

## Analyzing Imprecise Requirements Using Fuzzy Logic

Rajwinder Kaur<sup>1</sup> Manisha Bhardwaj<sup>2</sup> Neha Malhotra<sup>3</sup>

Department of computer science and engineering

Lovely professional university, Punjab, India<sup>1,3</sup>

Assistant professor in department of computer science and engineering

Lovely professional university, Punjab, india<sup>2</sup>

### Abstract

*The systems and software development industry is characterized by a paradigm of project failure. A major contributor to project failure is imprecise requirements which are the result of the failure to spend the time at the beginning of the project to clearly define the product requirements. Every requirement has to be verified before it is based lined. There is in fact a long list of factors which can leads to gathering of imprecise requirements. This paper presents a fuzzy logic based tool to estimate the effects of those factors on software requirements.*

### 1. Introduction

Software engineering is a systematic, disciplined, quantifiable approach to the development, maintenance of the software product (1). Many of the Software development phases are highly communication-intensive activity that involves, at minimum, analysts, architects, developers, testers, business stakeholders, and end users. Among all the development phases and products, requirements are key ingredient in the process of designing and realizing any systems. Without clearly understanding requirements and their proper management, large projects are likely to fail and could have very high maintenance cost [2]. So to analyze the imprecise requirements a fuzzy logic based approach is used. Fuzzy logic deals with uncertainty in engineering by attaching degree of certainty to answer to a logical question. Fuzzy logic is based upon uncertainties where there is an inherent impreciseness. It has been used with great success to control machines and consumer products. Fuzzy Logic system is simple to design and can be understood and implemented by a non-specialist in the control theory. Fuzzy logic has rapidly become one of the most successful of today's technologies for developing sophisticated control systems. The reason for which is very simple. Fuzzy logic addresses such

applications perfectly as it resembles human decision making with an ability to generate precise solutions from certain or approximate information [3]. It provides mathematical tools for solving and working out approximate reasoning processes when having to deal with imprecise, uncertain, and vague data. [4]

### 2. Related Work

Defining product requirements in a more precise manner ultimately increases the chances of product's success, but on the other side defining the same in an imprecise manner will leads to the failure of the product. Requirement analysis phase ensures the precision of the product requirements. There is a long list of factors which leads to gathering of imprecise requirements. Successful software projects are built on a foundation of well-understood requirements. However, many development organizations get caught in traps that prevent them from effectively collecting, documenting, or managing their requirements. Karl weigers describes ten requirement traps to avoid. which includes: confusion over requirements, Inadequate customer involvement, Vague and ambiguous requirements, Unprioritized requirements, Building functionality no one uses, Analysis paralysis, Scope creep, Inadequate requirements change process, Insufficient change impact analysis, Inadequate requirements version control.[7] To reduce the chances of software project failure, there is need to study all the factors which can have direct or indirect harmful effect on the success of the software project. There are so many contextual factors that have an effect on the software development process. [8] From the literature survey [9,10,11,12] I had choose those factors which have very strong adverse impact on the software requirements .These factors are given in Table 1.

Table 1  
Input factors

Sr. no.	Factor name
1	Unclear goals & objectives of customers
2	Inadequate conflict management
3	Communication gap
4	Undetected scope creep
5	No expert usage show up
6	Budget constraints
7	Lack of contextual detail
8	Inappropriate skills of analyst
9	Unrealistic expectations of customers
10	Poor planning

All the above factors lead to the imprecise requirements. Human communication involved in the requirement gathering process, is responsible for the uncertainty and ambiguity in requirements. Fuzzy logic is capable to handle ambiguous data. It is well known and can provide human-like expertise. [5] Fuzzy logic has become an important tool for number of different applications ranging from the control of engineering system to artificial intelligence. [6]

**3. Proposed work**

A fuzzy logic based tool is proposed to analyse imprecise requirements. In process of doing this analysis using fuzzy logic was to identify important factors which lead to gathering of imprecise requirements. From the literature survey, we found list of factors that leads to the gathering of imprecise requirements. Even after the deep discussion with the software engineers, we considered these factors for imprecise requirement analysis. Although the impact of these factors can be vary from organization to organization. We constructed rule base using 100 different rules with guidance from experts. With use of MATLAB a graphical user interface based tool is developed according to the rule base. The tool will take factors arise during the requirement gathering phase as input then give us output that what will be its impact on the requirements or in a more simplified way what would be the percentage of possibility of occurrence of imprecise requirements in existence of those factors.

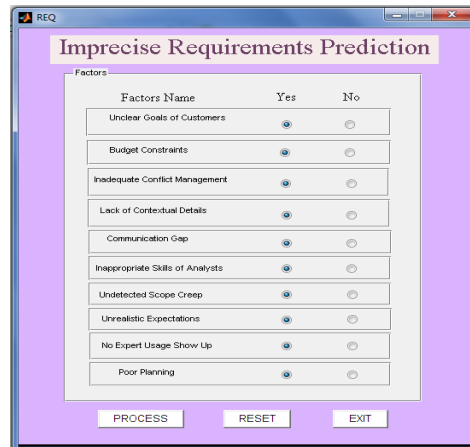


Figure 1. GUI based proposed tool

**3.1 Design Methodology**

As we discussed earlier that the impact of these factor may vary from organization to organization depending upon the factors considered at different environments. The impact levels are categorized into five levels. The impact can be “Very Low”, “Low”, “Medium”, “High”, and “Very High”. Inputs are represented by fuzzy sets to design fuzzy inference system. Membership functions are used to represent the fuzzy sets. One hundred Rules are designed after representing inputs and outputs of fuzzy sets in membership functions. Some rules are:

1. If (Unclear Goals And Objectives Of Cust is false) and (Inadequate Conflict Management is true) and (Communication Gap is true) and (Undetected Scope Creep is true) and (No Expert Usage Show Up is false) and (Budget Constraints is true) and (Lack Of Contextual Detail is true) and (Inappropriate Skills Of Analyst is true) and (Unrealistic Expectations Of Customers is false) and (Poor Planning is true) then (Imprecise Requirements is VH) (1)

2.If(Unclear Goals And Objectives Of Cust is true) and (Inadequate Conflict Management is true) and (Communication Gap is false) and (Undetected Scope Creep is true) and (No Expert Usage Show Up is true) and (Budget Constraints is true) and (Lack Of Contextual Detail is true) and (Inappropriate Skills Of Analyst is true) and (Unrealistic Expectations Of Customers is true) and (Poor Planning is true) then (Imprecise Requirements is VH) (1)

.....

100.If(UnclearGoalsAndObjectivesOfCust is false) and (InadequateConflictManagement is true)and (CommunicationGap is false) and (UndetectedScopeCreep is false) and (NoExpertUsageShowUp is true) and

(BudgetConstraints is true) and (LackOfContextualDetail is false) and (InappropriateSkillsOfAnalyst is true) and (UnrealisticExpectationsOfCustomers is true) and (PoorPlanning is false) then (ImpreciseRequirements is M) (1)

Different types of membership functions for designing the fuzzy inference system in MATLAB are used. For input factors, we used Z-shaped and S-shaped built-in membership functions. For output factors, five membership functions are used in which four functions are of triangular type and one is trapezoidal type. The Z-shaped membership function used is given in Eq. 1[15]

$$f(x, 0, 1) = \begin{cases} 1, & x \leq 0 \\ 1 - 2 \left( \frac{x-0}{1-0} \right)^2, & 0 \leq x \leq \frac{0+1}{2} \\ 2 \left( \frac{x-1}{1-0} \right)^2, & \frac{0+1}{2} \leq x \leq 1 \\ 0, & x \geq 1 \end{cases} \quad (1)$$

The trapezoidal-shaped membership function is given in Eq. 2[15]

$$f(x, 0.669, 0.892, 0.9, 1) = \begin{cases} 0, & x \leq 0.669 \\ \frac{x - 0.669}{0.892 - 0.669}, & 0.669 \leq x \leq 0.892 \\ 1, & 0.892 \leq x \leq 0.9 \\ \frac{1 - x}{1 - 0.9}, & 0.9 \leq x \leq 1 \\ 0, & 1 \leq x \end{cases} \quad (2)$$

### 3.2 Implementation Procedure

Fuzzy inference system as given in figure shows the mapping of inputs and outputs. The names of the processing methods and inputs can be changed.

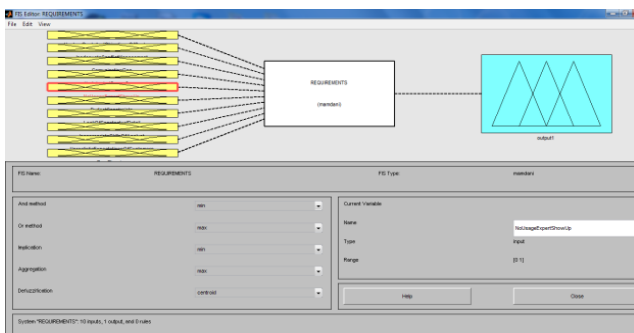


Figure 1. fuzzy inference system

Membership function editor as given in figure 3 shows input and output variables along with their range and probability of their occurrences. In given snapshot

an input factor has corresponding output with having probability of occurrence i.e. true and false.

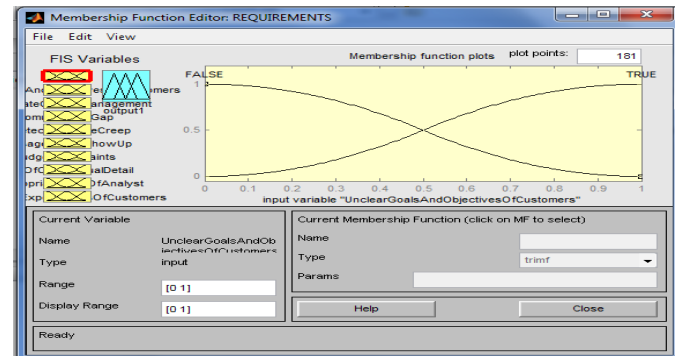


Figure 3. Membership function editor

Rule editor given in fig 4 helps to add, change and delete rules. One hundred rules are edited in rule base. Rule base is designed on the basis of expert opinion.

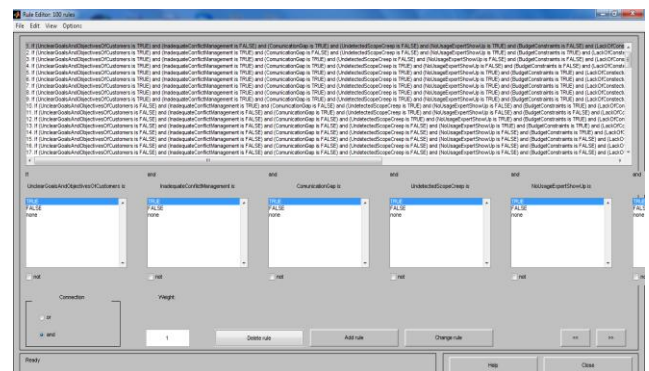


Figure 4. rule editor

We take, for instance, that if some factors (unclear goals of customers, communication gap, and inappropriate skills of analyst) arise during the requirement gathering phase then what will be its impact on the requirements or in a more clear way what would be the percentage of possibility of occurrence of imprecise requirements. The tool helps to predict the impact of various factors which leads to gathering of imprecise requirements. In the fig 5a, input is provided for a sample case and in fig 5b, the tool gives the result that the possibility of imprecise requirements occurrence will be 50.4642%

Factors Name	Yes	No
Unclear Goals of Customers	<input checked="" type="radio"/>	<input type="radio"/>
Budget Constraints	<input type="radio"/>	<input checked="" type="radio"/>
Inadequate Conflict Management	<input type="radio"/>	<input checked="" type="radio"/>
Lack of Contextual Details	<input type="radio"/>	<input checked="" type="radio"/>
Communication Gap	<input checked="" type="radio"/>	<input type="radio"/>
Inappropriate Skills of Analysts	<input checked="" type="radio"/>	<input type="radio"/>
Undetected Scope Creep	<input type="radio"/>	<input checked="" type="radio"/>
Unrealistic Expectations	<input type="radio"/>	<input checked="" type="radio"/>
No Expert Usage Show Up	<input type="radio"/>	<input checked="" type="radio"/>
Poor Planning	<input type="radio"/>	<input checked="" type="radio"/>

Figure 5(a). GUI with inputs

PERCENTAGE Imprecise Requirements 50.4642 %

Figure 5(b). Output

#### 4. Conclusion

In Software Engineering, researchers are still working to get the more knowledge of how to gather more precise requirements from the customers. There are various factors that can arise on requirement gathering phase and they ultimately leads to gathering of imprecise requirements. There are some factors which have more adverse impact rather than others. This paper discusses list of factors associated with requirement gathering phase and their impact on software requirements is predicted by using fuzzy logic based GUI tool. As researchers are working in the area of risk associated with requirement gathering phase, and as more and more data is collected, in future an expert system can be designed which can analyze imprecise requirements more efficiently .

#### 5. References

- [1] Softwareengineering.[http://en.wikipedia.org/wiki/Software\\_engineering](http://en.wikipedia.org/wiki/Software_engineering), last accessed 12-08-2011
- [2] Shradhanand, Amarjeet Kaur, Dr. Satbir Jain, "use of fuzzy logic in software development", Volume VIII, No. 2,2007, pp: 238-244

- [3] Boehm, B. W., "Software Engineering Economics" Prentice Hall, 1981.
- [4] Fuzzylogic,[http://www.doc.ic.ac.uk/~nd/surprise\\_96/journal/vol4/sbaa/report.intro.html](http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/sbaa/report.intro.html), last accessed 5-9-2011
- [5] Prajakta Chandrakant Dhote "handling ambiguous data during requirements verification using fuzzy logic", International Journal of Computer Science and Communication Vol. 2, No. 1, 2011,pp. 105-107
- [6] Norbert Sram "Practical Application of Fuzzy Logic from Software Engineering Point of View", Vol. 2, No. 1, 201, pp 285-291
- [7] Karl Wieger's 10 Requirements Traps to Avoid, <http://www.processimpact.com/articles/reqtraps.html>, last accessed 18-3 2012
- [8] Alexandre Bern, Satya Jaya Aparna Pasi, Uolevi Nikula, Kari Smolander "Contextual Factors Affecting the Software Development Process – An Initial View", pp:1-11
- [9] whyITprojectfail?,[http://www.projectperfect.com.au/info\\_it\\_projects\\_fail.php](http://www.projectperfect.com.au/info_it_projects_fail.php) last accessed 18-1-2012 (2012,
- [10] badrequirements,<http://alastair.cockburn.us/Bad+requirements+may+not+be+root+causes+of+project+failures> last accessed 19-3-2012
- [11] Trainingsaminars,<http://www.processimpact.com/seminars.shtml>,last accessed 19-1-2012
- [12] ClassicMistakesEnumerated,<http://www.stevemcconnell.com/rdenum.htm>, last accessed 19-1-2012
- [13] john yen & w. amos tios " A systematic trade off analysis of conflicting imprecise requirements"
- [14] Jonathan Lee , Nien-Lin Xue, Kuo-Hsun Hsu,Stephen J. Yang, "Modeling imprecise requirements with fuzzy objects" Information Sciences an international journal,1999 pp:101-119
- [15] Namrata Kapoor, Nitin Bhatia, Sangeet Kumar, 'software risk analysis using fuzzy logic', international journal of computer information system , vol. 2, No.2, 2011