International Journal of Computer Science and Software Engineering (IJCSSE), Volume 8, Issue 6, June 2019
ISSN (Online): 2409-4285 www.IJCSSE.org Page: 133-139

Risk Assessment and Management Method for Distributed Software Development Projects with "Fuzzy Approach"

Kökten Ulaş Birant[1], Ali Hakan Işık[2], Mustafa Batar[3], Hayri Bora Akarsu[4] and Ahmet Burak Tektaş[5]

[1] Department of Computer Engineering, Dokuz Eylül University, İzmir, Turkey

[2, 4, 5] Department of Computer Engineering, Burdur Mehmet Akif Ersoy University, Burdur, Turkey

[3] The Graduate School of Natural and Applied Sciences, Dokuz Eylül University, İzmir, Turkey

[1]ulas@cs.deu.edu.tr, [2]ahakan@mehmetakif.edu.tr, [3]mbatar@cs.deu.edu.tr, [4]ceborakarsu@gmail.com,

[5]abtektas@gmail.com

ABSTRACT

Software is a product or a service or a project whose requirements are captured; specification document is prepared by requirements analysis; in/out architecture is designed; related source codes, variables, methods, classes, modules and relationships among these are written; necessary tests and integration operations are done; when needed, maintenance, repair and update operations are executed by software or computer engineers and also that has its own documentation jobs and user manual, and has numeric and textual data, and also multimedia tools in its memory. Software development projects have a large financial burden and need to invest in high volumes. When looked at costs based on the international tangible data on computer software; it was $150 billion in 1985, it was $2 trillion in 2010 and it passed over $5 trillion after 2016. Also, in the year of 2018, just a daily giro of Apple Store was about $250 million. Despite of the costs, expenses and investments that are exponentially increasing every year, the rate of successful development of the software projects is not very high. Based on the "CHAOS" report (international size) prepared in 2016, only 17% of the software development projects were completed in a timely manner, in the allocated budget and in accordance with the requirements. 53% of the projects were completed over time and/or over budget and/or also without fulfilling the requirements exactly. 30% of the software projects cannot have been completed in the development phase and were cancelled. For that software development projects with such high expenses and low success rate can have a better quality structure, a risk assessment and management approach has to be determined for better software risk assessment and management methodology. So, some problems which may form software risks can be recognized and determined on time before causing trouble and endangering for software development projects. In this paper, several software risk assessment approaches underlying software risk management were introduced and explained in detail besides the key – "Fuzzy Approach". Moreover, that this "Fuzzy Technique" is more useful and more effective for software risk evaluation in comparison with the others was showed and expressed by giving meaningful and general linguistic rules and by applying a demo of software risk assessment under management with "Fuzzy Approach" (designed and developed by 18 original linguistic and logical rules with 15 different risk parameters in Python and in MATLAB) in this article. Also, this paper tried to explain the terms and the statements from the general to the specific as well as its content based on a hierarchical structure.

Keywords: *Software Engineering, Software, Software Project Management, Software Risk Management, Software Risk Assessment, Fuzzy Approach.*

1. SOFTWARE

"Software Engineering" may be described as a department that performs the engineering tasks in order to design and develop "Software" ("Computer Software"). This "Computer Software" is a product or a service or a project whose requirements are captured; specification document is prepared by requirements analysis; in/out architecture is designed; related source codes, variables, methods, classes, modules and relationships among these are written; necessary test and integration operations are done; when needed, maintenance, repair and update operations are executed by software or computer engineers and also that has its own documentation jobs and user manual, and has numeric and textual data, and also multimedia tools in its memory. In addition, "Computer Software" is an item that product engineers plan and assemble. It incorporates programs and applications that execute inside a PC or a machine or a hardware of any size and engineering that envelops printed version and virtual structures and information that join numbers and content, additionally incorporates portrayals of pictorial, video and sound data.

2. SOFTWARE DEVELOPMENT RISKS

"Dangers" are some piece of activities. Programming advancement ventures are not diverse as task arranging, and is finished with least data. Be that as it may, the level of hazard differs with multifaceted nature, size and area. Extension creep, absence of comprehension of issues, equivocal prerequisites and absence of assets, equipment, systems administration and security issues are a portion of the basic hazard components in programming advancement ventures. Along these lines, there is a need to oversee hazard in programming advancement. In spite of the fact that specialists and experts have composed on hazard board in programming improvement, almost no work have been done so as to include all the concerned partners in overseeing danger and incorporating the hazard the executives procedure with an all-encompassing undertaking the board approach. In different structures of programming hazard, the board has been proposed; some of which manage just the application part and a few models manage chance comprehensively covering application, association and between association levels. Nonetheless, there is a deficiency of writing on programming advancement chance administration from developers' viewpoint/perspective.

