International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-7, Issue-6, March 2019

Quantitative Analysis of Development Environment Risk for Agile Software through Machine Learning

Anand Kumar Rai, Shalini Agarwal,Mazahar Khaliq, Abhishek Kumar

***Abstract*: Agile methodology practice has increased in today's era of software industries. In this study the 9 risk elements of the agile software development environment have been identified. The qualitative value of risk elements have been converted into the quantitative form with the help of a fuzzy inference system. These quantitative values have been used to train the back propagation network. This study will contribute significantly in reducing risks in the use of the agile methodology, because the risks are accurately expressed in a quantitative way. This study has been performed on the software projects made on agile methods XP and Scrum with the help of MATLAB simulator.**

***Index Terms*: Agile Software, AI Learning, Back Propagation Network, Fuzzy InferenceSystem**

I. INTRODUCTION

An Agile Software Development Environment (ASDE) is defined as an environment that creates and supports the art that encourages a team to work toward a common goal. This is done by including the importance and value of individuals and their interactions particularly in terms of achieve collaboration, quality, and consent of frequent change in the company culture. Agile environments are highly collaborative, individuals and interactions are valued more than tools and processes, tele-communication and other tools should not replace face-to-face conversations, which are the most effective ways to communicate. Not only does productivity increase, but also the interactions between team members help create trust and an atmosphere of collaboration (Subhas Misra et al, 2012).

Agile is a set of principles that separate into pieces large size portions of a project into smaller operating part that can be developed quickly. This series of action also has been familiar as iterative software development. The word 'agile' became an officially recognized term used by large number of software development industries. In 2001, Seventeen software developers bring into existence the agile manifesto, which lists principles or standards of iterative software development. During the time there are many software development methods that fall under the class of being agile. The agile manifesto is more about behavior and refined understanding than a set of methods. A large number of software development companies make efforts to be agile because this helps them to deliver quality software to their customers faster in comparison of non-agile competitors. At the present time, we have a small history of the word agile in the context of software development (Laurie Williams, 2012). In this study, the risk elements of the ASDE have been identified and qualitative risk assessment value is converted into quantitative form with the help of a Fuzzy Inference System (FIS), so that the Agile Environment Risk (AER) can be estimated precisely. The quantitative value of risk elements has been collected for the backpropagation network training purpose from the FIS rule view.FIS is the method of expressing the mapping from a given input to an output using fuzzy logic. The mapping then make available a support from which decisions can be made, or patterns find out. The series of actions in FIS comprises membership functions, fuzzy logic operators, and if-then rules. Two kinds of FIS existing that can be instigated in the fuzzy logic that is Mamdani and Sugeno. These two types of inference systems differ rather in the way outputs are determined. FIS have been effectively applied in fields of expert systems, data classification, decision analysis, and computer vision. Because of its multidisciplinary nature, FIS are associated with a number of names, such as fuzzy-rule-based systems, fuzzy expert systems, fuzzy modelling, fuzzy associative memory and fuzzy logic controllers (Yi Wen Kerk,2017). Present work focuses on artificial neural networks, uses backpropagation learning algorithm for the computation of a gradient descent with respect to weights. Targeted outputs are compared with system outputs, again the systems are adjusting connection weights to minimize the difference between targeted output and system output. Nowadays the backpropagation algorithm is the reliably over a long period of learning in neural networks (De Jesús et al, 2007).The present study organized as follows, section 2.0 presents an overview of research on agile software development environment, and we examine journal publications and citations related to agile development environment to trace the outline of the structure of the field. Subsequently, in section 3.0, we proposed a model for agile software development environment risk evaluation and AI learningthrough backpropagation algorithm. section 3.1 describes identification of risk elements for agile software development environment, section 3.

Revised Manuscript Received on March 20, 2019.

Mr. Anand Kumar Rai, Research Scholar, Shri Ramswaroop Memorial University, Computer Science & Engineering, Barabanki, INDIA

Dr. Shalini Agarwal, Associate Professor, Shri Ramswaroop Memorial University, Computer Science & Engineering, Barabanki, INDIA

Dr. Mazahar Khaliq, Assistant Professor, Computer Application, KMC Urdu Arabi Farsi University,Lucknow, INDIA

Dr. Abhishek Kumar, Assistant Professor, Department of Computer science, Banaras Hindu University, BHU, Varanasi, INDIA

2 presents the risk elements fuzzification process, section 3.3 describe the fuzzy rule base creation, section 3.4 describe about the defuzzification and data collection, section 3.5 describe neural network setup and training process, section 4.0 contain research discussion and result, section 5.0 describe conclusion and future work.

