



# A Neuro-Fuzzy based Approach to Software Quality Requirements Prioritization

Hossein Momeni  
Faculty of Engineering,  
Golestan University, Gorgan,  
Golestan, Iran

Homayoun Motameni  
Computer Department, Islamic  
Azad University of Sari,  
Sari, Iran

Mahtab Larimi  
Mazandaran University of  
Science and Technology  
Babol, Iran

## ABSTRACT

Requirements engineering is one of the important topics in software engineering and correct understanding of requirements can affect the total quality of the whole software system. Lack of considering or wrong identification of requirements not only leads to customers' dissatisfaction, but also leads to the software failure and increases the costs of development. Requirement prioritizing plays an important role in decision making process in subsequent phases and eliminates the complexities resulting from vague requirements. The aim is to find the right order of main requirements. Neuro-fuzzy system combines the ability of human reasoning (logic) through predetermined laws with the learning ability of neural networks and can apply the human expertise to find the total output value by using a set of laws stored in knowledge base. In this paper, we propose an effective approach to prioritizing software quality requirements based on neuro-fuzzy system. This paper shows that our proposed approach is more efficient in terms of both the time consumption and the learning ability compared to related approach.

## General Terms

Software Engineering.

## Keywords

Requirement Prioritization, Software Requirement, Neural Network, Fuzzy Logic.

## 1. INTRODUCTION

Requirement engineering is a subfield of software engineering that deals with real world aims, function and system software/hardware limitations. In fact, it includes a routine systematic process of problem solving activities and documentation that provide a suitable framework to analyse, define, measure the value and confirm the accuracy of the requirements. Therefore, requirement engineering is one of the most important topics in software field that is assumed the key factor in software projects success. Accurate identification, deployment and implementation of requirements affect the total quality of the software system [1]. Therefore, developing such systems has always been challenging. Lack of considering or wrong identification of requirements may lead to the software failure or may result in higher costs [2]. Therefore based on the importance of the users' demands and prioritizing methods, a set of users' needs is formed in order to manage the requirements. Prioritizing is a crucial and important step in a right decision making in order to plan for delivering different copies of the products [3-5]. The main

challenge is characterize of measures to prioritizing process including importance, cost, time, risk and penalty measures [6]. Generally, different measures interact with each other and affect each other positively or negatively. Usually decision making is performed among several options [1,6]. When the available options increase in number, decision making turns to be a difficult task. One of the methods for an accurate decision making is prioritizing the available options. The quality of the software product is determined by its ability to satisfy the users' or customers' demands [1,7]. However, gathering and identifying the accurate requirements and planning for delivery of a version compatible with a desired function is considered as the main phase in a product's success [5]. If the requirements are not implemented correctly, users refuse to use the product [6]. Any adjustment of wrong decisions in later stages and managing changes reduce the post delivery costs, however, fixing the deficiencies during the development process is more expensive and costly than making sound decisions in early stages [2]. The main challenge in prioritizing process is to choose right requirements among a huge set of different requirements. Prioritizing helps the developer to identify the requirements with higher value. Therefore, all the main interests, technical limits and all stakeholders' requirements should be considered in order to maximize the business value of the product. Moreover, prioritizing has other advantages like indentifying the problems of a requirement. Such problems can be due to its ambiguity [6].

The rest of the paper is organized as follows: Section 2 provides a brief background. Section 3 presents some notable related works on requirements prioritization. Section 4 presents our proposed prioritizing approach. Section 5 presents experimental results and finally Section 6 concludes the paper.

## 2. Background

### 2.1 Prioritizing Quality Requirements

Quality Function Deployment (QFD) method provides a tool for translating and transforming the customers' needs into proper technical requirements. The starting point for every QFD is the customers' need, which is most often referred to as an immeasurable factor. The method consists of five crucial parts including *customers' needs*, *technical requirements*, *ranking the importance of the requirements*, *the relationship matrix between the customer's needs and technical requirements* and *correlation matrix* which represents the relationship between technical requirements. The output of prioritizing based on function development method is an

ordered list of a set of technical requirements. Lack of customers' ability in producing clear and accurate needs (especially their quality requirements), misunderstanding the requirements by developers, facing lots of different requirements in first stages of projects, incompatibilities between quality factors and other requirements, inability of the current techniques in a quick requirement prioritizing and the influence of potential stakeholders are the reasons which make us improve QFD prioritizing method with the help of neural networks and fuzzy logic techniques.