3. SOFTWARE PROJECT MANAGEMENT

A "Software Project" is an entire strategy of programming advancement from necessity social affair to testing and up-keeping, done by the execution procedures in a predefined timeframe to accomplish proposed programming item. Programming is said to be an immaterial item. Programming improvement is a sort of every single new stream in world business and there is next to no involvement in building programming items. Most programming items are customized to accommodate customer's prerequisites/requirements. The most vital is that the basic innovation changes and advances so oftentimes and quickly that experience of one item may not be connected to the next one. All such business and ecological limitations get hazard programming improvement thus it is basic to oversee programming ventures proficiently. Programming venture board is a workmanship and art of arranging and driving programming ventures. It is a sub-order of venture board in which programming ventures are arranged, executed, checked and controlled.

4. SOFTWARE RISK MANAGEMENT

"Programming Hazard Board" starts with the thought that product chance is an issue that should be overseen. Programming hazard at its center stems from issues inside the product itself, i.e., the source code that is presented amid advancement. Programming hazard board should then address two programming sorts of issues: "software disappointment and non-execution task" and "program executives and conveyance". Programming hazard board adopts a proactive strategy programming hazard by giving a methodology and system to search for zones where a product imperfection impacts the ease of use of the product for end clients and the business. For instance, a disastrous disappointment as the consequence of a product bug that does not enable the product to run accurately or at all is a kind of programming hazard that must be overseen. Programming hazard as an effect on undertaking the board, program boarding or conveyance is one in which programming imperfections and multifaceted nature affect the capacity to discharge programming on-time or inside spending plan. The effect here is in deferrals and expenses to the business that must be assimilated. For instance, an imperfection found late in the advancement procedure could result in re-work that takes days or weeks to address in this way deferring an undertaking. Both of these issues require solid hazard board practices in order to relieve against the hazard. However, "do individuals really deal with this hazard": in the first place, they ought to distinguish and comprehend the main driver/motorist.

5. SOFTWARE RISK ASSESSMENT

"Hazard Appraisal" is a term used to portray the general procedure or technique where: One recognizes dangers and hazard factors that can possibly cause hurt (risk distinguishing proof). One breaks down and assesses the hazard related with that danger (chance examination and hazard assessment). One decides fitting approaches to dispose of the danger or control the hazard when the peril cannot be dispensed with (chance control). A hazard evaluation is a careful take a gander at work environment to recognize those things, circumstances, forms and so forth that may cause hurt, especially to individuals. After ID is made, one investigates and assesses how likely and serious the hazard is. At the point when this assurance is made, one can straightaway, choose what measures ought to be set up to successfully take out or control the mischief from occurring. The CSA Standard Z1002 "Word Related Well-being and Security – Hazard ID and End and Hazard Evaluation and Control" utilizes/uses the accompanying terms: Hazard appraisal – the general procedure of peril

distinguishing proof, chance examination and hazard assessment.

6. SOFTWARE RISK ASSESSMENT APPROACHES

Risk may be defined as the likelihood and severity of occurrence of damage caused by the occurrence of bad or undesirable events. As a result, the system may suffer loss in terms of financial, structural or integrity. Risk management has a great importance in any field or project working. In software development, there are 4 main reasons to implement and apply "Software Risk Assessment under Management" according to Boehm:

A. In order to avoid overruns in budget and planning and to ensure that the software products run flawlessly and that software companies can develop software projects in their red lines/frames.

B. In order to prevent duplication of internal or external design or code resulting from defective, incomplete or unclear requirements that constitute in general 40% to 50% of the cost of software development.

C. In order to avoid unnecessary risk analysis and detection processes in areas that have (almost) no risk.

D. In order to promote a software solution when purchasing the software product/s that the customer needs, a win-win sat policy is applied to enable the sellers to obtain the desired profits.