II. LITERATURE REVIEW

A survey was conducted with agile professional, which collected survey data from 109 countries of agile project from 25 countries around the world. The result shows that only 10 of the 48 hypotheses were supported, in this research four agile project failure factors were identified, which are organizational, people, process and technical, under these four factors 19 risk elements are discussed. Team focus, agile planning, sponsor support etc. are the identified risk elements for agile environment (Tsun Chow et al., 2008).

In a case study, result shows that the extended XP can be constructive for medium scale and large scale software development projects (M. Rizwan et al., 2012). An attempt has been made to identify such risks that are due to the introduction of a tight mechanism for a project or become more important when using such method. The results of this research can be useful for any organization which is in the process of selecting a method for project management and is considering tight procedures. Product owner role, fixed time, resources and quality and few more risk elements were identified (WojciechWALCZAK,Dorota KUCHTA,2013). Distributed agile approach is rapidly growing nowadays and so, the study of risks in such environments becomes of vital importance. A number of studies have discussed about the problems tackled by distributed agile teams (Suprika Vasudeva Shrivastava et al., 2014).

Another study attempts to combine the prevailing studies on risks in distributed agile development. It helps in discovery the areas of risk management in distributed agile. A quantitative study with agile software professional in which data was collected through systematic questionnaire and was analysed using technical descriptive, factor and cluster analysis. Based on the classification of factors, three software organization profiles have been obtained in relation to the use of tight principles and the scope of success in production of software. This study is very useful because it helps in improving our understanding of the use of tight principles in software development; these principles are associated with success in software production; therefore, it explores a new risk elements which has not been seen before(Paulo Henrique de Souza Bermejo et al., 2014).

Nowadays software companies are responsive to severe competition and they need to release new versions. To achieve the speedy goal companies take shortcuts and these shortcuts resulting exclude quality are called technical debt. It was examined one middle-size Finnish Software Company to comprehend the reasons and effects of technical debt with two self- governing product lines and interviewed 12 persons in diverse positions. The researchers were also concerned in precise approaches and practices for managing technical debt (Jesse Yli- Huumo et al., 2014). The results of this research indicates that technical debt is usually formed as a result of deliberate decisions made to reach the project deadlines. Customer contentment was recognized as the key reason for taking technical debt but in the longer perspective it turned to monetary values and quality issues. It was seen that there was not any explicit management plan for minimizing technical debt but numerous practices have been acknowledged.

Another study presents and discourses the metric “Risk Point”, finding some points of corrections. This paper presents sustained effort of the metrics in the environment of multiple projects of software development with the objective of examine its applicability and usefulness for decision-making and risk monitoring during project development life cycle(Miguel Wanderley et al.,2015). Most important success factor categories were management support, training and coaching tight models, brain set and arrangement to choose and adapt (Kim Dikert et al. 2016). Agile software agent has been used to identify and monitor risk elements. The results shown using case studies that the risks of agents are used to detect risk and to dynamically react to changes in the project environment, thus helping to reduce the human effort of risk management (EdzreenaEdzaOdzaly et al. 2017).

III. PROPOSED MODEL FOR ASDE RISK EVALUATION &LEARNING

This study has been organized performed in the following steps.

Figure 1, shows the proposed model.

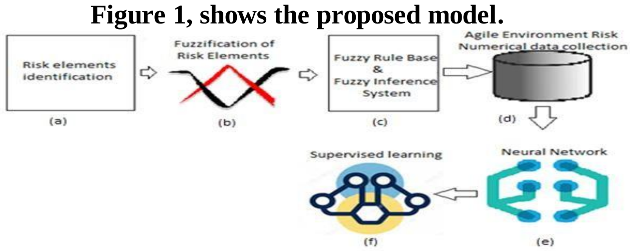

Figure 1. Agile development environment risk evaluation and risk learning(proposed model)

- Identification of agile software development risk.
- Fuzzification of risk elements.
- Fuzzy rule base creation&implementation.
- Defuzzification/ data creation and collection from fuzzy rule view (MATLAB Simulator). Neural network setup (Feed forward network).
- Supervised learningofbackpropagationnetwork.

IV. RISK ELEMENTS IDENTIFICATION

In this study the agile software development risk elements have been identified and incorporating them forming the rule base for the FIS to ensure the success of the agile software project. Table 1, shows the 9 identified risk elements which affects small scale, medium scale and large scale agile projects (EdzreenaEdzaOdzaly et al., 2017; Paulo Henrique de Souza Bermejo et al., 2014; Tsun Chow et al., 2008; Wojciech WALCZAK et al, 2013). The brief description of identified risk elements are as follows:

1-Distributed Project a distributed project is defined as a group of members actively work together on a shared software/ systems project separated by distance, time zone and Culture (Suprika Vasudeva Shrivastava et al, 2014).