## 2.2 Fuzzy Logic

Fuzzy sets and fuzzy logic theory has been introduced in 1965 by Lotfi Zadeh [8]. Fuzzy logic is a mathematical tool in order to deal with uncertainties. It also explains the ambiguous processes through human language and not correct-not incorrect statements. Fuzzy logic extends classic sets theory by considering uncertainties and gradual changes. Fuzzy arguments involve three main parts: fuzzification process, deduction from fuzzy rules and de-fuzzification process. Fuzzification is a process in which the desired terms are transformed into a fuzzy concept. At this stage the inputs and outputs are presented in language terms. Deduction and analysis of inputs involve defuzzification of conditions and laws.

Membership in classic logic sets is considered 1 or 0. But in fuzzy logic the elements of a fuzzy set has a degree of membership to that set. Degree of membership represents certainty of an element to that set. Therefore, the output of fuzzy logic membership function is a domain of [0,1]. Membership function acts as a specific function, which defines the fuzzy set. The aim is to allocate the membership degree of each element of the domain to its corresponding fuzzy set. Trapezoidal function, triangular function, Gaussian function, semi-exponential function, L, Gama, S and univalent functions are the most famous functions in demonstrating the membership degree in fuzzy sets. The concept of lingual variables (terms) or fuzzy variables were introduced instead of numbers. These variables are words and terms in a natural language whose aim is to translate the natural language into logical or numeral statements for approximate argumentation or logic. Generally, the results are determined through a set of if-then laws. These laws explain the possible situations. Due to the relative contribution in fuzzy sets, there exist the possibilities of several activities at the same time. The fuzzification process deals with fuzzy representation of defuzzy input values. It can be obtained through membership functions related to each fuzzy set at the law input area. The aim of the defuzzification process is to transform the fuzzy input into defuzzy values. There are different deduction methods for finding the approximate numerical value of the output, like min-max method, average method and square root.

## 2.3 Neural Network

Neural networks are the simplified neurobiological systems and consist of many neural elements which have the ability to learn and acquire knowledge [9]. Therefore, by presenting the recognized samples in a question, These can be thought to acquire the required knowledge.

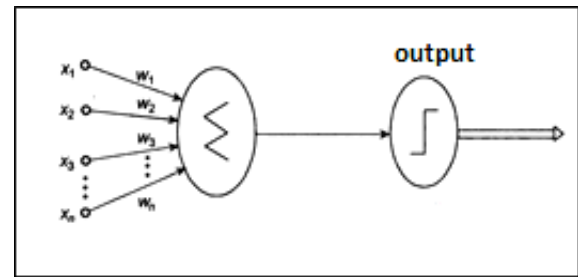


Fig 1. Neuron model

Figure 1 shows the behaviour of neuron. In this model  $x_1, x_2, x_3, \dots, x_n$  are the neuron input and  $w_1, w_2, w_3, \dots, w_n$  are the functions of each input weight. Neurons receive the inputs and put them together. If it exceeds from a certain point, the output will produced. The total received input in a neuron will be as follows:

$$I = w_1x_1 + w_2x_2 + \dots + w_nx_n = \sum_{i=1}^n w_i x_i$$

The neural architecture is divided into single layered feedforward networks, multiple layered feedforward network and recurrent networks. This paper uses the multiple layered feedforward architecture. The architecture of this network consists of one input layer, one output layer and one or multiple mid layers which are called hidden layers. The hidden layers are helpful in mid calculation before sending the inputs to output layer. In input layer, the linear transfer function is used. In hidden and output layers, the S shape functions are used.

The proposed approach uses back propagation neural network having regulative learning algorithm for prioritizing issue. Regulative learning algorithms are error-based and benefits from teacher; they can find the error at the end by comparing the reference and the obtained answer. In fact, at the input stage the answer is calculated through propagation the input models and after calculation of error amount, the network weights will be updated.