If "Software Risk Assessment under Management" is to be applied to a system, then appropriate methods should be addressed, implemented and used. There are mainly 9 methods/ techniques in this context:

"Decision Trees" is one of the hierarchical algorithms frequently used for complex problems. It is a structure that makes a whole with a large number of inputs into smaller pieces according to the determined decision rules. There are rules, events and situations, as well as their outputs, i.e. the results.

"Event Trees" is a technique used for situations/circumstances/events that are first applied in the nuclear industry but which are used as a result of a risky event, which has been used actively in other fields of work over time. This technique contains a tree structure and has a design mechanism from the bottom to the top. Also, it has an inductive approach by trying to conclude from the current situations and events to the end.

"Fault Tree Analysis" may be defined as a process that deals with the combination of situations that may occur due to expected/unexpected reasons in the system under consideration. This tree structure has the opposite function of the "Event Trees" and has a working principle with deductive mechanism from the top to the bottom. In addition, there are logical rules and circuits in which the results attempt to reach the leading causes.

"Probabilistic Risk Analysis" has a changing structure based on the sector. It consists of 3 levels in total: "System Analysis", "Coverage Analysis" and "Results Analysis". The first level is the part that analyzes the dangerous situations and events that may occur in the system under consideration. The second level is the part where these dangerous situations and events (which parts of the system, modules, functions, etc.) determine the extent and how it affects them. At the last level, the results of damage or danger are revealed.

"Failure Mode and Effects Analysis (FMEA)" is a tool that was created to keep the rate of failures that cannot be repaired in the devices used by the United States Army as low as possible. It is divided into two main areas: "Design Failure Mode and Effects Analysis" and "Process Failure Mode and Effects Analysis". In the design area, parts failures and defects in the system are taken as a basis, and some researches are carried out in several areas such as security. Moreover, in the process part, it is concentrated on the human factor (man-power), and the machines and the equipment used in the production phase of the device. Furthermore, in FMEA, risks are considered and evaluated according to "Risk Preference Number". This indicator is calculated on the basis of "Probability of Risk Occurrence", "Severity of Risk" and Probability of Risk Detection". As a result of the multiplication of these three parameters, "Risk Preference Number" gets formed. Based on the value of this number (from the high to the low and in a directly proportional way), the risks are prioritized and the necessary prevention and minimization operations are performed in order, and sometimes, as the value of the number of some risks may be quite low, no action is taken and the risks can be left as be. In this analysis technique, measurement is very important, and evaluation is made and done according to this measurement process.

"Hazard Analysis and Critical Control Points" adopts the control, evaluation and definition of risk as a systematic approach, and has an abbreviation of HACCP in the literature. This risk assessment and management technique is used for general foodstuffs and products, and also in Turkey and in the world, to ensure that the movement in accordance with the standard "TS 13001". With contribution of this risk approach, several operations and processes such as planting food products, harvesting from the field, loading them to vehicles, storing, marketing, taking them from the market and taking them to the kitchen are analyzed, and necessary precautions are tried to be taken.

"Root Cause Analysis" may be defined as a set of processes that aim to define an event, why and how it occurs and also to prevent the repetition of the same event. By means of this technique, it is tried to reach

the main-basic cause/s by means of possible symptoms to treat the possible bad outcome/s of a situation or event, and to eliminate or minimize them.
"Risk Ranking and Filtering" is an important branch of risk assessment and management, and it is defined as the task of rating and filtering the risks and the relationships among each other based on their importance in some situations and events involving more than one risk. Also, the relationship between risks is calculated based on the probability of occurrence of risks and how much impact or violence will occur when risks occur/pop-up.
"Hierarchical Holographic Modeling" refers to the fact that each model part has a holographic sub-model where there is more than one model, and that the whole system cannot be viewed from a single point of view. When passing from a model to a sub-model/s, the hierarchy mechanism is operated and the upper-lower relationship is revealed and the upper and lower mechanisms are seen more clearly. In this way, the risks of the smallest unit/s can be revealed and seen more clearly and also, possible measures and precautions can be taken more easily for these unit risks.

7. "FUZZY APPROACH" IN SOFTWARE RISK ASSESSMENT UNDER MANAGEMENT

"Fuzzy Approach ("Fuzzy Logic")" can offer a much broader responses or response options to users rather than a binary system called dual/duplexed "0" and "1", which is known as the "Classic Logic" and also, the working principle of the computer. They have a rational reasoning and working mechanism, giving results and outputs based on probability and percentages like human beings. In this way, "Fuzzy Logic ("Fuzzy Approach")" has much more appropriate functions to identify the risks since the nature of the risk does not have a certainty structure and mechanism and is expressed in proportions.
"Fuzzy Approach" facilitates the control of complex systems that may change over time. Also, the logic of operations is not partial but rather partial in nature like in human beings. In addition, it gives good results in non-linear systems. Moreover, it is easy to be implemented and applied, and gives quick results. However, "Fuzzy Logic"s rules for the correct operation of the system are important. Experts' assistance should be taken into consideration when determining these rules. Some difficulties may occur in the definition of the rules. After these rules, loss time may occur since the trial-and-error method will be used. In addition, "Stability", "Observability" and "Auditability" analysis cannot be performed in this approach; this is the main problem of this method. Also, there are no clear and definite rules; therefore new definitions are required for each system/sub-system.

Some/Several linguistic rules of software risk assessment underlying management based on "Fuzzy Approach" for distributed software development projects:

*If "PM" is high and "TZD" is low then Risk is low.
*If "F" is high and "LD" is low and "CI" is high then Risk is low.
*If "T" is high and "TZD" is low then Risk is low.
*If "RS" is high and "NP" is low and "LD "is low and "CD" is low then Risk is low.
*If "CE" is high and "TC" is low then Risk is low.
*If "AK" is high and "CI" is high then Risk is low.
*If "Productivity" is high and "Communication" is high then Risk is low.
*PM: Process Maturity *TZD: Time Zone Difference
*F: Formality *LD: Language Difference
*CI: Communication Infrastructure *T: Transparency
*RS: Requirements Stability *NP: Novelty of Product
*CD: Cultural Differences *CE: Common Experiences *TC: Task Coupling *AK: Application Knowledge

8. AN APPLICATION OF SOFTWARE RISK ASSESSMENT AND MANAGEMENT WITH "FUZZY APPROACH"

"Software Risk Management" may be defined as the planned and regular control of the problems and troubles that may occur while designing and developing the software. In this way, this productive application has been developed by keeping away from the high cost of the development process and following the software life cycle in the most (as far as) accurate way. In this demo, a "Software Risk Assessment and Management System" with integrated "Artificial Intelligence" has been planned, designed and developed. As a conclusion of this work, software project managers may be able to reach the most accurate (as far as and based on the determined risk parameters set) result by using and applying this system instead of dealing with long plans (consume critically lot of time), and as well, this system may save "man", "time", and "price", which are leading both inputs and outputs of software development mechanism risk parameters.
The application has been designed and developed by the help of combination of "Python" (for user interface) and "MATLAB" (for functionality and operation). Also, in this work and this demo, there are 18 linguistic rules (own original logical rules) about risk assessment and management with 15 different risk parameters (determined and decided by literature review according to expert opinions) based on "Fuzzy Approach" for distributed software development projects:

*If “IDP” is medium and “IC” is medium and “UMSR” is high then risk is high.
*If “IDP” is low and “IC” is medium and “UMSR is medium then risk is medium.
*If “IDP” is low and “IC” is low and “UMSR” is low then risk is low.
*If “PMNCD” is high and “LEPMM” is medium then risk is high.
If “PMNCD” is medium and “LEPMM” is low then risk is medium.
If “PMNCD” is low and “LEPMM” is low then risk is low.
*If “IPP” is medium and “ITM” is high then risk is high.
If “IPP” is low and “ITM” is medium then risk is medium.
If “IPP” is low and “ITM” is low then risk is low.
*If “PIUNT” is high and “ITDTM” is medium and “NUSMFS” is medium then risk is high.
If “PIUNT” is medium and “ITDTM” is low and “NUSMFS” is medium then risk is medium.
If “PIUNT” is low and “ITDTM” is low and “NUSMFS” is low then risk is low.
*If “LAUI” is medium and “HLTC” is high then risk is high.
If “LAUI” is low and “HLTC” is medium then risk is medium.
If “LAUI” is low and “HLTC” is low then risk is low.
*If “COMDP” is medium and “CPNEP” is high and “FIAS” is medium then risk is high.
If “COMDP” is low and “CPNEP” is medium and “FIAS” is medium then risk is medium.
If “COMDP” is low and “CPNEP” is low and “FIAS” is low then risk is low.
*IDP: Inappropriate development process
*IC: Ineffective communication
*UMSR: Unclear or misunderstanding system requirements
*PMNCD: Project milestones not clearly defined
*LEPMM: Lack of an effective project management methodology
*IPP: Inadequate project planning
*ITM: Inexperienced team members
*PIUNT: Project involved the use of new technology
*ITDTM: Inadequately trained development team members
*NUSMFS: New and/or unfamiliar subject matter for the system
*LAUI: Lack of adequate user involvement
*HLTC: High level of technical complexity
*COMDP: Change in organizational management during the project
*CPNEP: Corporate politics with negative effect on project
*FIAS: Failure to identify all stakeholders
In this “Fuzzy Approach” application about software risk assessment and management;
*Input Range: low: [0 2 4] medium: [3 5 8] high: [6 8 10]
*Output Probability: low: [0 0.2 0.4] medium: [0.3 0.5 0.7]
high: [0.6 0.8 1]