2-Agile Training Agile training is the perfect means to level-set in organization or project team on the basic concepts of agile and implementation of it. There are some misconception about the distinctions between different methods. Agile training helps to expose the core agile concepts and make clear the differences between the several implementation methods (Kim Dikert et al. 2016).

3-Product Owner Role The product owner is the responsible for handling the product backlog. This includes (WojciechWALCZAK,Dorota KUCHTA,2013):

(i)Evidently expressing product backlog items.
(ii)Ordering the items in the product backlog to achieve goals.
(iii) Improving the value of the work which the development team performs.
(iv) Safeguarding that the product backlog is observable, transparent and displays what the scrum team will work on next.
(v)Ensuring that the development team recognizes things in the product backlog accordingly the level.

4-Team Focus is a cross-functional group of employees that sees regularly to the key organizational performance areas, monitoring the usefulness of the systems that backing this focus area, asses action success thus far and make the adjustments needed (Tsun Chow et al.,2008).

5-Agile Planning is the action that brings into line the teams to the common project goal. It interprets business purposes to features and user stories which the team solves to reach the project goal (Tsun Chow et al., 2008).

6-Sponsor Support Sponsors are responsible for the size of the project in terms of cost and degree of importance. Sponsors authorize all project expenditure and resource usage. Sponsor ensures that the project meets the original objectives (Tsun Chow et al., 2008).

7-Automation Testing in agile software means using an automation tool to execute the test case suite. Automation software can also store test data in the system under testing, compare expected and actual results, and generate detailed test reports. When the test suite is automated, human intervention is not required. The automation reduce the number of test cases to be run manually and not to exclude manual testing altogether.

8-Technical debt to build up is a concept in software development that imitates the indirect cost of supplementary re-work triggered by choosing an easy solution, instead of using a better approach that would take longer. If technical debt is not reimbursed, it can hoard 'interest', later this makes the change difficult. Unadjusted technical debt increases software entropy (Jesse Yli-Huumo et al., 2014).

9-Fixed time, Resources and Quality Time is the available schedule to complete the project, cost signifies the sum of money or resources existing and quality signifies the goal that the project must attain to be a success. Typically one of these factors

is fixed and the other twowill differ in inverse ratio to each other (WojciechWALCZAK,Dorota KUCHTA,2013).

Table 1. Agile Development Environment Risk Elements List

S.N.	Risk Elements	Abbreviation
1	Distributed Project.	DE1
2	Agile training.	DE2
3	Product owner role.	DE3
4	Team focus.	DE4
5	Agile planning	DE5
6	Sponsor support.	DE6
7	Test automation.	DE7
8	Technical debt to build up.	DE8
9	Fixed time, resources and quality.	DE9

Many researchers estimated/evaluated the environment risk by putting the value of risk elements in true/false or qualitative manner.In this study risk elements qualitative values have been represented in low, medium and high manner, and the qualitative data (low, medium, and high) have been converted in quantitative manner and further the learning has been performed for precise risk value prediction. Table 2, shows the rule base for FIS, in which total 12 rules have been constructed with the help of literature review and case studies. Agile environment success and failure project literature helped in making the rule base. Figure 2, describes the method of converting the qualitative data into quantitative data. In this manner low, medium and high qualitative values are displayed through the membership value, which range is given through 0 to 1. Figure 3, shows the relation of agile environment risk elements, FIS and agile environment risk.

V. FUZZIFICATION OF RISK ELEMENTS

Because We will consider fuzzy set A,
$A = \{(x, \mu A(x))| x \in X\}$, where $\mu A(x)$ is known as the membership function for the fuzzy set A. X is mention as the universe of discourse.

The membership function make conceptual connection between each element x Є X in the interval [0, 1]. In fuzzy sets, each elements are mapped to [0, 1] by membership function. That is, $\mu A : X \in [0, 1]$.The fuzzy set A can be available as another fact of being possible denoted as follows:

If X is discrete then $A = \Sigma \mu A(xi) / xi$ Eq. 1
If X is continuous then $A = \int \mu A(x) / x$ Eq. 2

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Where, μA (x) is the "membership function", value of this function lies between 0 to 1. This value act as a substitute for the "degree of membership" of element x in set A. The members of a fuzzy set holds some degree which is grade or degree of membership. The greater the number between 0-1, more the degree of belonging. The process of translating from x to μA(x) is called fuzzification. According to fuzzy theory,

fuzzy set A of universe X is defined by function μA(x), which is a membership function of set A.

$\mu A(x): X \rightarrow [0, 1]$, where $\mu A(x) = 1$, if x is full in A;

$\mu A(x) = 0$, if x is not in A;

$0 < \mu A(x) < 1$, if x is not completely in A.