## 3. Related works

This section reviews many methods that have been presented in the area of prioritizing process in order to calculate the priority value of each requirement; like hierarchical analysis process [10, 11], binary search process, minimum spanning tree [11], grouping, ranking, top ten requirements [4], win-win approach, cost-benefit approach and etc. The above mentioned methods are used in prioritizing functional and non-functional(quality) requirements. The quality function development method which is called quality home is a method to prioritize the quality requirements of the project [12]. Although the implementation of the above mentioned methods are simple, they have not supported high level targets and they have not offer a scalable solution. They are suitable for small projects.

In [13], the authors have suggested a technique based on perceived value of each requirement, which is based on intelligent fuzzy logic. The process is time consuming and prone to errors duo to over reliance on expert system. The improved version of this technique [2,14] consists of three

phases including requirements elicitation and prioritizing stakeholders, prioritizing experts and professionals. This method is not intelligent and consumes a lot of time and finally prioritizing based on fuzzy logic, which reduces bias. In [2], the above research has been improved by applying neural networks at the second phase and the role of expert or software engineer had been eliminated.

Kahraman [15] have presented a QFD prioritizing method based on neuro-fuzzy approach for producing the optimal output from multiple standards decision making. In this method, the analytic hierarchy process (AHP) technique was used. They have prioritized the requirements by calculating the importance value of each technical requirement through applying the fuzzy laws. They did not consider the relationship between technical requirements. The requirements may be in contrast to each other or in the same line. The AHP method grows exponentially due to the number of the requirements, it is also a complicated method and prone to error because of pair comparison.

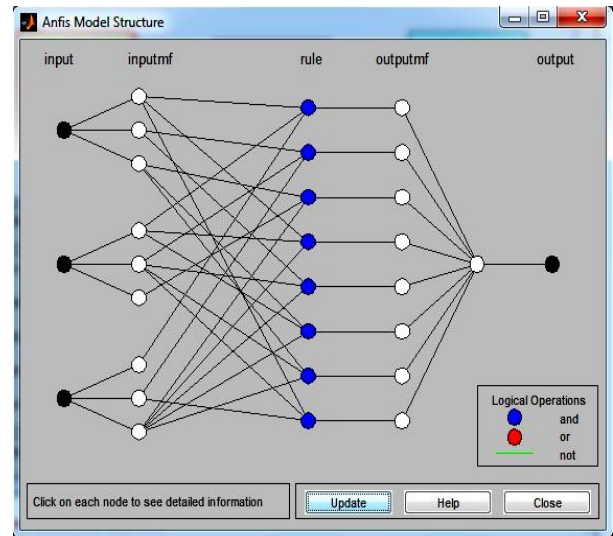
Sen *et al.* [16] suggested a method for prioritizing and selecting the optimal order of alternatives using the QFD fuzzy approach. They identified the technical criterion by categorizing the project’s requirements into three categories of environmental requirements, technical requirements and quality characteristics.

Given this background, the works have been reviewed so far in this paper have not offered a scalable solution when the number of requirements increases. Most of the techniques are time consuming and they are prone to errors. Also The effect of personal bias (stakeholders of the project) is visible on the results of prioritizing in most techniques.

#### 4. Proposed Approach

In this section our prioritization technique using neuro-fuzzy approach will be described. The neuro-fuzzy system is a combination of neural network with back propagation learning and fuzzy logic. The suggested neural network consists of triple layered feedforward network with three input nodes at the input layer, three hidden nodes in hidden layer and one output node at the output layer and uses LR fuzzy numbers. Figure 2 illustrates the overall layout of the implemented network in MATLAB programming environment.

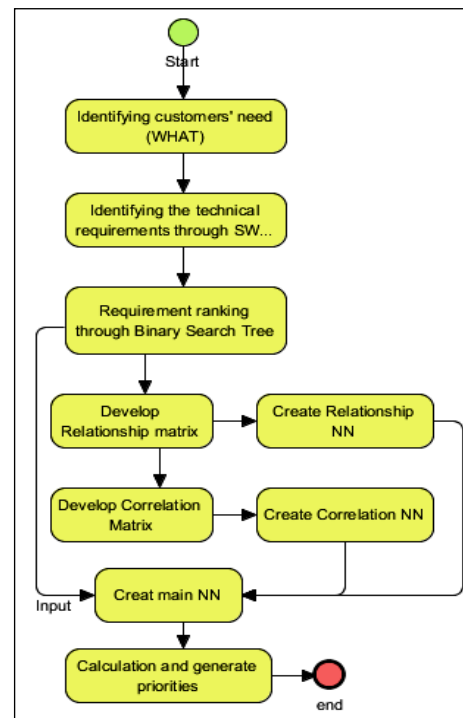
By applying QFD prioritizing method and neuro-fuzzy network, the inputs of the main network consist of *Ranking the customers*, *Relationship matrix*, and *Correlation matrix*. Input number 2 and 3 are resulted from similar neural networks which are called sub-networks of the main networks. The aim of the present paper is to achieve the optimal and rational order of technical requirements.



