the knowledge base

F = Frequent patterns in the knowledge base

G = New case in the form of risk factors with mitigations

H = Case(s) from learned cases in the form of rule(s)

I = Reuse the rule(s)

J = Test Rule and Store in Learned Cases (s)

KB = Knowledge Base

Step 1: BEGIN

Step 2: Input 'X'

Step 3: Select 'Y'

Step 4: IF rule(s) exist in 'KB' **THEN**

Apply 'A' \wedge Generate 'B'

GO TO STEP 5

ELSE

Prioritize 'C' \wedge Create 'D' \wedge Enter 'E' \wedge Mining 'F'

After mining take 'G' \wedge Retrieve 'H' \wedge Reuse 'I'

IF solved case **THEN**

Apply 'A' \wedge Generate 'B'

GO TO STEP 5

ELSE IF new rule **THEN**

Test 'J' \wedge Apply 'A' \wedge Generate 'B'

GO TO STEP 5

ELSE update the rule

Test 'J' \wedge Apply 'A' \wedge Generate 'B'

GO TO STEP 5

END IF

END IF

END IF

Step 5: END

The mathematical proof of the proposed DSS engine design is given below.

Given Data:

RF = Risk Factors

RM = Risk Mitigations

RKB = Rules in Knowledge Base

KDE = Knowledge with Domain Expert

KL = Knowledge in Literature

Hypothesis:

$x \in (RKB) \vee (KDE) \vee (KL)$

Where $x = RKB$ or $x = KL$, $x \in KDE$

Proof By Contradiction:

Suppose that $\neg (x \in ((RKB) \vee (KDE) \vee (KL)))$

Then $x \notin ((RKB) \wedge (KDE) \wedge (KL))$

$x \notin RKB$ ($x \notin RKB$)

$x \notin KL$ ($x \notin KL$)

But $x \subseteq KDE$ as provided in hypothesis

$x \in KDE$

This is a contradiction. $x \in (RKB) \vee (KDE) \vee (KL)$ **Q.E.D**

A. STEP I: IDENTIFICATION OF RISK FACTORS AND MITIGATIONS

A survey was conducted to get information about risks and mitigations from different users, including software practitioners and developers. On the basis of their feedback, ratings have been assigned to risk factors in terms of percentage. A total number of 20 software risk factors and 50 risk mitigations have been identified which is mentioned below. The survey has 20 software risk factors and 50 risk mitigations mentioned below. These software risk factors and mitigations have been extracted from [8], [24]. To enhance the research work, the researcher also collected 6 risk factors and 7 mitigations from [21]. So, there are a total number of 26 software risk factors and 57 risk mitigations in the survey. The survey questions are also available in Appendix 'A'. All of these risk factors and risk mitigations have been listed below in Table 1, 2, 3 and 4, along with their brief descriptions.

Risk factors of software from [8], [24] in Table 1:

Risk factors of software from [21] in Table 2:

Risk mitigations software from [8], [24] in Table 3:

Risk mitigations of software from [21] in Table 4:

B. STEP II: IDENTIFICATION OF RULES IN THE FORM OF INTELLIGENT RELATIONSHIP OF RISK FACTORS AND MITIGATIONS

First 20 rules have been identified from [1], [3] and [8], whereas the rules from 21 to 26 are from [21] in this research. These rules act as risk control techniques against software risks. Some of the rules have also been updated from [21], such as, M6 and M45 have been added in Rule 3, M53 has been added in Rule 5, M54 has been added in Rule 8 and M34 has been added in Rule 12. These new additions of mitigations have been formatted as bold, shown below. So, the KB has been improved from the review of literature of [21]. The rules are the conjunctions of risk mitigations. Risk mitigations against risk factors are entered as input from the domain engineers. Software risk factors are the antecedent part of the rules and Software risk mitigations are the consequent part of the rules. The arrow mentioned below is the implication between antecedent and consequent.

Software Risk Factors (TID) \rightarrow Risk Mitigation (Item set)

TABLE 1. Software risk factors.

Sr	Risk factors	Description
1	IMPROPER FEASIBILITY REPORT (IF) 90%	The feasibility study includes proper budget, schedule, deadlines, etc. An improper feasibility study lead to project failure due to bad decisions and deadlines.
2	HIGHER MANAGEMENT DECISIONS (HMD) 88%	A project's success or failure always depends on the decisions of project's management. An incompetent personnel take bad decisions
3	UNDERSTANDING PROBLEMS OF CUSTOMERS (UPOC) 85%	Customers are basically laymen. They are not aware of the latest technologies due to which they are unable to understand the developer's viewpoint
4	UNDERSTANDING PROBLEM OF DEVELOPERS (UPOD) 82%	In most of the occasions, the software developer doesn't understand the customer's requirements due to differences in perception levels of the developer and the customer
5	IMPROPER PLANNING (IP) 80%	A project's success is based on strong planning and decisions. Successful projects demand a comprehensive planning; otherwise, the project may fail
6	IMPROPER SCOPE DEFINITION (ISD) 78%	The scope of a project is basically defining the goals and objectives of that project
7	LACK OF EXPERIENCE OF PROJECT MANAGER (LOEPM) 76%	The project manager must be experienced in terms of technicality and decision making. An inexperienced person may prove to be disastrous
8	GOVERNMENT FACTORS (GF) 75%	The government decisions play a vital role in providing confidence to national and international investors. All the projects may get affected due to bad governance
9	IMPLEMENTATION (IMP) 72%	A skillful human resource is required for the development of a software project. Lack of technical skill definitely leads to failure
10	CULTURAL DIVERSITY (CDR) 70%	The coordination among the members of a team should be very strong and no differences of opinion, religion, culture, gender, etc. among team members should be reflected
11	LACK OF MOTIVATION (LOM) 68%	Motivated teams can lead to high quality results, e.g., appraisals and bonuses
12	PERSONNEL HIRING (PH) 67%	The hiring of employees is a critical task for the employers.

(Rule 1) SOTP → {M49, M48, M28, M13, M12, M1, M30}

TABLE 1. Software risk factors.