Fuzzy set allows a continuous sequence of possible choice.

In this study for each input and output variable selected, we define three Membership Functions (MF), we have a qualitative category for each of them, for example: low, medium and high. The shape of these functions can be varied in triangles and trapezoids but in this study usually we will work with triangles. In given Figure 2, the triangle which represent the low value of the input variable ranges from 0 to 0.4, second triangle represent medium value of input variable ranges from 0.1 to 0.9 and third triangle represent the high value of input variable ranges from 0.6 to 1. The membership function value ranges from 0 to 1. The qualitative values of input variables (risk elements) have been converted in quantitative form through de- fuzzification process which is described in next section 3.3.

VI. FUZZY RULE BASE CREATION

Once we define the input, output variables and the Membership Function (MF), then we create the rule base IF <antecedents> THEN <conclusions> rules. These rules convert the output risk to operational value based on the input variables. This output variable also have to be defined with MF, usually low, medium and high risk. A rule base shown in Table 2, is created by reviewing literature and online interview with the agile experts. The rule base formation is based on the qualitative value of all risk elements and estimated qualitative value of agile environment risk, thus total 12 rules have been made (Bruno Antunes et al., 2011; EdzreenaEdzaOdzaly et al., 2017; Kim Dikert et al., 2016; Paulo Henrique de Souza Bermejo et al., 2014; Tsun Chow et al., 2008; Tore Dyba et al., 2008; Wojciech WALCZAK et al, 2013).

Table2. Agile Development Environment Risk Rule Base

S.N	DE1	DE2	DE3	DE4	DE5	DE6	DE7	DE8	DE9	DER
1.	L	L	L	L	H	L	L	H	H	H
2.	M	M	M	M	M	M	M	M	M	M
3.	L	H	H	H	L	H	H	M	M	H
4.	H	H	H	H	H	H	L	L	H	M
5.	M	L	M	H	M	H	M	H	M	M
6.	L	L	L	M	M	L	L	M	M	H
7.	H	H	H	M	H	H	H	M	L	L
8.	H	L	L	M	L	L	L	L	H	H
9.	M	H	H	H	M	H	M	M	M	L
10.	L	L	L	L	L	L	L	M	M	H
11.	H	H	H	H	H	H	H	L	H	L
12.	H	L	M	H	M	M	M	L	L	L

After implementing this rule base in a FIS, the quantitative value of all the risk elements (inputs) and the agile environment risk (output) has been evaluated. Figure 4, shows the rule base with FIS.

VII. DEFUZZIFICATION AND DATA COLLECTION

Defuzzification is the process of creating a quantitative result in crisp logic given fuzzy sets and have a close similarity membership degree. It is the series of actions that maps a fuzzy set to a crisp set, this is usually used in fuzzy control logic. In this, the variable values are converted into fuzzy results based on rule base. These values are added with the aggregation method and further these aredefuzzfied with the help of Center of Gravity Method (COG), which is a common technique of defuzzification. The common fuzzy set membership function triangle graph is used in this study.

The data set shown in Table 3, is collected through

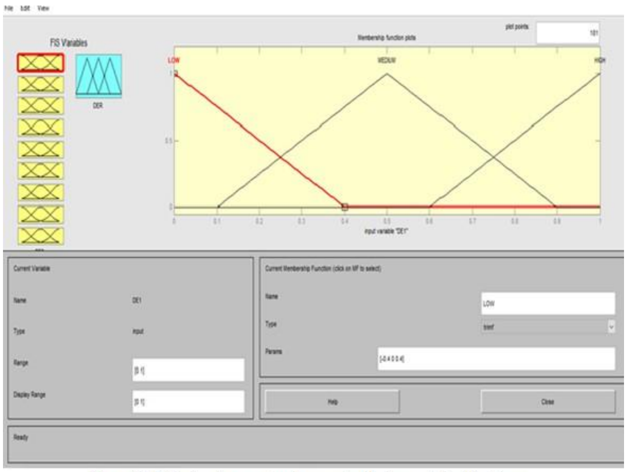

Figure 2. Agile development environment risk elements fuzzification

the fuzzy rule view of the matlab simulator. Figure 5, shows that vertical lines have been formed with the triangle, the value of the risk elements are determined by sliding these lines forward or backward and based on the value of these elements the Agile Environment Risk (AER) is calculated by the aggregation of the rule base, thus, the value of the risk elements and the value of the AER values are read. This process runs until the minimum value of the AER is received. Table 2, shows the rule base representing the risk elements qualitative values. The rule base is further applied to FIS.