**Fig 2. Tree layered feedforward neural-network**

The step by step stages of the suggested prioritizing method is presented in Figure 3. The steps of our proposed process are described as follows:

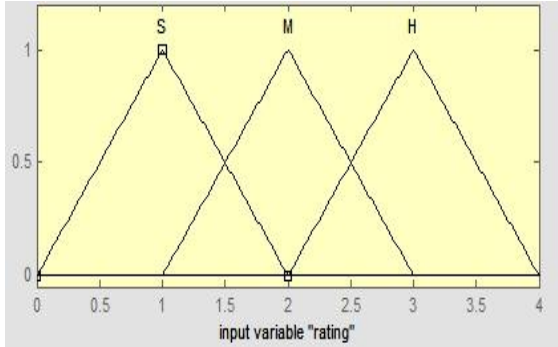
**Identifying customers’ need:** the first part of process deals with identifying requirements.



**Fig 3. Our proposed process**

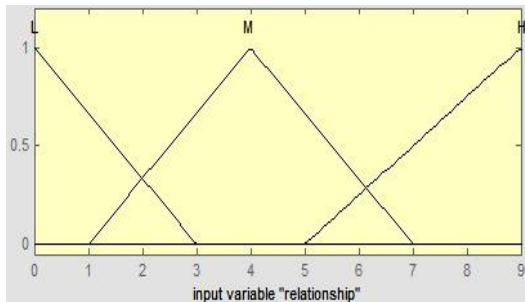
**Identifying the technical requirements:** after implementing the last phase, the system experts identify the required technical requirements in order to satisfy the customers’ need. Technical requirements mean the required quality characteristics. Categorizing the quality requirements and producing them is carried out based on one of the software quality models. The ultimate aim here is to achieve an ordered set of technical requirements.

**Requirement ranking:** Customers’ needs are ranked by binary search method in part five. The fuzzy membership function used in neuro-fuzzy system is illustrated as Figure 4 based on the resulted rank.



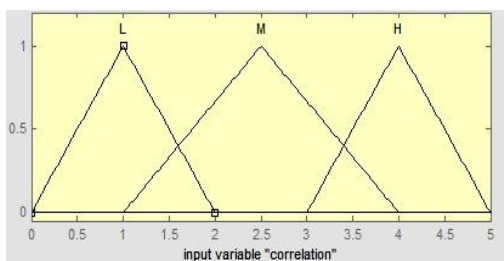
**Fig 4. Ranking membership function**

**The relationship matrix:** The relationship between the customers’ need and quality characteristics is determined by this type of matrix. Internal matrix is presented in three parts. There is a network consists of  $n$  input (the number of requirements) for each quality characteristics. Considering the obtained results, fuzzy membership function used in neuro-fuzzy system can be illustrated as Figure 5.



**Fig 5. Relationship membership function**

**Correlation matrix:** Because of interdependence between quality characteristics (positive interdependence and negative ones or discrepancies), the relationship between these characteristics should be determined in part four. For each quality characteristics, there is a network consists of  $m$  (the number of technical properties) and the difficulty degree of each characteristic is determined due to the relationship between them. Considering the obtained results, the fuzzy membership function used in neuro-fuzzy system is presented in Figure 6.



**Fig 6. Correlation membership function**

**Calculation:** finally, after determining the main networks’ inputs, training data is applied in networks and the synopsis weights will be updated. Then the given input is applied and an optimal output is produced. The output of the above mentioned neural network is an optimal order of technical characteristics required for implementation.

The above mentioned method has a lower overhead costs comparing the existing methods, because of using binary search tree approach and neural network.

## 5. Experimental Results

Sale system and financial system have been considered to evaluate the effectiveness of our proposed approach. Also the required data in this research have been collected through interview with managers, developers, field research, questionnaire and studying the existing documents in *System Software Group*. Then, our approach creates and ranks a list of requirements and expected functions of the system based on binary search tree method as shown in Table1.

**Table 1. Functional requirements and their ranks**

No	Functional Requirement	Rank
1	System should have the ability to issue pro-forma invoice.	0.72
2	System should have the ability to confirm the pro-forma invoice by customer.	0.5
3	System should have the ability to issue invoice.	3.82
4	System should have the ability to manage the orders.	1.7
5	System should have the ability to make sales contract.	1.88
6	System should have the ability to manage the payments.	2.78
7	System should have the ability to issue inventory receipt.	1.012
8	System should have the ability to do treasury operations (receive/payment cash and receive/payment cheque ).	3.67
9	System should have the ability to manage income and wages operations.	2.47
10	System should have the ability to do business accounting.	2.64
11	System should have the ability to manage costs and budget prediction based on the company’s long term targets and strategies.	3.03
12	System should have the ability to predict income based on market situation.	2.01
13	System should have the ability to get the reports.	3.12



A set of quality requirements in this system is listed as Table 3 such as replacement, installation, adjustment, changeability, testability, time, understanding, operability, recoverability, speed, fault tolerance, security, accuracy, traceability, interoperability and resource behaviour.

At this stage the quality home matrix will be developed. When the related values to each matrix elements are obtained, the fuzzy values of quality home matrix training data will be calculated. At this stage the neural networks will be trained by the above mentioned samples. Then, the weights are obtained after 200 repetitions as in Table 2.

**Table 2. Hidden layer weights**

Output of Hidden Layer Weight		Input of Hidden Layer Weight	
1.253	$W'_{00}$	0.678	$W_{10}$
		0.968	$W_{11}$
		0.65	$W_{12}$
2.942	$W'_{10}$	0.642	$W_{20}$
		0.347	$W_{21}$
		0.557	$W_{22}$
		0.569	$W_{30}$
2.569	$W'_{20}$	0.667	$W_{31}$
		0.411	$W_{32}$

In order to determine the priority of each identified quality requirements in quality home matrix, the corresponding numbers are given to the network. The results obtained from fuzzy inference for each quality parameter is shown as Table 3.

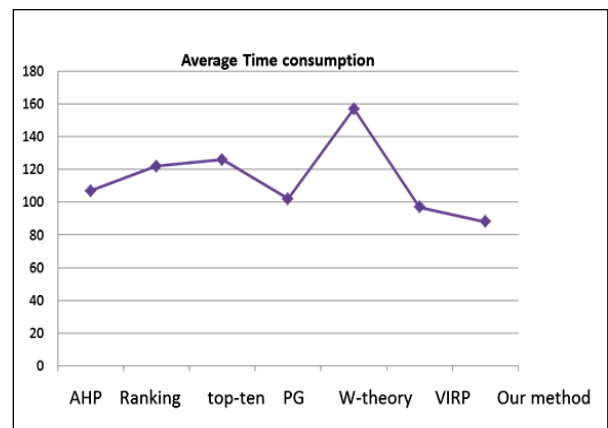
**Table 3. Calculated value and priority of each quality requirement**

Calculated Value	Quality Requirement	Priority
0.12	Replacement	Low
0.18	Installation	Low
0.23	Suitability	Low
0.14	Modifiability	Low
0.36	Testability	Medium
0.84	Time	High
0.27	Understand ability	Medium
0.88	Operability	High
0.07	Recoverability	Low

0.2	Fault-tolerancy	Low
0.77	Speed	High
0.98	Accuracy	High
0.92	Security	High
0.63	Traceability	High
0.74	Interoperability	High
0.41	Resource behaviour	Medium

Considering the results obtained from fuzzy inference in our case study on sales system and financial system, our approach shows that the quality parameters like accuracy, security, interoperability, traceability, understandability, time and speed of doing tasks have high importance and there should be great attention to them.

Our experimental results as Figure 7 show that the speed of calculation and output production by network is 2.43 with 8% faster than the QFD prioritizing method approximately. Moreover, the average time consumption in our proposed approach decreased about 9% compared to the presented methods in [2]. Because of the major part of consumed time is allocated to the pre-output production activities like negotiation, ranking and network training, therefore, our proposed approach has learning ability and human inference ability using neuro-fuzzy system and can deliver greater results comparing the existing research.



**Fig 7 . Time consumption comparison**

## 6. Conclusion

Requirement engineering is one of the most important topics in the software engineering. Nevertheless, its importance has been considered in recent years and there have been activities to overcome its shortcomings. One of the crucial steps in requirements engineering field is prioritizing for choosing the right set of requirements from the whole requirements. Prioritizing is one of the most important and crucial steps in planning, identifying the main requirements and delivering different versions of a product. In fact, it makes the development team able to rank and implement the customers' needs based on standards and important aspects. The prioritizing process and determining which requirements



should go to design and implementation phases first, has always been challenging and full of ambiguity and uncertainties; the stakeholders' opinion has always played an important role here. This paper have presented a new method, which is able to provide an optimal order of quality characteristics required for implementing the products that satisfy the customers. Our experimental results showed lower cost and time consumption compared to the related approaches. In addition, this paper showed that our approach has learning ability and human inference ability using neuro-fuzzy system and can deliver greater results compared to other approaches.

## 7. REFERENCES

- [1] Svensson, R. B., T. Gorschek, et al. 2011. Prioritization of Quality Requirements: State of practice in eleven companies. 19th IEEE International Conference on Requirements Engineering.
- [2] Babar, M. I., M. Ramzan, et al. 2011. Challenges and Future Trends in Software Requirements Prioritization. IEEE International Conference on Computer Networks and Information Technology (ICCNIT).
- [3] Wiegers, K. 1999. First Things First: Prioritizing Requirements. *Software Development* Vol. 7 No.9, pp. 11-19.
- [4] Berander, P. and A. Andrews 2005. Requirements Prioritization. *Engineering and Managing Software Requirements*, Springer, pp. 69-94.
- [5] Svensson, R. B., M. Host, et al. 2010. Managing Quality Requirements: A systematic review. 36th IEEE EUROMICRO Conference on Software Engineering and Advanced Applications.
- [6] Johansson, E., A. Wesslén, et al. 2001. The Importance of Quality Requirements in Software Platform Development: A Survey. 34th IEEE Annual Hawaii International Conference on System Sciences.
- [7] Lehtola, L., M. Kauppinen, et al. 2004. Requirements Prioritization Challenges in Practice. *Product Focused Software Process Improvement*, Springer, pp. 497-508.
- [8] Zadeh, L. A. 1965. Fuzzy Sets. *Information and Control*, Vol. 8 No.3, pp.338-353.
- [9] Rajasekaran ,S. and G. V. Pai 2003. *Neural Networks, Fuzzy Logic and Genetic Algorithm. Synthesis and Application*. PHI Learning Pvt. Ltd.
- [10] Martínez Carod, N. and A. Cechich 2005. Classifying Software Requirement Prioritization Approaches. XI Congreso Argentino de Ciencias de la Computación.
- [11] Chan, K. Y., C. Kwong, et al. 2012. Determination of Importance of Customer Requirements Using the Fuzzy AHP Method. *Computational Intelligence Techniques for New Product Design*, Springer, pp. 59-77.
- [12] Chan, L.-K. and M.-L. Wu 2002. Quality Function Deployment: a Literature Review. *European Journal of Operational Research*, Vol. 143, No. 3 pp. 463-497.
- [13] Lubars, M., C. Potts, et al. 1993. A Review of the State of the Practice in Requirements Modeling. IEEE International Symposium on Requirements Engineering.
- [14] Ramzan, M., M. A. Jaffar, et al. 2010. Value based Intelligent Requirement Prioritization (VIRP): Expert Driven Fuzzy Logic based Prioritization Technique. *International Journal of Innovative Computing, Information and Control (IJICIC)*.
- [15] Kahraman, C. 2008. *Fuzzy Multi-Criteria Decision Making: Theory and Applications with Recent Developments*, Springer.
- [16] Şen, C. G. and H. Baraçlı 2010. Fuzzy Quality Function Deployment based Methodology for Acquiring Enterprise Software Selection Requirements. *Expert Systems with Applications*, Vol. 37, No.4 pp. 3415-3426.