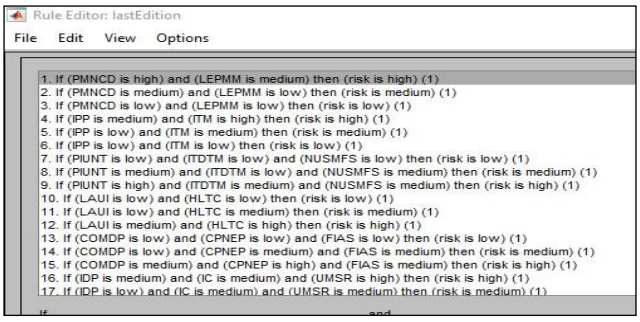

Fig. 1. 18 original linguistic rules with 15 different risk parameters on MATLAB.

Fig. 2 .User interface with 15 different risk parameters on Python.

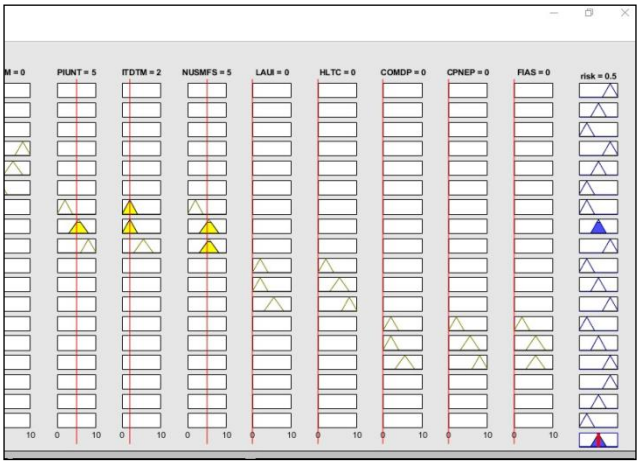

Fig. 3. Output of the rule on MATLAB: if “PIUNT” is medium (5) and “ITDTM” is low (2) and “NUSMFS” is medium (5) then risk is medium (0,5).

9. CONCLUSION

“Software Risk Management” is the most efficient way to design and develop software projects as one of the first works to be performed. Together with this risk management planning and its implementation, project managers make healthier decisions about the

future of projects and take more solid steps. In addition, the task of software project managers is to implement risk management in a project planning and to shape the project accordingly. In this context, this study means an example of that. The purpose of the study is to create and reveal risks to show the importance of "Fuzzy Logic" structure and working mechanism in order to see the results/outputs more clearly (with percentages and ratio/proportion). As a result of this importance, the risks can be expressed more clearly, and the planning of the road map and the operations can be made, implemented and done more clearly, the necessary prevention and minimization efforts can start at an earlier stage in the software development process, and thus, the success rate of software development projects can noticeably increase (will have increased).

In this paper, software risk assessment which is the basis of software risk management was dwelt and its prominent approaches (9 in total) – Fault Tree Analysis, Decision Tree, Probabilistic Risk Analysis, Event Tree, Failure Mode and Effects Analysis, Hazard Analysis Critical Control Point, Root Cause Analysis, Risk Ranking and Filtering and Hierarchical Holographic Modeling – were showed and told. In addition, "Fuzzy Approach" of software risk evaluation underlying management was explained in detail and by its linguistic and logical rules, that it works more effectively in the assessment level of software risk management was expressed in this work. Moreover, the designed and developed application (has original linguistic rules) about risk assessment and management with 15 different risk parameters based on "Fuzzy Approach" for distributed software development projects has showed and proved the effectiveness of the fuzzy logic and the rules like working as human mind.