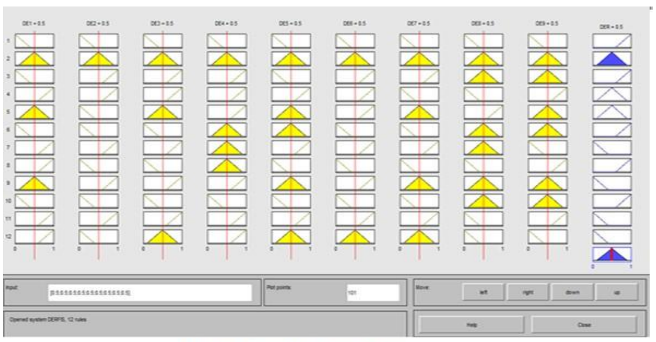

Figure 5. Fuzzy Inference rule view

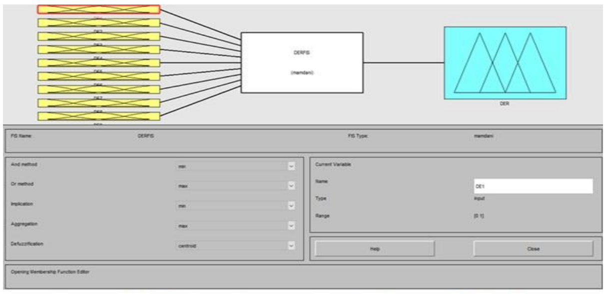

Figure 3. Agile development environment risk , risk elements and FIS relation

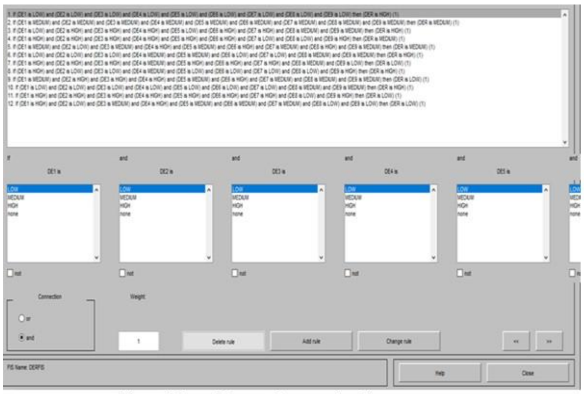

Figure 4. Fuzzy Inference System and rule base

VIII. NEURAL NETWORK SETUP AND TRAINING

The To set up the neural network for the training purpose, we have to decide the number of input neurons and output neurons. The number of input neurons are depend on the number of input variables and the number of hidden neurons are determined according to training performance. The number of neurons on the output layer depends on the output variable. The number of input variables are 9 and the output is single which is DER. The MATLAB simulator create the network according to Figure 6, with the help of the fitting function of the neural network. Set of DERINPUT[] input vectors as row and columns in a matrix are arranged. Then, it is arranged another set of DERTARGET[] vectors. With these settings, the input vectors and target vectors are randomly divided into three sets , which are as follows:

(i)70% of the input and output vector data sets are used in training.

(ii)15% of the input and output vector data sets are used in network verification.

(iii)The last 15% are used as a completely independent test of network generalization.

The quality network that is used for function fitting, consist of two-layer feed forward network, in the hidden layer with sigmoid transfer function and an output layer with linear transfer function. The default number of hidden neurons are set to 10 (De Jesus et al., 2007). If the network training performance is poor the number of hidden neurons may be increased or decreased. The following regression plots in Figure 7, display the network outputs with respect to targets for training, validation, and test sets. The data should fall along a 45 degree line for the perfect fit, where the network outputs are equal to the desired targets. The fit is fairly good for all data sets with R values 0.93 or above in each case. If we want more accurate results yet, we could retrain the network by clicking retrain in nftool. This might change the initial weights and biases of the network, and may produce better network after retraining. on the following pane another options are provided. It is a good idea to check the outliers to determine if the data is bad, or if those data points are different than the rest of the data set. If the outliers are valid data points, but are unlike the rest of the data, then the network is extrapolating for these points.(Feng,J.et al. 2003). In this case, we can test the network with new data or increasing the number of hidden neurons and we can take another larger data set for the training purpose. In case, training set performance is good, but the test set performance is significantly worse, which could indicate overfitting, in such case reducing the number of neurons can improve the results. If training performance is poor, then number of neurons can be increased, if the network performance is satisfactory then simulator result can be saved for application purpose.