Sr	Risk factors	Description
13	UNREALISTIC DEADLINES (UD) 60%	Lack of skilled employees is risky for any organization The project should meet its proper deadlines. Unrealistic timeline for deliverables, milestones and project completion causes a project's failure
14	INAPPROPRIATE DESIGN (ID) 55%	Software design is the second phase of the Software Development Life Cycle (SDLC). Inappropriate design is unacceptable for the success of the project
15	IMPROPER BUDGET (IB) 52%	The project should be completed within the allocated budget. Over budgeting can cause project failure
16	INAPPROPRIATE TECHNOLOGY (IT) 49%	The selection of tool and technology for the development of a project is very crucial. A project cannot succeed without the selection of proper technology
17	LACK OF RESOURCES (LOR) 44%	The project team must have required resources for the project, i.e. software, hardware, human, technical resources, etc.
18	SIZE OF THE PROJECT (SOTP) 40%	There should be a certain balance between the resources and the size of the project. A project with fewer resources may not gain any success
19	IMPROPER MARKETING TECHNIQUES (IMT) 35%	Low quality marketing of good projects can make them unsuccessful
20	MARKET DEMAND OBSOLETE (MDO) 30%	The project must be in compliance with the industry demand or else it would not fulfill the user's expectations

(Rule 2) LOR → {M6, M29, M47}
 (Rule 3) HMD → {M4, M17, M5, M35, M36, M6, M45}
 (Rule 4) MDO → {M32, M33, M49, M43}
 (Rule 5) UD → {M45, M25, M44, M16, M2, M43, M24, M23, M53}
 (Rule 6) PH → {M10, M29, M17, M22, M21, M20, M6}
 (Rule 7) ID → {M25, M26, M27, M21, M22}
 (Rule 8) ISD → {M32, M23, M14, M1, M13, M12, M54}
 (Rule 9) GF → {M33, M39, M40}
 (Rule 10) UPOD → {M37, M10, M9, M8, M7, M6}
 (Rule 11) IMP → {M41, M35, M9, M3, M2, M16, M1}
 (Rule 12) IP → {M10, M11, M3, M1, M12, M34}
 (Rule 13) CDR → {M17, M14, M10, M42}
 (Rule 14) UPOC → {M22, M21, M20, M31, M17, M5}

TABLE 2. Software risk factors.

Sr	Risk factors	Description
1	LOW KEY USER INVOLVEMENT (LKUI) 89%	The clients of the required software product must be involved during the development phase; otherwise it will be a total mess
2	INSUFFICIENT/INAPPROPRIATE STAFFING (IIS) 84%	Less and non-technical staff members are directly related to the project failure
3	LACK OF AN EFFECTIVE SOFTWARE PROJECT MANAGEMENT METHODOLOGY (LOESPM) 74%	There should be compact software project methodologies available, such as, Agile, Scrum, etc.
4	CHANGE IN ORGANIZATIONAL MANAGEMENT DURING THE SOFTWARE PROJECT (COMSP) 64%	A major change in organizational management during the development stage, e.g., replacement of a team member, is harmful to the success of the project
5	INEFFECTIVE COMMUNICATION SOFTWARE PROJECT SYSTEM (ICSPS) 54%	Strong and effective communication among the employees, clients and departments has a pivotal role in a complete software project system
6	ABSENCE OF A HISTORICAL DATA (AOHD) 48%	It is difficult to progress in the absence of an authentic historical data or template

- (Rule 15) IB → {M4, M28, M2, M3, M53}
- (Rule 16) IF → {M12, M28, M34, M3, M2, M1}
- (Rule 17) LOEPM → {M38, M5, M23, M11, M10, M4, M15, M14}
- (Rule 18) LOM → {M39, M19, M8, M7, M6, M18}
- (Rule 19) IMT → {M31, M18, M50, M26, M27}
- (Rule 20) IT → {M28, M46}
- (Rule 21) LKUI → {M34, M51, M52 }
- (Rule 22) ISS → {M34, M3, M24, M25, M47, M30}
- (Rule 23) LOESPM → {M10, M55, M56}
- (Rule 24) COMSP → {M57 }
- (Rule 25) ICSPS → {M34, M5, M13, M45 }
- (Rule 26) AOHD → {M53 }

C. STEP III: IDENTIFICATION OF STAR SCHEMA

To have a clear picture, a database has been maintained in the form of simple star schema which has one fact table and four dimension tables. Fact table has surrogate keys and dimension tables have surrogate primary keys. The four dimension tables are: Risk Factors Dimension, Risk Mitigation Dimension, Risk Type Dimension and Risk Priority Dimension. A single Fact Table Risks has five keys. Risk_Avoidance is

TABLE 3. Software risk mitigations.

Sr	Risk mitigations	Description
1	Clear idea of the requirements (M1)	User’s requirements must be crystal clear. There should be no ambiguity requirements in the project
2	Proper feasibility report making (M2)	There should be a sound and a clear feasibility study report. It includes proper decisions that lead towards a successful project
3	Requirements specification (M3)	There should be a complete Software Requirement Specification (SRS), which includes all the requirements of the project
4	IT consultants (M4)	An IT consultant is the entity who advises the organization about proper implementation, cost estimating, management, deployment, administration, etc.
5	Proper communication channel (M5)	Communication between the client and developers is an important factor for the success of the project, e.g., regular meetings
6	Retaining and preservation of good employees (M6)	Skillful employees are the assets of any organization. Such employees should be retained by the companies
7	Bonuses (M7)	Bonuses, in terms of attractive packages and extra salaries, must be given to employees
8	Attractive packages (M8)	Handsome salary leads to employee satisfaction and motivation
9	Developer’s faithfulness (M9)	Employees faithful towards their work can be an asset to the organization. There should be a relation of confidence and trust among employees and project managers
10	Proper team structure (M10)	The software development team should be properly managed and centralized
11	Proper backup plan (M11)	There should be a proper backup plan during the analysis phase of the SDLC
12	Define goals and objectives (M12)	Goals cannot be achieved without objectives. These are the desired results, a project manager wants to achieve
13	Ensure communications and milestones (M13)	Deliverables and milestones should be taken care of during the development phases. Communication among the stakeholders of the project is very successful guide
14	Leadership (M14)	A good creative leader is the key to success
15	Past experience (M15)	An experienced person is an asset to the organization and a successful guide

TABLE 3. Software risk mitigations.