"You can't manage the process which you don't measure." This statement which is claimed to be said by Peter DRUCKER shows that the software development process has to be measured by clear and objective variables –"Process Maturity", "Time Zone Difference", "Formality", "Language Difference", "Communication Infrastructure", "Transparency", "Requirements Stability", "Novelty of Product", "Cultural Differences", "Common Experiences", "Task Coupling" and "Application Knowledge" (these are meaningful and general software risk parameters). Thus, some reliable data are specified and determined so that this software risk evaluation process can be managed by benefiting from these trustworthy results. According to the results of this evaluation, manpower, what is the main resource of software development process will be used more effectively. And then, the benefits of the "Software Risk Assessment and Management" based on "Software Engineering" may be seen more tangible.

REFERENCES

[1] Roger S. Pressman, and Bruce Maxim, Software Engineering: A Practitioner's Approach, New York: McGraw-Hill Education, 2015.

[2] Ian Sommerville, Software Engineering, England: Pearson Education Limited, 2016.

[3] Prasanta Kumar Dey, Jason Kinch, and Stephen O. Ogunlana, "Managing Risk In Software Development Projects: A Case Study", Industrial Management & Data Systems, Vol. 107, No.2, 2007, pp. 284-303.

[4] Mohd. Sadiq, Md. Khalid Imam Rahmani, Mohd. Wazih Ahmad, and Sher Jung, "Software Risk Assessment and Evaluation Process (SRAEP) Using Model Based Approach", in International Conference on Networking and Information Technology, 2010, pp. 171-177.

[5] Abdullahi Mohamud Sharif, and Shuib Basri, "Software Risk Assessment: A Review on Small and Medium Software Projects", in International Conference on Software Engineering and Computer Systems, 2011, pp. 214-224.

[6] D. Verdon, and G. McGraw, "Risk Analysis in Software Design", IEEE Security & Privacy, Vol. 2, No. 4, 2004, pp. 79-84.

[7] Marvin Rausand, and Arnljot Høyland, System Reliability Theory: Models, Statistical Methods, and Applications, New Jersey: John Wiley & Sons, Inc, 2004.

[8] Yacov Y. Haimes, "Hierarchical Holographic Modeling", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 11, No. 9, 1981, pp. 606-617.

[9] Y.Y. Haimes, J. Lambert, Duan Li, R. Schooff , and V. Tulsiani, "Hierarchical Holographic Modeling For Risk Identification In Complex Systems", in IEEE International Conference on Systems, Man and Cybernetics, 1995, pp. 1027-1032.

[10] Ansgar Lamersdorf, Jürgen Münch, Alicia Fernández-del Viso Torre, Carlos Rebate Sánchez, Markus Heinz, and Dieter Rombach, "A Rule-based Model for Customized Risk Identification in Distributed Software Development Projects", in 5th IEEE International Conference on Global Software Engineering, 2010, pp. 209-218.

[11] Noel Garcia-Diaza, Cuauhtemoc Lopez-Martin, and Arturo Chavoya, "A Comparative Study Of Two Fuzzy Logic Models for Software Development Effort Estimation", Procedia Technology, Vol. 7, 2013, pp. 305-314.

[12] Sandeep Kad, and Vinay Chopra, "Fuzzy Logic Based Framework For Software Development Effort Estimation", An International Journal of Engineering Sciences, Vol.1, 2011, pp. 330-342.

[13] Ming-Yuan Hsieh, Yu-Chin Hsu, and Ching-Torng Lin, "Risk Assessment in New Software Development Projects at the Front End: A Fuzzy Logic Approach", Journal of Ambient Intelligence and Humanized Computing, Vol. 9, No. 2, 2018, pp. 295-305.

[14] Stephen G. MacDonell, and Andrew Gray, "The Viability of Fuzzy Logic Modeling in Software Development Effort Estimation: Opinions and Expectations of Project Managers", International Journal of Software Engineering and Knowledge Engineering, Vol. 15, No. 5, 2005, pp. 893-918.