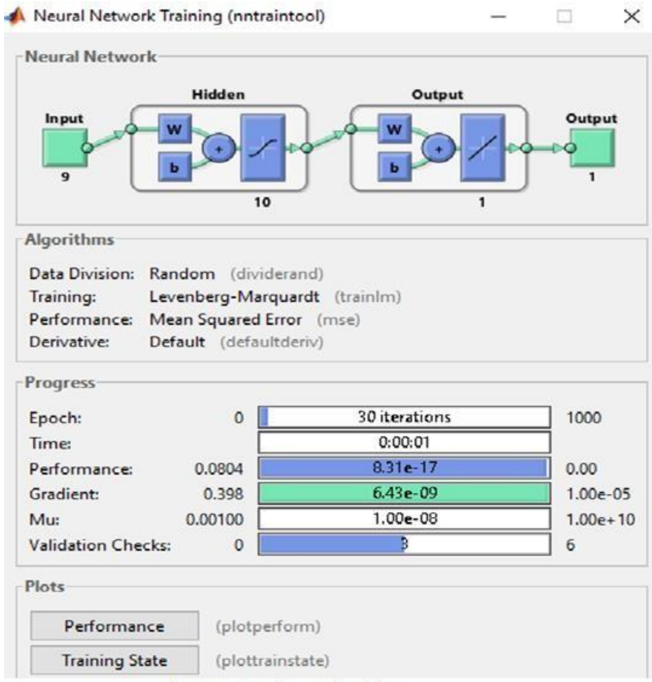

Figure 6. Neural network training

If there is more deviation in the training result, then the number of hidden neurons can be increased. Initially training was started with 20 dataset but the training performance was not upto the mark. In case of worse training performance we have to increase the number of data sets. In this study 62 data sets have been taken, this data set has been obtained from FIS. The Neural Network was trained with this same data set. The performance of network training can be seen in Figure 8. The result is well described in next section 4.0.

IX. DISCUSSION AND RESULT

Figure 7, shows the regression analysis chart for mse error performance function with logsig activation function.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Within this graph the training R=0.99999 and the test R=0.96997 and the validation is 0.99357. Finally the all R =0.99357. In this research by varying the simulation is trained the NN tool in three performance functions and the three activationfunctions. Figure 8, also shows the error histogram of the trained neural network for the training, validation and testing steps. This figure shows that the data fitting errors are distributed within a reasonably good range around zero. Figure 9, shows the training state graph with mse performance function and transig activation functionthe Gradient=6.4268e-009 at epoch 30, mu=1e-008 at epoch 30 and the validation checks=3 at epoch30.

Table 3. ASD Risk Elements and Environment Risk Data Set for trained network performance testing

S.N.	DE1	DE2	DE3	DE4	DE5	DE6	DE7	DE8	DE9	DER
1	0.0812	0.1187	0.0937	0.1063	0.0937	0.0812	0.0937	0.0937	0.1062	0.8600
2	0.1438	0.1937	0.1687	0.1812	0.1813	0.1687	0.1562	0.1938	0.1563	0.7240
3	0.2312	0.2687	0.2437	0.2313	0.2813	0.2312	0.2187	0.2563	0.1563	0.6470
4	0.2312	0.2687	0.2437	0.2313	0.2813	0.2312	0.2187	0.2563	0.2937	0.5910
5	0.3063	0.3438	0.3312	0.3438	0.4063	0.4063	0.3562	0.3563	0.3812	0.5000
6	0.7813	0.7937	0.7437	0.7687	0.7313	0.7937	0.8062	0.7688	0.6187	0.4240
7	0.7813	0.8562	0.7437	0.7687	0.7313	0.7937	0.8062	0.7688	0.6187	0.3540
8	0.7813	0.8562	0.8312	0.8438	0.7438	0.8938	0.8062	0.7688	0.6187	0.2150
9	0.8688	0.9063	0.8813	0.9063	0.8188	0.8938	0.8437	0.8688	0.8062	0.1900
10	0.8688	0.9063	0.8813	0.9063	0.8188	0.8938	0.8437	0.2063	0.8187	0.1540

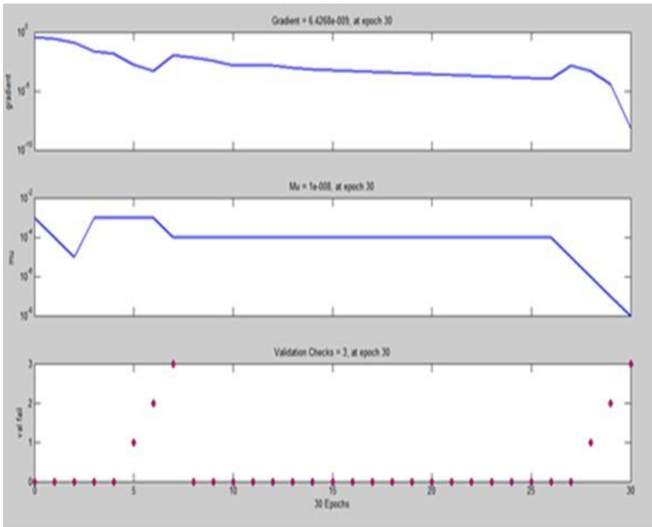

Figure 9. Neural network training state

Figure 10, shows the performance validation graph with transig transfer function and mse error performance function. The best validation performance is .00068052 at epoch 27. The train graph and test graph having slight difference in performance. The validation is almost similar to train graph. It is best fitted at epoch 27.