Sr	Risk mitigations	Description
16	Proper use of methodologies and software process models (M16)	Software process models should be adopted using latest methods and technologies
17	Work unit culture (M17)	Team work is the key to success. No project can be successful without team work
18	On the job and off the job training (M18)	Training of the employees is very important. On the job training should be given to employees during the working hours and off the job training should be given away from the working organization
19	Respect and honour of employees (M19)	The respect, honour and dignity are the rights of any worker in the organization. It leads to motivation of employees in the organization
20	Employee attitude (M20)	The positive and constructive attitude towards work is the best tool for success
21	Employee skill (M21)	The main asset of any organization is the individual skill level of the employees
22	Employee awareness (M22)	Employees must know about the policies of the organization. In this regard, employee awareness training programs should be organized
23	Continuous review (M23)	Verification and validation in terms of technical reviews are helpful for a project to be successful
24	Project scheduling (M24)	The software development team should formally schedule the project using Activity Network diagram and bar chart diagram
25	Prototyping (M25)	It is the initial version of an under developed project. It should meet the requirements of the client
26	User involvement (M26)	The main stakeholder of the project is the user. The user should be involved during the requirements gathering and testing phase
27	Use statistical methods (M27)	It is mandatory to use mathematical methods and models for the solution of the problem
28	Choice of technology after thorough research of available tools and technologies (M28)	The project manager should select latest technology and tool for the development of the project
29	Human resource role (M29)	An existence of proper human resource department is very important in the organization so that it may hire skilled employees

TABLE 3. Software risk mitigations.

Sr	Risk mitigations	Description
30	Proper testing techniques (M30)	The developers and quality assurance members should use proper testing techniques, i.e. Unit testing, integration testing, black box, white box, gray box, stress testing, performance testing, etc.
31	Proper sales marketing team (M31)	A strong marketing campaign is required for the product to be sold. A good professional sales team is important in this regard
32	Identification of success criteria (M32)	It is important to identify a criterion that leads to success
33	Policy setting and enforcement (M33)	A crystal clear policy statement of the organization should be announced and enforced by the higher management
34	Scrutable requirements (M34)	User's requirements are the scrutable requirements
35	Top management commitment (M35)	The higher authority should be committed to perform well in every situation
36	Facilitated Application Specification Technique (M36)	It is a formal team, which has both the client and developer representatives. The members of this team work together for the accomplishment of the assigned task
37	Centralization (M37)	The project team should be centralized
38	Intuitive and creative (M38)	Creative and innovative ideas are the key to existence in the software industry
39	Positive behaviour and problem solving skills (M39)	The positive intentions and attitude enable an individual to solve any kind of problem
40	Security checklist and authentication process (M40)	An authentication process in the project is the main entity. Checklist for security is mandatory, such as, a login page, encrypted passwords, two pass securities, etc.
41	Set key performance indicator (M41)	A key performance indicator (KPI) is a criterion for measurement of efficiency and performance of the organization
42	Stress testing (M42)	It is concerned with a system under heavy stress to find errors
43	Regular updates (M43)	New versions or regular updates of the software should be properly managed
44	Assess past communications (M44)	It is a good practice to assess previously developed software and customer communication of the software should be properly managed

TABLE 3. Software risk mitigations.

Sr	Risk mitigations	Description
45	Contingency plan (M45)	It is a plan for handling predictive and upcoming problem
46	Troubleshooting (M46)	It is the identification and fixation of errors in the software project
47	Reusability (M47)	It is the best approach to use already developed components for the development of the software
48	Project tracking and control (M48)	A project must be tracked and controlled from the start to end
49	Impact assessment (M49)	It is defined as the impact of risk on the software
50	Consistent commitment (M50)	The key to success is the consistent commitment of all the team members of the project

TABLE 4. Software risk mitigations.

Sr	Risk mitigations	Description
1	Including formal and periodic risk assessment (M51)	A proper risk identification and assessment is necessary for the mitigation of risks
2	Maintain proper documentation of each individual's work (M52)	Each and every task of the project should be properly documented
3	Developing and adhering a software project plan (M53)	A project plan during the analysis phase is the backbone of software development
4	Reusable user documents (M54)	Properly documented requirements are essential for the success of a project
5	Dividing the software project into controllable portions (M55)	It is simply a divide and conquer rule that divides the tasks into sub-tasks, which makes them easy to handle
6	Ensuring the quality factor deliverables and task analysis (M56)	The requirements should be analyzed and the quality product should be delivered to the client
7	Implement/ utilize benchmarking and tools of technical analysis (M57)	It is very important to use authentic source for the technical analysis

the fact column. Risk Avoidance for a particular factor, for a particular risk mitigation, for a particular factor associated with mitigations based on Risk Type and Risk severity. The values of risk avoidance column show the relationships of risk factors with mitigations. There are two types of risks, i.e. Dependent and Independent risks. Risk severity has three values, i.e. High, Normal and Low. This Fact Table basically contains the above mentioned 26 rules. This data warehouse is

useful for composing data into one corporate database. After this, data mining extracts eloquent data from that common database and a star schema, making Dimensions and Fact tables are also presented in Table 5.

D. STEP IV: DRY RUN OF éclat ALGORITHM

Dry Run of Eclat algorithm: Three step approach [22], [23].

STEP 1) Calculate minimum support, the candidate and frequent item sets generations.

By taking 10% minimum support(S) for Eclat, we get the answer 03 after applying the formula. Total items in step I of section IV is 26. As a result, only those items are selected in the frequent items which have minimum 03 occurrences in step I of section IV.

$$10/100 * 26 = 03. \tag{1}$$

On the basis of 10% minimum support, the candidates and frequent items are generated. Candidate generation is a move where frequent subsets are stretched (one item at a time) and candidates are tested with the actual data. In the first step, 56 candidates are generated from M1 to M50. Total 19 frequent items are selected on the basis of 10% minimum support as M1:5 M2:4 M3:5 M4:3 M5:4 M6:5 M10:6 M12:4 M13:3 M14:3 M17:4 M21:3 M22:3 M23:3 M25:3 M28:4 M34:4 M45:3 M53:3. After this, Frequent1, Frequent 2 and Frequent 3-itemsets have been generated for finding frequent itemset.

Frequent 1-itemsets in Vertical Format

Item sets	TID
M1	IMP, ISD, IP, IF, SOTP
M2	UD, IMP, IB, IF
M3	IMP, IP, IB, IF, ISS
M4	HMD, IB, LOEPM
M5	HMD, UPOC, LOEPM, ICSPS
M6	LOR, PH, UPOD, LOM, HMD
M10	PH, UPOD, IP, CDR, LOEPM, LOESPM
M12	SOTP, ISD, IP, IF
M13	SOTP, ISD, ICSPS
M14	ISD, CDR, LOEPM
M17	HMD, PH, CDR, UPOC
M21	PH, ID, UPOC
M22	PH, ID, UPOC
M23	UD, ISD, LOEPM
M25	UD, ISS
M28	SOTP, IB, IF, IT
M34	IP, IF, LKUI, ISS, ICSPS
M45	HMD, ICSPS, UD
M53	UD, IB, AOHD

Frequent 2-itemsets in Vertical Format

Item sets	TID
(M1 M3)	IMP, IP, IF
(M1 M12)	SOTP, ISD, IP, IF
(M2 M3)	IMP, IB, IF
(M3 M34)	IP, IF, ISS
(M21 M22)	PH, ID, UPOC

TABLE 5. Dimensions and fact data.

DmSRFactors		
Factor_id	Factor_name	Factor_desc
F1	SOTP	Size of the project
F2	LOR	Lack of resources
.....
n		

DmRMitigation		
Rmitigation_id	RMitigate_name	RMitigation_desc
RM30	M30	Proper Testing Techniques
RM1	M1	Clear-Idea-of-the-Requirements
RM12	M12	Define-Goals & Objectives
RM13	M13	Ensure Communications and Milestones
RM28	M28	Choice-of- technology-after- thorough-research-of-available-tools- and-technologies
RM48	M48	Project Tracking and Control
RM49	M49	Impact Assessment
RM6	M6	Retaining-and-preservation-of- Good-Employees
RM29	M29	Human Resource Role
RM47	M47	Reusability
.....
n		

DmRTypes		
Risk_type_id	Risk_type_name	Risk_type_desc
RT1	Drisk	Dependent Risk
RT2	IDRisk	Independent Risk
.....
n		

DmRPriority	
RP_id	R_severity
RP1	High
RP2	Normal
RP3	Low
.....
n	

DmRMitigation				
Rmitigation_id	Factor Id	Risk type Id	RP id	Risk Avoidance
RM30	F1	RT1	RP1	F1>>RM30
RM1	F1	RT1	RP1	F1>>RM1
RM12	F1	RT1	RP1	F1>>RM12
RM13	F1	RT1	RP1	F1>>RM13
RM28	F1	RT1	RP1	F1>>RM28
RM48	F1	RT1	RP1	F1>>RM48
RM49	F1	RT1	RP1	F1>>RM49
RM6	F2	RT1	RP1	F2>>RM6
RM29	F2	RT1	RP1	F2>>RM29
RM47	F2	RT1	RP1	F2>>RM47
.....
n				

Frequent 3-itemsets in Vertical Format

Item sets	TID
(M1 M3 M12)	SOTP, IP, IF

So, the Frequent Items = (M1 M3 M12). Where {M1} is the Clear Idea of the Requirements, {M3} is Requirements Specification and {M12} is Define Goals and Objectives.

Subsets of L = {M1}, {M3}, {M12}, {M1 M3}, {M1 M12}, {M3 M12}

Association Rules:

- 1) (M1 M3) => M12.....Rule A
- 2) (M1 M12) => M3.....Rule B
- 3) (M3 M12) => M1.....Rule C
- 4) M12 => (M1 M3)Rule D
- 5) M3 => (M1 M12)Rule E
- 6) M1 => (M3 M12)Rule F

STEP 2) Rules Generation with minimum confidence

By taking 50% Confidence (C),

$$C = (A => B) = P(B|A) = S(AUB)/S(A). \quad (2)$$

- 1) For Rule 'A', C = (2/3)*100 = 66.6% Selected Rule
- 2) For Rule 'B', C = (2/4)*100 = 50% Selected Rule
- 3) For Rule 'C', C = (2/2)*100 = 100% Selected Rule
- 4) For Rule 'D', C = (2/4)*100 = 50% Selected Rule
- 5) For Rule 'E', C = (2/5)*100 = 40% Not Selected Rule
- 6) For Rule 'F', C = (2/5)*100 = 40% Not Selected Rule

In this case we can also select strong rules, which satisfy both the Support and Confidence, i.e. Rule A, Rule B, Rule C and Rule D.

STEP 3) Measuring Correlation Analysis

Lift is a correlation measure,

If >1, A and B are positively correlated

If <1, A and B are negatively correlated

If =1, A and B are independent

$$\text{Lift}(A,B) = P(AUB)/P(A)P(B) \quad (3)$$

- 1) For Rule 'A', Lift = (2/3*4) = 0.1666 Negatively Correlated
- 2) For Rule 'B', Lift = (2/4*5) = 0.1 Negatively Correlated
- 3) For Rule 'C', Lift = (2/2*5) = 0.2 Negatively Correlated
- 4) For Rule 'D', Lift = (2/4*3) = 0.1666 Negatively Correlated
- 5) For Rule 'E', Lift = (2/5*4) = 0. Negatively Correlated
- 6) For Rule 'F', Lift = (2/5*2) = 0.2 Negatively Correlated

E. STEP V: IDENTIFICATION OF NEW RULES

For CBR, six new cases in the form of risk factors and seven new cases in the form of risk mitigations have been generated. These are: LKUI, ISS, LOESPM, COMSP, ICSPS, AOHD and M51, M52, M53, M54, M55, M56, M57 respectively.

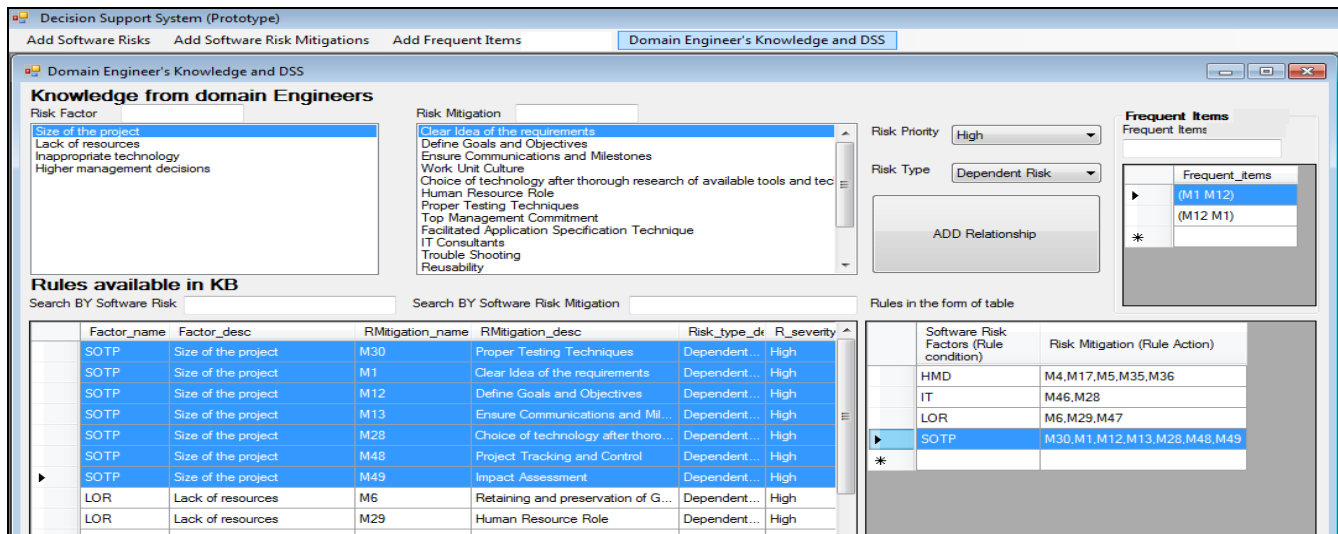


FIGURE 1. DSS prototype.

Five rules have been revised as per CBR and placed under revised cases category. Hence, these are the learned cases stored in the KB. New mitigations have been added into the rules and mentioned in bold. The KB has been increased by adding six new rules as well as the revised five rules.

- HMD = {M4 M17 M5 M35 M36 M6 M45}
- ISD = {M32 M23 M14 M1 M13 M12 M54}
- IP = {M10 M11 M3 M1 M12 M34}
- IB = {M4 M28 M2 M3 M53}
- UD = {M45 M25 M44 M16 M2 M43 M24 M23 M53}

V. PROTOTYPE DEVELOPMENT

The DSS prototype in Fig. 1 is developed in the C# language by using Microsoft Visual Studio and SQL server. Software risks factors and Software risk mitigations can be added by Domain Engineers such as Factor Id, Factor Name, Factor Description, Mitigation Id, Mitigation Name and Mitigation Description. By clicking ADD button, they can be entered into the Knowledge Base. Then, the DSS shows rules and frequent patterns in the prototype software. For example, the size of the project is a risk factor and can be mitigated through clear idea of the requirements from customers, by using proper and exhaustive testing techniques, goals and objectives must be defined, etc. The frequent patterns of risk mitigations are also generated using Eclat algorithm automatically, i.e. Clear-Idea-of-the-requirements (M1), Define-Goals-and-Objectives (M12). You can also add relationships by using risk Dependency and severity of risk from the drop-down lists. Searching option is also available in the prototype for the convenience of users. You can search by Risk Factors or Risk Mitigations.

VI. VALIDATION

This DSS tool and its rules have been validated by three experts of different IT companies based in Islamabad Pakistan. Some of the past projects have been checked

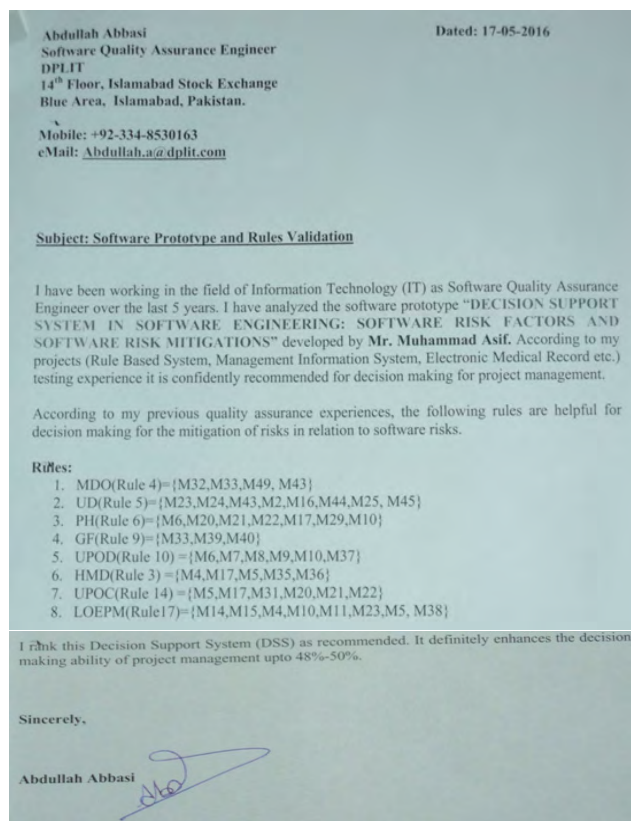


FIGURE 2. Expert-1 validation.

through this tool and satisfactory results have been gained. Rules have also been validated and then suggested right factors and their mitigations. The average percentage of effectiveness of DSS is 51-55%. Each expert has highlighted his past project's experience in the figures 2, 3 and 4. They have faced different development risks in them. During validation, they have calculated the overall issues occurred in the previous projects and found some rules by using the

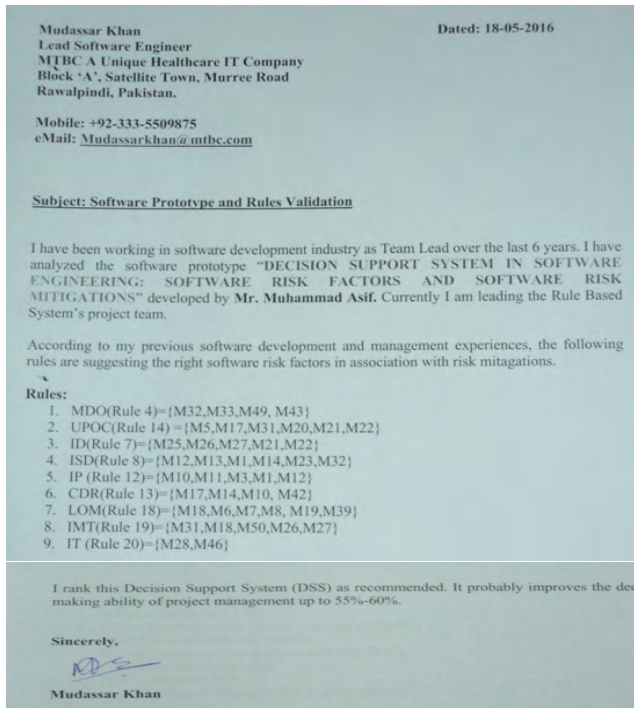


FIGURE 3. Expert-2 validation.

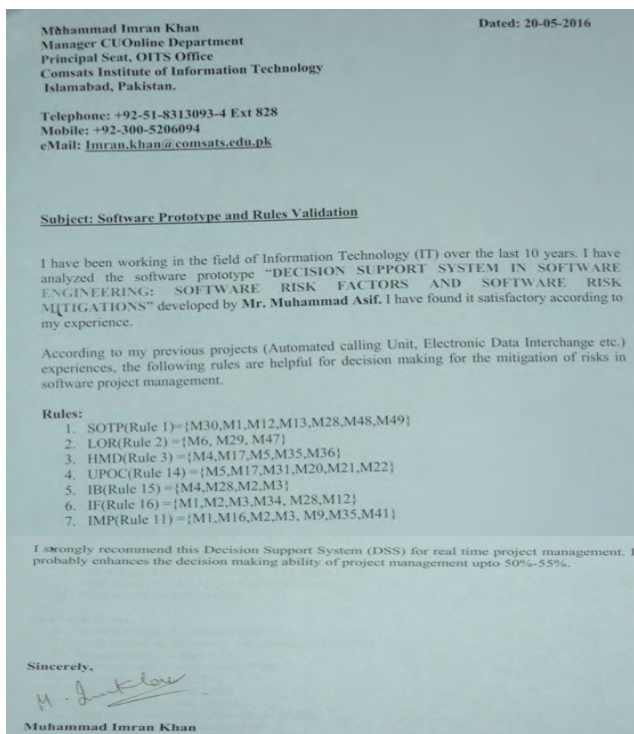


FIGURE 4. Expert-3 validation.

proposed prototype. These rules are helpful to mitigate risks in those projects. So they have calculated the percentage of effectiveness by the percentage formula.

$$\text{Effectiveness\%} = \left(\frac{\text{Estimated resolved issues/}}{\text{Estimated issues occurred}} \right) * 100 \quad (4)$$

$$\text{Average Effectiveness\%} = (\text{Expert1\%} + \text{Expert2\%} + \text{Expert3\%}) / 3 \quad (5)$$

A Prospective validation was performed for the decision rules as these are individually evaluated on the basis of past experience of the individuals. The evaluation of prospective case study was carried out in two phases. The first phase involved the participation of the three expert individuals from three different companies. In the second phase, the tool has been used by the same experts. First expert, Mr. Abdullah Abbasi (Software Quality Assurance Engineer, DPLIT) tested the rules on the past tested projects, including management information system and electronic medical record. He then recommended the rules 4, 5, 6, 9, 10, 3, 14 and 17 with the effectiveness of 48-50%. Second expert, Mr. Mudassar Khan, MTBC) tested the rules in the past tested project rule based system. He then recommended the rules 4, 7, 8, 12, 13, 14, 18, 19 and 20 with the effectiveness of 55-60%. Third expert, Mr. Muhammad Imran (Manager CU Online department, COMSATS) tested the rules on the past tested projects, including automated calling unit and Electronic data interchange. He then recommended the rules 1, 2, 3, 11, 14, 15 and 16 with the effectiveness of 50-55%.

So, the effectiveness of these rules can reduce the risk factors of software development failures. The screenshots of the validation letters from the three experts are given below in Fig. 2, Fig. 3 and Fig. 4 respectively.

VII. CONCLUSION

In this study, an intelligent relationship between risk factors and mitigations, in the form of rules, has been presented by using a novel idea of CBR and frequent pattern base. Risk factors and mitigations are the key parameters of this new study. The present study has a number of advantages. Firstly, the new way of dealing with DSS is based upon the highly cited limitations of risk handling like, lack of generic DSS and intelligent relationship of risks and mitigations. Secondly, it is dealing with the individual software risk and mitigation. The KB will increase with time in the form of relationships between risks and mitigations. This means that the proposed model is capable of mitigating future occurring risks based on knowledge discovery. Thirdly, this study presents an idea of a knowledge discovery based DSS, which has a compact KB of the link between risks and mitigations. Fourthly, Rule based machine learning is conceptually a type of RBS, and it is a technique of Artificial Intelligence. So, a total no. of 26 rules in the form of 26 risk factors and 57 mitigations has been identified in this study. Eclat algorithm has some advantages: a) It uses depth-first search technique, which reduces memory requirement. b) It is faster than Apriori algorithm. c) There is no need to scan database each time. d) It follows the vertical data format. On the other hand, Apriori and Fp-Tree algorithms are using horizontal data format.

The contributions of this research work are: (1) The dataset has been extended, as six new risk factors and seven risk mitigations have been introduced along with the previously

Risks	Rank (From 1 to 26)
<ol style="list-style-type: none"> 1. Improper feasibility report (IF) 2. Higher management decisions (HMD) 3. Understanding problems of customers (UPOC) 4. Understanding problem of developers (UPOD) 5. Improper planning (IP) 6. Improper–scope definition (ISD) 7. Lack of experience of project manager (LOEPM) 8. Government factors (GF) 9. Implementation (IMP) 10. Cultural diversity (CDR) 11. Lack of motivation (LOM) 12. Personnel hiring (PH) 13. Unrealistic deadlines (UD) 14. Inappropriate design (ID) 15. Improper budget (IB) 16. Inappropriate technology (IT) 17. Lack of resources (LOR) 18. Size of the project (SOTP) 19. Improper marketing techniques (IMT) 20. Market demand obsolete (MDO) 21. Low key user involvement (LKUI) 22. Insufficient/ Inappropriate staffing (IIS) 23. Lack of an effective software project management methodology (LOESPM) 24. Change in organizational management during the software project (COMSP) 25. Ineffective communication software project system (ICSPTS) 26. Absence of a historical data (AOHD) 	

existing data. (2) Rule based machine learning approach (association rule mining) has been used i.e., Eclat algorithm. (3) Case Based Reasoning (CBR) technique has been used in the work for finding new and previous cases. It provides better reasons for decision making. (4) A star schema for data warehouse has been proposed for data mining. (5) A prototype has been developed and prospective validation has been performed.

During this research, there were some threats to the validity of the proposed approach. First, Whether or not the case base reasoning and frequent pattern base technique lead to a successful decision support system. Second, how this work will be validated to prove the effectiveness of the proposed model. Both the threats are overcome successfully. The section IV proposed DSS engine design and section VI validation, proves the validity of the proposed approach.

The future studies need to focus on risk monitoring and risk management of RMMM plan.

APPENDIX

- Q1: How is your experience in the software industry?
- Q2: Does improper feasibility report affect the software development?
 - a) High b) Medium c) Low
- Q3: What is the level of higher management decisions in your organization?
 - a) High b) Medium c) Low
- Q4: What is your opinion about the understanding abilities of developers in response of customer’s requirements?
- Q5: What is your opinion about the impact of improper planning and scope definition on the overall project budget?
- Q6: How much an experienced project manager is required for a successful project?
- Q7: What is the required skill level of employees for the implementation stage of the development?
 - a) High b) Medium c) Low
- Q8: Are the tools/ techniques compatible with the industry demand?
 - a) High b) Medium c) Low

Risk Factors	Link (Mitigations)
<ol style="list-style-type: none"> 1. Improper feasibility report (IF) 2. Higher management decisions (HMD) 3. Understanding problems of customers (UPOC) 4. Understanding problem of developers (UPOD) 5. Improper planning (IP) 6. Improper–scope definition (ISD) 7. Lack of experience of project manager (LOEPM) 8. Government factors (GF) 9. Implementation (IMP) 10. Cultural diversity (CDR) 11. Lack of motivation (LOM) 12. Personnel hiring (PH) 13. Unrealistic deadlines (UD) 14. Inappropriate design (ID) 15. Improper budget (IB) 16. Inappropriate technology (IT) 17. Lack of resources (LOR) 18. Size of the project (SOTP) 19. Improper marketing techniques (IMT) 20. Market demand obsolete (MDO) 21. Low key user involvement (LKUI) 22. Insufficient/ Inappropriate staffing (IIS) 23. Lack of an effective software project management methodology (LOESPM) 24. Change in organizational management during the software project (COMSP) 25. Ineffective communication software project system (ICSPS) 26. Absence of a historical data (AOHD) 	
<p>Risk Mitigations</p> <ol style="list-style-type: none"> 1. Clear Idea of the requirements(M1) 2. Proper Feasibility Report making (M2) 3. Requirements Specification(M3) 4. IT Consultants (M4) 5. Proper Communication Channel (M5) 6. Retaining and preservation Good Employees (M6) 7. Bonuses (M7) 8. Attractive packages (M8) 9. Developers Faithfulness (M9) 10. Proper Team Structure (M10) 11. Proper backup plan (M11) 12. Define Goals and Objectives (M12) 13. Ensure Communications and Milestones (M13) 14. Leadership (M14) 15. Past Experience (M15) 16. Proper methods and process Models (M16) 17. Work Unit Culture (M17) 18. On the job and off the job training(M18) 19. Respect and Honour of Employees (M19) 20. Employee Attitude (M20) 21. Employee Skill (M21) 22. Employee Awareness (M22) 23. Continuous Review (M23) 24. Project Scheduling (M24) 25. Prototyping (M25) 26. User Involvement (M26) 27. Use Statistical Methods (M27) 28. Choice of technology after thorough research of available tools technologies (M28) 	<ol style="list-style-type: none"> 29. Human Resource Role (M29) 30. Proper Testing Techniques (M30) 31. Proper Sales Marketing Team (M31) 32. Identification of Success Criteria (M32) 33. Policy Setting and Enforcement (M33) 34. Scrub able requirements (M34) 35. Top Management Commitment (M35) 36. Facilitated Application Specification Technique (M36) 37. Centralization (M37) 38. Intuitive and Creative (M38) 39. Positive behaviour problem solving skills (M39) 40. Security Checklist Authentication (M40) 41. Set Key Performance Indicator (M41) 42. Stress testing (M42) 43. Regular Updates (M43) 44. Assess Past Communications (M44) 45. Contingency Plan (M45) 46. Trouble Shooting (M46) 47. Reusability (M47) 48. Project Tracking and Control (M48) 49. Impact Assessment (M49) 50. Consistent Commitment (M50) 51. Including formal and periodic risk assessment (M51) 52. Maintain proper documentation of each individual's work (M52) 53. Developing and adhering a software project plan (M53) 54. Reusable user documents early (M54) 55. Dividing the software project into controllable portions (M55) 56. Ensuring that quality-factor deliverables and task analysis (M56) 57. Implement/ utilize benchmarking and tools of technical analysis (M57)

Q9: How can the motivation level be increased among the employees?

Q10: What is the trust level among the employees of different cultures?

a) High b) Medium c) Low

Q11: How do you compare unrealistic deadlines, insufficient resources and size of the project during the development?

Q12: Are government factors also affecting the software project development? If yes then please provide your opinion.

Q13: Rank the Software risk factors according to their severity.

Q14: Please link the following risk mitigations with the risk factors, according to your experience.

REFERENCES

- [1] M. Uzzafer, "A risk classification scheme for software projects," *Int. J. Softw. Eng. Appl.*, vol. 7, no. 1, pp. 57–68, Jan. 2013.
- [2] M. Keil, P. Cule, K. Lyytinen, and R. Schmidt, "A framework for identifying software project risks," *Commun. ACM*, vol. 41, no. 11, Nov. 1998, pp. 76–82.
- [3] B. W. Boehm, "Software risk management: Principles and practices," *IEEE Softw.*, vol. 8, no. 1, pp. 32–41, Jan. 1991.
- [4] P. L. Bannerman, "Risk and risk management in software projects: A reassessment," *J. Syst. Softw.*, vol. 81, no. 12, pp. 2118–2133, Dec. 2008.
- [5] F. Ruiz, A. Vizcaíno, M. Piattini, and F. García, "An ontology for the management of software maintenance projects," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 14, no. 3, pp. 323–349, Jun. 2004.
- [6] A. Abran, W. Moore, P. Bourque, and R. Dupuis, *Guide to the Software Engineering Body of Knowledge (SWEBOK)*. Los Alamitos, CA, USA: IEEE Press, Dec. 2004.
- [7] H.-W. Jung, S.-G. Kim, and C.-S. Chung, "Measuring software product quality: A survey of ISO/IEC 9126," *IEEE Softw.*, vol. 21, no. 5, pp. 88–92, Sep. 2004.
- [8] M. Asif and J. Ahmed, "Analysis of effectiveness of Apriori and frequent pattern tree algorithm in software engineering data mining," in *Proc. 6th Int. Conf. Intell. Syst., Modeling Simulation*, Feb. 2015, pp. 28–33.
- [9] B. U. Haque, R. A. Belecheanu, R. J. Barson, and K. S. Pawar, "Towards the application of case based reasoning to decision-making in concurrent product development (concurrent engineering)," *Knowl.-Based Syst.*, vol. 13, nos. 2–3, pp. 101–112, Apr. 2000.
- [10] R. Belecheanu, K. S. Pawar, R. J. Barson, B. Bredehorst, and F. Weber, "The application of case based reasoning to decision support in new product development," *Integr. Manuf. Syst.*, vol. 14, no. 1, pp. 36–45, Feb. 2003.
- [11] I. Rus and M. Lindvall, "Knowledge management in software engineering," *IEEE Softw. Mag.*, vol. 19, no. 3, pp. 26–38, May 2002.
- [12] P. Donzelli, "Decision support system for software project management," *IEEE Softw.*, vol. 23, no. 4, pp. 67–75, Jul. 2006.
- [13] S. H. Iranmanesh and S. B. Khodadadi, "Risk assessment of software projects using fuzzy inference system," in *Proc. IEEE Comput. Ind. Eng.*, Jul. 2009, pp. 1149–1154.
- [14] R. Schmidt, K. Lyytinen, M. Keil, and P. Cule, "Identifying software project risks: An international delphi study," *J. Manage. Inf. Syst.*, vol. 17, no. 4, pp. 5–36, Mar. 2001.
- [15] P. H. Ronald and Y. Y. Haines, "Software risk management," *Softw. Eng. Inst., Carnegie Mellon Univ. Pittsburgh, Pittsburgh, PA, USA, Tech. Rep. CMU/SEI-96-TR-012, ESC-TR-96-012*, Jun. 1996.
- [16] B. B. Roberts and L. Fussell, "Risk-based decision support," in *Proc. IncoSE Int. Symp.*, Jun. 1999, vol. 9, no. 1, pp. 604–610.
- [17] F. Basak Aydemir, P. Giorgini, and J. Mylopoulos, "Multi-objective risk analysis with goal models," in *Proc. IEEE 10th Int. Conf. Res. Challenges Inf. Sci. (RCIS)*, Jun. 2016, pp. 1–10.
- [18] F. Marmier, D. Gourc, and F. Laarz, "A risk oriented model to assess strategic decisions in new product development projects," *Decis. Support Syst.*, vol. 56, pp. 74–82, Dec. 2013.
- [19] S. Alhawari, L. Karadsheh, A. N. Talet, and E. Mansour, "Knowledge-based risk management framework for information technology project," *Int. J. Inf. Manage.*, vol. 32, no. 1, pp. 50–65, Feb. 2012.
- [20] B. Kamsu-Foguem and P. Tiako, "Risk information formalisation with graphs," *Comput. Ind.*, vol. 85, pp. 58–69, Feb. 2017.
- [21] A. Elzamly, B. Hussin, S. S. A. Naser, and M. Doheir, "Classification of software risks with discriminant analysis techniques in software planning development process," *Int. J. Adv. Sci. Technol.*, vol. 81, pp. 35–48, Aug. 2015.
- [22] M. P. S. O. M. Zaki and W. Li, "New algorithms for fast discovery of association rules," in *Proc. KDD*, vol. 97, Aug. 1997, pp. 283–286.
- [23] M. J. Zaki and K. Gouda, "Fast vertical mining using difffsets," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Aug. 2003, pp. 326–335.
- [24] M. Asif, J. Ahmed, and A. Hannan, "Software risk factors: A survey and software risk mitigation intelligent decision network using rule based technique," in *Proc. Int. MultiConf. Eng. Comput. Sci.*, vol. 1, pp. 12–14, 2014.
- [25] A. Aslam, N. Ahmad, T. Saba, A. S. Almazayad, A. Rehman, A. Anjum, and A. Khan, "Decision support system for risk assessment and management strategies in distributed software development," *IEEE Access*, vol. 5, pp. 20349–20373, 2017.
- [26] A. Benlian and T. Hess, "Opportunities and risks of software-as-a-service: Findings from a survey of IT executives," *Decis. Support Syst.*, vol. 52, no. 1, pp. 232–246, Dec. 2011.
- [27] A. Yavari, M. Golbaghi, and H. Momeni, "Assessment of effective risk in software projects based on Wallace's classification using fuzzy logic," *Int. J. Inf. Eng. Electron. Bus.*, vol. 5, no. 4, p. 58, 2013.
- [28] J. M. Verner, O. P. Brereton, B. A. Kitchenham, M. Turner, and M. Niazi, "Risks and risk mitigation in global software development: A tertiary study," *Inf. Softw. Technol.*, vol. 56, no. 1, pp. 54–78, Jan. 2014.
- [29] J. Noll, S. Beecham, and I. Richardson, "Global software development and collaboration: Barriers and solutions," *ACM Inroads*, vol. 1, no. 3, pp. 66–78, Sep. 2010.
- [30] A. B. Marques, R. Rodrigues, and T. Conte, "Systematic literature reviews in distributed software development: A tertiary study," in *Proc. IEEE 7th Int. Conf. Global Softw. Eng.*, Aug. 2012, pp. 134–143.
- [31] D. Šmite, C. Wohlin, Z. Galvina, and R. Prikladnicki, "An empirically based terminology and taxonomy for global software engineering," *Empirical Softw. Eng.*, vol. 19, no. 1, pp. 105–153, Feb. 2014.
- [32] S. Jalali and C. Wohlin, "Agile practices in global software engineering—A systematic map," in *Proc. 5th IEEE Int. Conf. Global Softw. Eng.*, Aug. 2010, pp. 45–54.
- [33] F. Q. B. da Silva, C. Costa, A. C. C. Franca, and R. Prikladnicki, "Challenges and solutions in distributed software development project management: A systematic literature review," in *Proc. 5th IEEE Int. Conf. Global Softw. Eng.*, Aug. 2010, pp. 87–96.
- [34] S. Patil and R. Ade, "A global software engineering knowledge management approach for intensive risk mitigation," *Int. J. Manag. Public Sector Inf. Commun. Technol.*, vol. 6, no. 1, pp. 1–8, Mar. 2015.
- [35] M. Pasha, G. Qaiser, and U. Pasha, "A critical analysis of software risk management techniques in large scale systems," *IEEE Access*, vol. 6, pp. 12412–12424, 2018.
- [36] M. Shahin, M. Ali Babar, and L. Zhu, "Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices," *IEEE Access*, vol. 5, pp. 3909–3943, 2017.
- [37] M. Mohsin, M. U. Sardar, O. Hasan, and Z. Anwar, "IoTRiskAnalyzer: A probabilistic model checking based framework for formal risk analytics of the Internet of Things," *IEEE Access*, vol. 5, pp. 5494–5505, 2017.



MUHAMMAD ASIF received the master's degree in computer science from Pir Mehr Ali Shah Arid Agriculture University, Rawalpindi, Pakistan. He is currently pursuing the Ph.D. degree in computer science with Abasyn University–Peshawar, Peshawar, Pakistan. He is also a Permanent Faculty Member of Foundation University Islamabad–Rawalpindi, Pakistan. His areas of research are software engineering, artificial intelligence, machine learning, and decision support systems.



JAMIL AHMED (Senior Member, IEEE) received the M.Sc. degree from the University of Warwick, U.K., and the Ph.D. degree from the King's College London. He is currently working as a Professor and the Vice Chancellor with Hazara University Mansehra, Pakistan. His areas of research are software engineering, artificial intelligence, artificial neural networks, image processing, and machine learning. He is a Professional Fellow of the British Computer Society.

...