[15] M. Wasif Nisar, Yong-Ji Wang, and Manzoor Elahi, "Software Development Effort Estimation Using Fuzzy Logic - A Survey", in Fifth International Conference on Fuzzy Systems and Knowledge Discovery, 2008, pp. 421- 427.

[16] P. K. Suri, and Kanchan Narula, "Strategies for Evaluating Risks to Project Schedule", in The Next Generation Information Technology Summit Confluence, 2013, pp. 1-6.

[17] Barry Boehm, Software Risk Management, New Jersey: IEEE Press, 1989.

[18] Sezgin Irmak, and Uğur Ercan, "Karar Ağaçları Kullanılarak Türkiye Hanehalkı Zeytinyağı Tüketimi Görünümünün Belirlenmesi", Uluslararası Yönetim İktisat ve İşletme Dergisi, Vol. 13, No. 3, 2017, pp. 553-564.

[19] Andrews, John D., and Sarah J. Dunnett, "Event-Tree Analysis Using Binary Decision Diagrams", IEEE Transactions on Reliability, Vol. 49, No.2, 2000, pp. 230- 239.

[20] Ahmed Ali Baig, Risza Ruzli, and Azizul B. Buang, "Reliability Analysis Using Fault Tree Analysis: A Review",International Journal of Chemical Engineering and Applications, Vol. 4, No. 3, 2013, pp. 169-173.

[21] Mayara De Melo Paranhos, Stella Jacyszyn Bachega, Dalton Matsuo Tavares, and Naiara Faiad Sebba Calife, "Aplicação Da Análise De Modo E Efeitos De Falha Para O Gerenciamento De Riscos De Um Projeto", Sistemas & Gestão, Vol. 11, No: 4, 2016, pp. 444-454.

[22] Frank L. Bryan, Hazard Analysis Critical Control Point Evaluations : A Guide To Identifying Hazards and Assessing Risks Associated with Food Preparation and Storage, Lithonia: World Health Organization, 1992.

[23] Mark T. Holdsworth, Rucha Bond, Saumeel Parikh, Bahie Yacop, and Kristina M. Wittstrom, "Root Cause Analysis Design and Its Application to Pharmacy Education", American Journal of Pharmaceutical Education, Vol. 79, No. 7, 2015, pp. 1-7.

[24] Kaplan S, Haimes YY, and Garrick BJ, "Fitting Hierarchical Holographic Modeling into the Theory of Scenario Structuring and a Resulting Refinement to the Quantitative Definition of Risk", Risk Analysis, Vol. 21, No. 5, 2001, pp. 807, 819.

[25] Gökhan Gökmen, Tahir Çetin Akıncı, Mehmet Tektaş, Nevzat Onat, Gökhan Koçyiğit, and Necla Tektaş, "Evaluation of Student Performance in Laboratory Applications Using Fuzzy Logic", Procedia Social and Behavioral Sciences, Vol. 2, No. 2, 2010, pp. 902-909.

[26] Buket Aksu, "Tasarımla Kalite (Quality By Design: QbD) ve Kalite Risk Yönetimi", Marmara Pharmaceutical Journal, Vol. 19, 2015, pp. 12-18.

[27] Viktorija Stepanova, "Quality Control Approach in Developing Software Projects", International Journal of Computer Science and Software Engineering, Vol. 8, No. 1, 2019, pp. 1-5.

[28] Elizabeth S. Furtado, and Danielly Barboza Guimarães, "A Framework to Evaluate the Quality in Use of a System from Sociotechnical Questions", International Journal of Computer Science and Software Engineering, Vol. 2, No. 1, 2014, pp. 42-49.

[29] Mazin Abed Mohammed, "Design and Implementing an Efficient Expert Assistance System for Car Evaluation via Fuzzy Logic Controller", International Journal of Computer Science and Software Engineering, Vol. 4, No. 3, 2015, pp. 60-68.

[30] Shahzad Hussain, Shahid Farid, and Iqra Mumtaz, "Is Customer Satisfaction Enough for Software Quality?", International Journal of Computer Science and Software Engineering, Vol. 8, No. 2, 2019, pp. 40-47.

[31] Sanjay Kumar, Dr. Rahul Rishi, and Dr. Rajkumar, "Metrics to Determine the Quality and Productivity during Software Development", International Journal of Computer Science and Software Engineering, Vol. 5, No. 8, 2016, pp. 175-182.