After the training completion of the network, it has been tested with different data sets which is given in table 3. It is clear from the given Figure 11, regression plot shows that the training performance is around 97%, which is good. This data set has been generated from Fuzzy Inference system. With the help of Figure 2, we can assume quantitative value in different qualitative band which are shown in Table 4. The present work suggests 4 sensitive risk elements out of 9. The Project manager will have to well manage these 4 sensitive risk elements for the successful agile project development; the agile training(DE2), product owner role(DE3), team focus(DE4) and sponsor support(DE6) and rest 5 may be well managed or partially managed. In this study we have found that if distributed project(DE1) is low(0.1062), agile training (DE2) is high(0.9187), product owner role (DE3) is high(0.9312), team focus (DE4) is high(0.8937),agile planning (DE5) is low(0.1938),sponsor support (DE6) is high(0.8812), test automation (DE7) is low (0.1562), technical debt to build up (DE8) is low(0.1438), fixed time, resources and quality (DE9) is low(0.1937) then development environment risk (DER) will be low(0.195).

Table 4 Qualitative value representation in quantitative band

S.N	Qualitative value	Quantitative value	MF value
1	Extreme low	0	1
2	Low	0< to <0.4	1>to <0.75
3	Medium	0.4 to <0.6	0.75 to 1 to >0.75
4	High	0.6 to <0.9	0.75 to <1
5	Extreme high	1	1

During the agile project development agile training must be provided to the team, product owner role must be described at high level, development team must be highly focused towards the goal and agile sponsor support is necessary for the successfully accomplishment of the project.

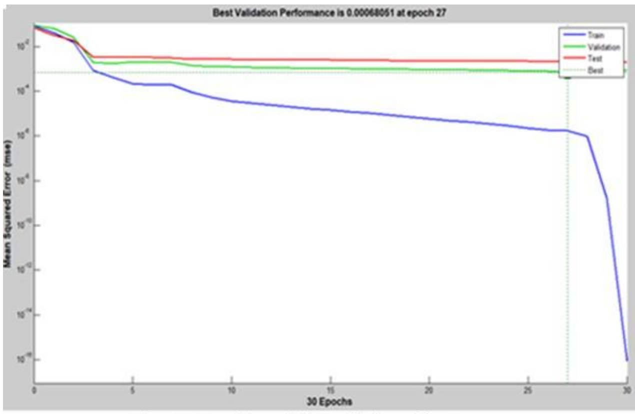

Figure 10. Neural network best validation performance

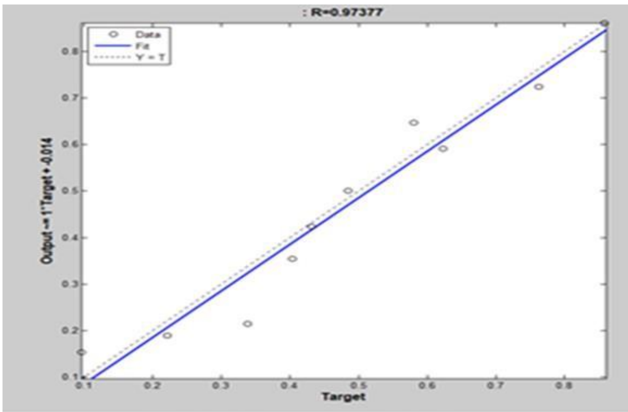

Figure 11. Neural network performance testing

X. CONCLUSION AND FUTURE RESEARCH

In this study, the qualitative data has been presented in the quantitative form through Fuzzy

System. In this study, we have identified 9 risk elements, out of nine risk elements 4 sensitive risk elements have been identified, that is the agile training, product owner role, team focus and sponsor support which are affecting to agile development environment. A rule base has been formed with the help of a case studies, this rule base has been combined with fuzzy inference system using MATLAB simulator. The training simulation result indicate the feedforward backpropagation model produce a best predictive results for neural network.

In this study, 15% data set out of the total data set is randomly selected for the validation check and 15% data set for the performance testing. Ultimately network performance on some different data set was checked which contained up to 97% of accuracy. XP and SCRUM agile methods have been studied to identify the risk elements and forming the rule base. Further this study can be perform on other methods of agile development. In future study, more risk elements can be identified and rule base may also be increased and thus training accuracy can be increased.

REFERENCES

1. A. Qumer, B. Henderson-Sellers," A framework to support the evaluation, adoption and improvement of agile methods in practice", The Journal of Systems and Software 81 (2008)1899–1919, Elsevier.
2. AalaaAlbadarneh, AalaaAlbadarneh, AalaaAlbadarneh," Risk management in Agile software development: A comparative study", 2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT).
3. Bruno Antunes, Diogo Santos, Eurico Lopes, Filipe Fidalgo, Paulo Alves," Blisstrail: an agile project business case study", Procedia Computer Science 64 (2015) 529 – 536, Elsevier.
4. Bruno Antunes, Diogo Santos, Eurico Lopes, Filipe Fidalgo, Paulo Alves," Blisstrail: an agile project business case study", Procedia Computer Science 64 (2015) 529 – 536, Elsevier.
5. Breno G. Tavares, Carlos Eduardo S. da Silva and Adler D. de Souza," Risk Management in Scrum Projects: A Bibliometric Study",JOURNAL OF COMMUNICATIONS SOFTWARE AND SYSTEMS, VOL. 13, NO. 1, MARCH 2017
6. De Jesús, O., and M.T. Hagan, "Backpropagation Algorithms for a Broad Class of Dynamic Networks," IEEE Transactions on Neural Networks, Vol. 18, No. 1, January 2007, pp. 14 -27.
7. EdzreenaEdzaOdzaly, Des Greer ,Darryl Stewart," Agile risk management using software agents", J Ambient Intell Human Comput(2017), 12652-017-0488-2, Springer.
8. Juhani Iivari , NettaIivari ," The relationship between organizational culture and the deployment of agile methods", Information and Software Technology 53 (2011) 509–520, Elsevier
9. Kamran Ghane," Quantitative planning and risk management of Agile Software Development", 2017 IEEE Technology & Engineering Management Conference (TEMSCON)
10. Kim Dikert , Maria Paasivaara , Casper Lassenius," Challenges and success factors for large-scale agile transformations: A systematic literature review", The Journal of Systems and Software 119 (2016) 87–108, Elsevier.
11. Laurie Williams ,"What Agile Teams Think of Agile Principles",Communications of the ACM, Vol. 55 No. 4(April 2012), Pages 71-76,ACM.
12. M. Rizwan,Jameel Qureshi," Agile software development methodology for medium and large projects", IET Software (Volume: 6, Issue: 4, August 2012) Page(s): 358 – 363, IEEE Xplore.
13. Miguel Wanderley, Julio Menezes Jr., Cristine Gusmao, Filipe Lima," Proposal of risk management metrics for multiple project software development", Procedia Computer Science64 (2015) 1001 – 1009, Elsevier
14. Paulo Henrique de Souza Bermejo, André Luiz Zambaldea, Adriano Olímpio,Tonelli, Samara Alyne Souza, Larissa Avelino Zuppo, Priscila Luiz Rosa," Agile principles and achievement of success in software development: A quantitative study in Brazilian organizations", Procedia Technology 16 (2014) 718 – 727, Elsevier.
15. Rita Cunha , Carla Sofia Pereira ; Jose Angelo Pinto," Agile software project: Proposal of a model to manage risks", Information Systems and Technologies (CISTI), 2013 8th Iberian Conference .

AUTHORS PROFILE

Mr. Anand Kumar Rai is Assistant Professor in Mumtaz PG College since 2000. His subject specialization in software engineering and artificial intelligence.Now he is doing his research work on the topic" Risk minimization plan for agile software using fuzzy logic" from Shri Ramswaroop memorial University, Deva Road ,Barabanki.

Dr. Shalini Agrawal is HoD of Department of Computer Science and Engineering at Sri Ramswaroop Memorial University. Her area of specialization is Cloud Computing, High Performance Computing and Software Engineering. She is also the chairperson of Digital Monitoring Cell of the University where she is taking initiatives to develop MOOC courses and streamline other University processes in accordance with Smart Campus directives of MHRD. She is a professional member of IEEE and CSI

Dr. Mazhar Khaliq did M.C.A. from Aligarh Muslim University and got Ph.D.(Computer Science) from Integral University, Lucknow. He has more than 20 years of teaching experiences. Now he is an Assistant Professor & Subject In-charge Department of Computer Science, Khwaja Moinuddin Chishti Urdu, Arabi-Farsi University.

Dr. Abhishek Kumaris Assistant Professor, Computer Science at Banaras Hindu University, BHU.He is Apple Certified Associate, Autodesk Certified & Adobe Certified Educator. His research area is Stereoscopy, 3D Animation, Image Processing, VR & AR, Multimedia, Game Technology, MOOCs, Graphics & Visual Effects.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication