Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2015, 2(7): 70-75

Research Article ISSN: 2394 - 658X

A Fuzzy Logic Based Approach for Selecting the Software Development Methodologies Based on Factors Affecting the Development Strategies

Rituraj Jain

Department of Computer Science Engineering, Vyas Institute of Engineering and Technology, Jodhpur, India
jainrituraj@yahoo.com

ABSTRACT

Project management discipline has showed its impact on software industry and has been now established as core concept for organizing, innovative as well as strategic endeavors. Software development organization follows some process when developing a software product. Software Methodology plays a key role software development process on which the process is based. The particular methodology can significantly affect overall life cycle cost associated with a software product. The selection of the methodology is also based on the some predefined basic characteristics of the project under design. In this paper a fuzzy logic based approach is presented to assist organizations in making the decision regarding which software development methodology to select from Rational Unified Process (RUP), Rapid Application Development (RAD) and Extreme Programming (XP) by considering the associated initial characteristics with the project. Taking the correct decision regarding what software development methodology is to be use can lead in the first step towards the success of the project.

Key words: Fuzzy Logic, Rapid Application Development, Rational Unified Process, Software Methodology

---

INTRODUCTION

Project success is defined as based on the three basic dimensions of project which are delivery within schedule, within cost, and quality (satisfaction of customer based on the requirements). Software development methodologies have an impact on the project success while talking about all three dimensions mentioned above. Wide range of software development methodologies have been elaborated, so selection of suitable one from these many available methodologies is the tedious task which must be carried out during the inception of the development process and will show great impact on the overall success of the project. False decision of the selection of unsuitable organizational software development process can increase the risk which in turn may be result as a project failure. While an organization follows documented processes for software development, there is a greater chance that this organization will be able to consistently deliver successful projects. The activities required to produce software include project planning, development of requirements, development of specifications, architecture design, software configuration, software development, testing, documentation, training, support, and maintenance. The success rate of software development projects can be increased by using a methodology that is adequate for the specific characteristics of those projects. Project based characteristics must be considered for taking the initial decision about the methodology to be used for the specific project.

There is always certain amount of uncertainty involved in undertaking software engineering activities since these relate in many ways to software projects. Software projects have several risks and assumptions associated with them. Such kind of problems can be better solved using the fuzzy logic. Selection of suitable methodology for the ongoing project is also comprises of some of the uncertainty. Fuzzy logic approach can be help to take a decision for selection of suitable methodology during the initial phase of the software development and overcome the uncertainly problem.

The goal of this research is to help organizations identify what software development methodology should be used for specific projects to facilitate delivering projects within schedule, within cost and meeting all the project requirements. During this research only three methodologies wiz Rational Unified Process (RUP), Rapid Application Development (RAD) and Extreme Programming (XP) were considered.

SOFTWARE DEVELOPMENT METHODOLOGIES

Software development methodologies have been evolved right from the evaluation of programming languages. These methodologies guides development process and usually divides the development process into phases, and for each phase, the developer has to apply the guidelines that the selected methodology provides for that phase. These methodologies enhanced the software development process and the number of failed and incomplete software project ratio was reducing by a big numbers [1]. Avison et al defined a software development methodology as 'it is a collection of procedures, techniques, tools and documentation aids which will help the systems developers in their efforts to implement a new information system' [2].

Following methodologies have been evolved and applied to developing of the software's: Waterfall, Spiral, Feature Driven Development (FDD), Joint Application Development (JAD), Lean Development (LD), Agile Software Development, Dynamic Systems Development Model (DSDM), Rapid Application Development (RAD), Extreme Programming (XP), Rational Unified Process (RUP) etc.

Rapid Application Development (RAD) model is 'high-speed' model which was developed to respond to the need to deliver systems very fast. It adapts many steps from waterfall model in which rapid development is achieved by using component based construction approach. [3-4]. RAD works to reduce overall project risk by deconstructing the project into smaller segments during which prototypes are created and refined. RAD supports changes during the development process – change can be introduced at any time in the cycle. RAD is based on iterative prototyping, user participation, and the use of tools that facilitate rapid development. The foundation of RAD is to satisfy customer business requirements in a short amount of time [5]. Phases of RAD Model are Communication, Planning, Modeling (Business modeling, Data modeling, Process modeling) and Construction. Risk related requirements, delivery deadlines involved during the implementation of RAD methodology.

Extreme Programming (XP) is based on values of simplicity, communication, feedback, and courage. It works by bringing the whole team together in the existence of simple practices, with enough feedback to enable the team to see what they are doing and where they are. In XP, every member of the project is an integral part of the whole team and plays a specific role [6]. The main philosophies of XP include planning by creating user stories, managing small releases with continuous integration and refactoring, using a metaphor for the project, having developers work using pair programming, and emphasis on collective ownership of the project. XP projects work through four phases: planning, designing, coding, and testing. Planning involves designing the user stories, creating the project schedule, planning the iteration for each cycle. The main purpose of it is to deliver what the customer needs, at the time it is needed. In addition to this, one of the main reasons of its success is its ability to accept changes at anytime during the development. XP also emphasizes teamwork; experiences from all stakeholders are employed to meet the specific goals, and within the given constraints [7]. Factors which affect the XP are team size, physical environment, communication and coordination between team members, project size, interdependencies between modules.

RUP is another software development approach which ends up producing & delivering a quality software product which is aligned with the requirements of its end users in a controlled schedule and finances. It is also a process framework which is followed and enriched in order to best qualify the requirements of the organization [8]. It is an iterative and incremental software development approach in which a project is broken down into various modules which is dev eloped one after the other. RUP establishes four phases of development which are inception, elaboration, construction, transition. Each phase is organized into a number of separate iterations that must satisfy defined criteria before the next phase is undertaken. Each iteration starts with a milestone to achieve and similarly ends up with a deliverable which takes the product into the final stages gradually. Every iteration of RUP is constituent of various steps like requirements gathering, requirements analysis, implementation, assimilation and test phase [9].

FUZZY LOGIC

Fuzzy logic is originate from the Fuzzy set theory and can be classify as an extensive form of the classical logical system. These techniques have found mass appeal in various computational and manufacturing engineering domains. In numerous problems of different domain, fuzzy logic has been successfully applied and also gave the useful results [10-12]. In the software engineering domain also fuzzy logic was applied in various development phases and on the artifacts released through these phases. A fuzzy model structure can be represented by a set of fuzzy If-Then rules [13]. It serves as a conceptual framework which works to cater to the uncertainty in the knowledge representation. In the fuzzy logic intermediate values will be defined between conventional evaluations like yes or no, true or false, good or bad, low – medium - high, etc and these notions can be formulated mathematically and processed by computers [14].

Fuzzy logic based approach, to solve any problem, is divided into three steps which are Fuzzification, Development of Fuzzy Rules, and Defuzzification. Fuzzification process is carried out by developing membership

functions generated from different input sources. The fuzzy rule base is usually constructed from the experience of the decision maker, which will be applied over fuzzy input and arriving at the fuzzy output. They encode knowledge about a system in statement of the form:

IF (x_1 is X_1 , x_2 is X_2 ,……..x_n is X_n) THEN (y_1 is Y_1, y_2 is Y_2,……..y_n is Y_n)

where linguistic variables xi, yj take the value of fuzzy sets Xi and Yj respectively [15].

Defuzzification is the reverse procedure of the fuzzification and used to take crisp decision by applying membership functions like Max membership principle, Centroid Method (Center of Gravity Method), Weighted Average Method, Mean–max Membership etc. on the fuzzy outputs and used to represent them in a single scalar quantity [15].

FUZZY BASED PROPOSED APPROACH

While taking the decision of appropriate methodologies for the software development, few characteristics and factors must be considered by the project manager. Following are the few characteristics or factors which were under study in this research:

- Clarity of Requirements (COR)
- Accurate Initial Estimations (AIE)
- Changes In Requirements (CIR)
- Software Criticality (SC)
- System Complexity (SyC)
- High Reliability Requirements (HRR)
- Reusable Components (RC)
- Tight Project Schedule (TPS)

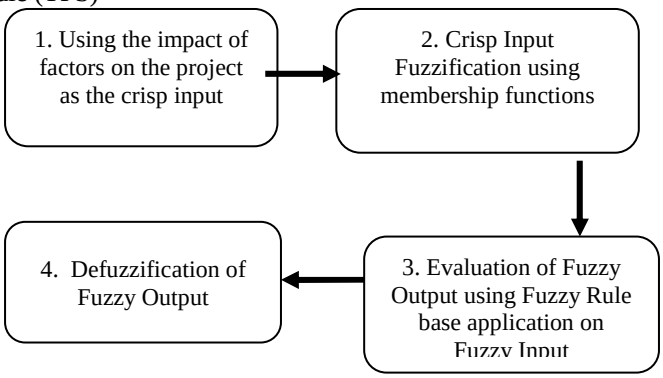


Fig. 1 Flowchart for Fuzzy Approach

All these factors are uncertain and can be expressed in qualitative terms like LOW, MEDIUM, and HIGH. Therefore it is needed to apply fuzzy based approach to quantify these qualitative terms by deriving suitable membership functions to find suitable methodology for software development process for current project. The flowchart of the used fuzzy approach is shown in Fig 1. Mamdani's fuzzy inference method is used in the proposed approach. The first step is to take the inputs and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions. The factors, can be interpreted as linguistic variables, are given as input to the Fuzzy Inference System (FIS). The methodology choice (RAD, XP, RUP) are the linguistic variables for output as shown in Fig 2.

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The membership functions can be of different types such as triangular membership functions and trapezoid membership function depending upon the experience of the decision maker/project manager. Using the impact of factors on the project and output preference of methodology and their evaluation in qualitative terms such as low, medium, high as the crisp input, membership functions can be generated by fuzzifing them as shown in Fig 3 and Fig 4.

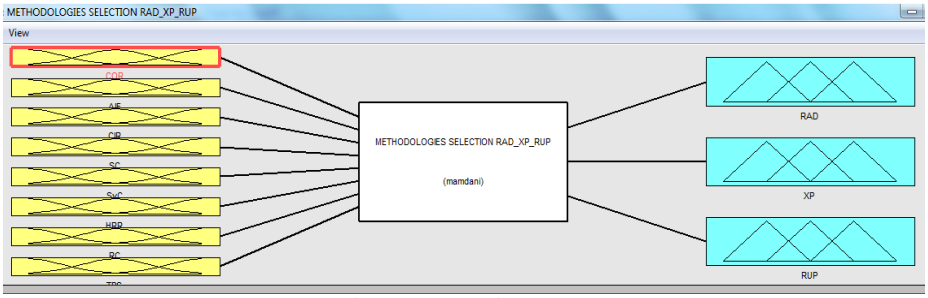
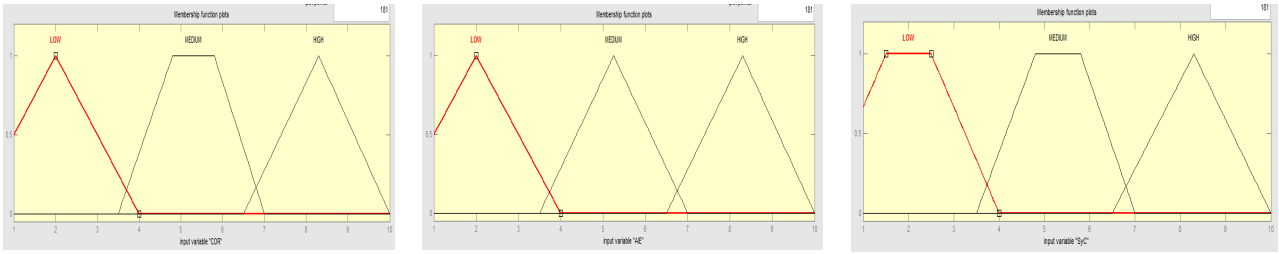


Fig. 2 Fuzzy Inference System for Proposed Approach



(a) (b) (c)

Fig. 3 Membership Function of Inputs (a) Clarity of Requirements (COR), Software Criticality (SC), High Reliability Requirements (HRR), Reusable Components (RC), Tight Project Schedule (TPS) (b) Accurate Initial Estimations (AIE), Changes In Requirements (CIR), (c) System Complexity (SyC)

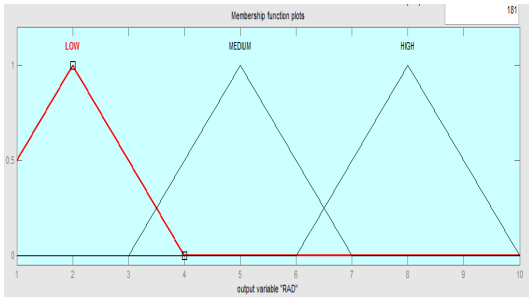


Fig. 4 Membership Function of Outputs Rapid Application Development (RAD), Extreme Programming (XP), Rational Unified Process (RUP)

Membership Functions and their parameter values for all these variables used for the experiment are as follows:

1. If (COR is HIGH) or (AIE is HIGH) or (CIR is MEDIUM) or (SC is HIGH) or (SyC is HIGH) or (HRR is HIGH) or (RC is HIGH) or (TPS is HIGH) then (RAD is MEDIUM)(XP is LOW)(RUP is HIGH) (1)
2. If (COR is HIGH) or (AIE is MEDIUM) or (CIR is LOW) or (SC is MEDIUM) or (SyC is HIGH) or (HRR is MEDIUM) or (RC is HIGH) or (TPS is MEDIUM) then (RAD is LOW)(XP is LOW)(RUP is HIGH) (1)
3. If (COR is HIGH) or (AIE is HIGH) or (CIR is MEDIUM) or (SC is HIGH) or (SyC is HIGH) or (HRR is HIGH) or (RC is MEDIUM) or (TPS is MEDIUM) then (RAD is LOW)(XP is LOW)(RUP is HIGH) (1)
4. If (COR is HIGH) or (AIE is MEDIUM) or (CIR is MEDIUM) or (SC is HIGH) or (SyC is MEDIUM) or (HRR is HIGH) or (RC is MEDIUM) or (TPS is HIGH) then (RAD is LOW)(XP is MEDIUM)(RUP is HIGH) (1)
5. If (COR is MEDIUM) or (AIE is MEDIUM) or (CIR is HIGH) or (SC is MEDIUM) or (SyC is MEDIUM) or (HRR is HIGH) or (RC is MEDIUM) or (TPS is MEDIUM) then (RAD is HIGH)(XP is MEDIUM)(RUP is LOW) (1)
6. If (COR is MEDIUM) or (AIE is MEDIUM) or (CIR is HIGH) or (SC is MEDIUM) or (SyC is MEDIUM) or (HRR is HIGH) or (RC is LOW) or (TPS is LOW) then (RAD is HIGH)(XP is LOW)(RUP is LOW) (1)
7. If (COR is MEDIUM) or (AIE is MEDIUM) or (CIR is HIGH) or (SC is HIGH) or (SyC is MEDIUM) or (HRR is MEDIUM) or (RC is LOW) or (TPS is LOW) then (RAD is HIGH)(XP is MEDIUM)(RUP is LOW) (1)
8. If (COR is MEDIUM) or (AIE is MEDIUM) or (CIR is HIGH) or (SC is MEDIUM) or (SyC is MEDIUM) or (HRR is MEDIUM) or (RC is MEDIUM) or (TPS is LOW) then (RAD is HIGH)(XP is LOW)(RUP is LOW) (1)
9. If (COR is LOW) or (AIE is LOW) or (CIR is HIGH) or (SC is MEDIUM) or (SyC is LOW) or (HRR is LOW) or (RC is MEDIUM) or (TPS is HIGH) then (RAD is MEDIUM)(XP is HIGH)(RUP is LOW) (1)
10. If (COR is LOW) or (AIE is MEDIUM) or (CIR is HIGH) or (SC is MEDIUM) or (SyC is LOW) or (HRR is LOW) or (RC is MEDIUM) or (TPS is MEDIUM) then (RAD is LOW)(XP is HIGH)(RUP is LOW) (1)
11. If (COR is LOW) or (AIE is MEDIUM) or (CIR is HIGH) or (SC is MEDIUM) or (SyC is LOW) or (HRR is HIGH) or (RC is MEDIUM) or (TPS is HIGH) then (RAD is MEDIUM)(XP is HIGH)(RUP is LOW) (1)
12. If (COR is LOW) or (AIE is LOW) or (CIR is HIGH) or (SC is MEDIUM) or (SyC is MEDIUM) or (HRR is LOW) or (RC is MEDIUM) or (TPS is MEDIUM) then (RAD is LOW)(XP is HIGH)(RUP is LOW) (1)

Fig. 5 Fuzzy Rule base for finding the appropriate software development methodology

After generating the membership values of all the factors, fuzzy rule base, as shown in Fig 5, is constructed to arrive at the fuzzy output. Also all the 12 rules are listed in the table 2, where H, M, L stands for HIGH, MEDIUM, and LOW qualitative values.

Fuzzified input gives the degree to which each part of the antecedent has been satisfied for each rule. Here we have eight inputs on which the fuzzy operator OR is applied to obtain one number that represents the result of the antecedent for that rule. This number will then be applied to the output function to get suitability of each of the methodology (RAD, XP and RUP) for the current scenario.

The Rule Viewer displays a roadmap of the whole fuzzy inference process. Form the rule viewer, as shown in Fig 6, we can study the relationship of specific parameter to output and analyze the change in output function membership as the changes happened in the specific factor.

As much as fuzziness helps the rule evaluation during the intermediate steps, the final desired output for each variable is generally a single number for showing suitability of that output variable representing for a methodology. However, the aggregate of a fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single output value from the set. Rules 1 to 4 suggest the selection of RUP on priority while Rules 5 to 8 suggest the selection of RAD on priority and Rules 9 to 12 suggest XP on priority for software development. With respect to values of input parameters for the project under development, surface graphs in Fig 7(a), 7(b) and 7(c) show reflection on output.

Table -1 Membership Functions and their Parameter Values for all these Variables

		MEMBERSHIP FUNCTION PARAMETERS		
		LOW	MEDIUM	HIGH
FACTORS	COR	TRIANGULAR FUNCTION [0 2 4]	TRAPEZODIAL FUNCTION [3.5 4.8 5.8 7]	TRIANGULAR FUNCTION [6.5 8.3 10]
	AIE	TRIANGULAR FUNCTION [0 2 4]	TRIANGULAR FUNCTION [3.5 5.25 7]	TRIANGULAR FUNCTION [6.5 8.3 10]
	CIR	TRIANGULAR FUNCTION [0 2 4]	TRIANGULAR FUNCTION [3.5 5.25 7]	TRIANGULAR FUNCTION [6.5 8.3 10]
	SC	TRIANGULAR FUNCTION [0 2 4]	TRAPEZODIAL FUNCTION [3.5 4.8 5.8 7]	TRIANGULAR FUNCTION [6.5 8.3 10]
	SyC	TRAPEZODIAL FUNCTION [0 1.5 2.5 4]	TRAPEZODIAL FUNCTION [3.5 4.8 5.8 7]	TRIANGULAR FUNCTION [6.5 8.3 10]
	HRR	TRIANGULAR FUNCTION [0 2 4]	TRAPEZODIAL FUNCTION [3.5 4.8 5.8 7]	TRIANGULAR FUNCTION [6.5 8.3 10]
	RC	TRIANGULAR FUNCTION [0 2 4]	TRAPEZODIAL FUNCTION [3.5 4.8 5.8 7]	TRIANGULAR FUNCTION [6.5 8.3 10]
	TPS	TRIANGULAR FUNCTION [0 2 4]	TRAPEZODIAL FUNCTION [3.5 4.8 5.8 7]	TRIANGULAR FUNCTION [6.5 8.3 10]
OUTPUTS	RAD	TRIANGULAR FUNCTION [0 2 4]	TRIANGULAR FUNCTION [3 5 7]	TRIANGULAR FUNCTION [6 8 10]
	XP	TRIANGULAR FUNCTION [0 2 4]	TRIANGULAR FUNCTION [3 5 7]	TRIANGULAR FUNCTION [6 8 10]
	RUP	TRIANGULAR FUNCTION [0 2 4]	TRIANGULAR FUNCTION [3 5 7]	TRIANGULAR FUNCTION [6 8 10]

Table -2 Fuzzy Rule base for finding the Appropriate Software Development Methodology

		FACTORS AS INPUT VARIABLES								OUTPUT Suitability of Methodology
		COR	AIE	CIR	SC	SyC	HRR	RC	TSP	
RULES SET	RULE 1	H	H	M	H	H	H	H	H	RAD = M XP=L RUP = H
	RULE 2	H	M	L	M	H	M	H	M	RAD = L XP=L RUP = H
	RULE 3	H	H	M	H	H	H	M	M	RAD = L XP=L RUP = H
	RULE 4	H	M	M	H	M	H	M	H	RAD = L XP=M RUP = H
	RULE 5	M	M	H	M	M	H	M	M	RAD = H XP=M RUP = L
	RULE 6	M	M	H	M	M	H	L	L	RAD = H XP=L RUP = L
	RULE 7	M	M	H	H	M	M	L	L	RAD = H XP=M RUP = L
	RULE 8	M	M	H	M	M	M	M	L	RAD = H XP=L RUP = L
	RULE 9	L	L	H	M	L	L	M	H	RAD = M XP=H RUP = L
	RULE 10	L	M	H	M	L	L	M	M	RAD = L XP=H RUP = L
	RULE 11	L	M	H	M	L	H	M	H	RAD = M XP=H RUP = L
	RULE 12	L	L	H	M	M	L	M	M	RAD = L XP=H RUP = L

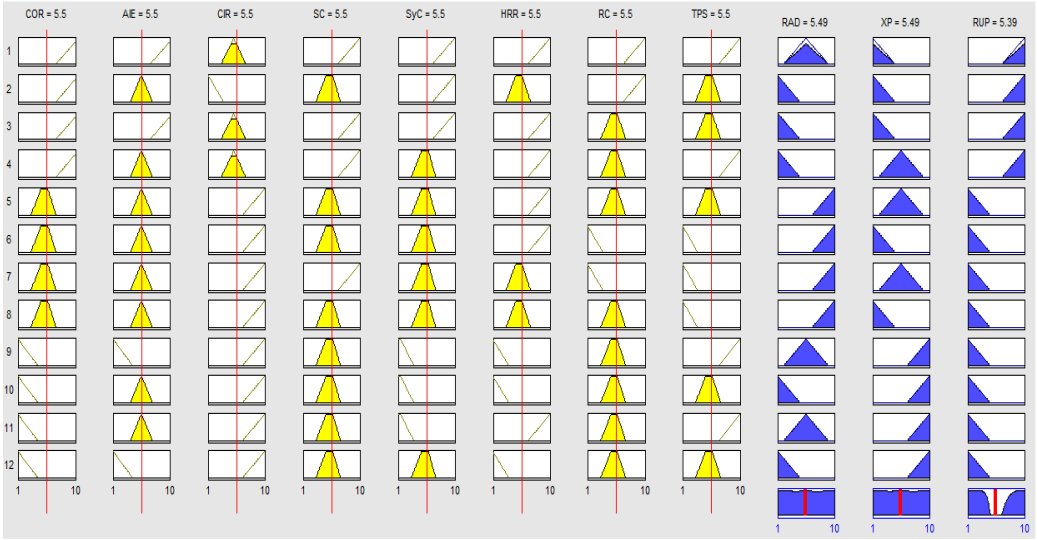
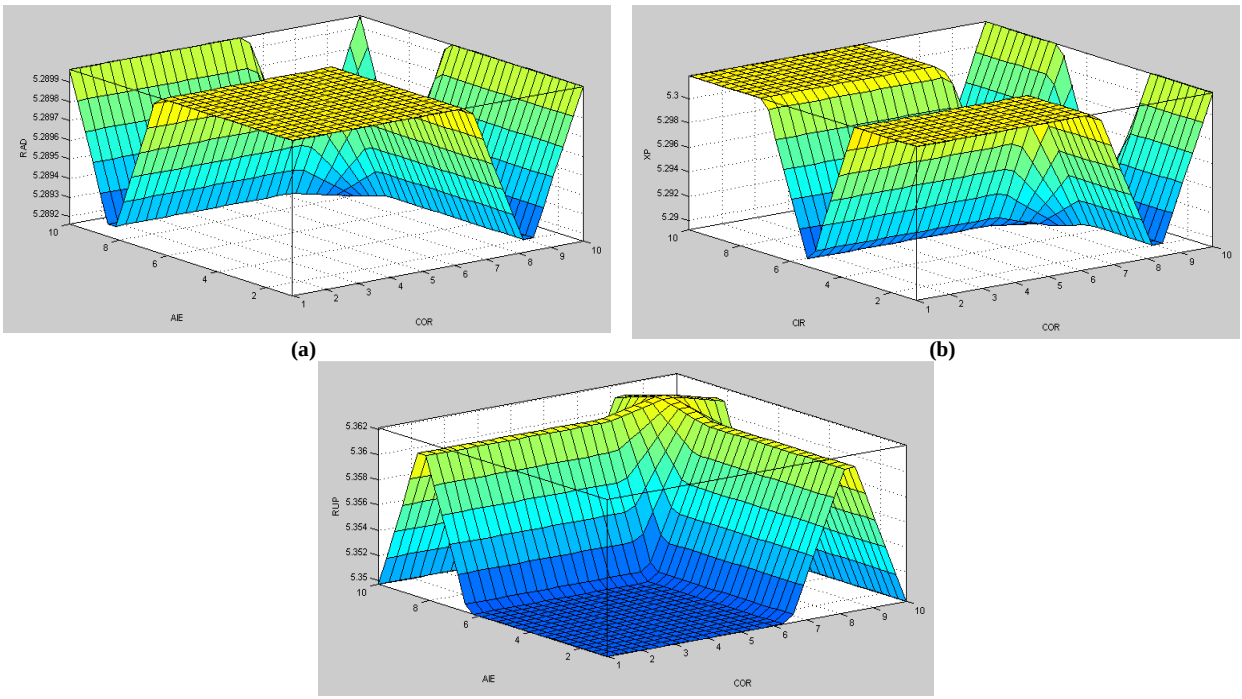


Fig 6: Rule View of Input /output Membership Functions



(a) **(b)**

(c)

Fig 7: Surface View of (a) RAD (b) XP and (c) RUP

CONCLUSION

This paper evaluates the selection suitability of a methodology (RAD, XP, and RUP) for a development of the current project. To perform this task, a Fuzzy Inference System is used in MATLAB which takes into account all the arguments that must affect the selection choice of a methodology. Each parameter is defined by membership functions. Then 12 different rules are made based on 'if-then' conditions. This research can be used for future scope to get the more accurate selection decision in the field of software project development. If number of inputs parameters can be increased or some other parameters can be added which actually affects the selection suitability so then same model with little modification can give the accurate selection decision. The results of our research are useful for the developers by helping them to identify what software development methodology can be used with success for a specific project.

REFERENCES

[1] Malik Hneif, Siew Hock Ow, Review of Agile Methodologies in Software Development, International Journal of Research and Reviews in Applied Sciences, **2009**, 1 (1), 1-8.
[2] D Avison and G Fitzgerald, Information Systems Development: Methodologies, Techniques & Tools, 4th Edition, McGraw-Hill Education, **2006**.
[3] *www.projectmanagement.com/content/processes/11306.cfm*, Accessed on 27th Dec, **2014**.
[4] *http://www.technotrice.com/rad-model-software-engineering*, Accessed on 27th Dec, **2014.**
[5] Adrienne Farrell, Selecting a Software Development Methodology based on Organizational Characteristics, Athabasca, Alberta October, **2007.**
[6] K Beck, Extreme Programming Explained: Embrace Change, Addison-Wesley Longman Pub. Co., USA, **2000**.
[7] *http://www.extremeprogramming.org*, Accessed on 28th Dec, **2014**.
[8] P Kruchten, *Rational Fellow 'What is the Rational Unified Process? Rational Software Canada*, **2001**.
[9] G Tattersall, Supporting Iterative Development through Requirements Management, http://www.ibm.com/developerworks/rational/library/2830.html**,** Accessed on 21st Jan, 2015.
[10] DA White and DA Sofge, *Handbook of Intelligent Control, Neural, Fuzzy, and Adaptive Approaches*, Van Nostrand Reinhold, New York, **1992**.
[11] LX Wang and JM Mendel, Fuzzy Adaptive Filter: with Application to Nonlinear Channel Equalization, *Proceeding of 2nd International Conference on Fuzzy System*, **1993**, 895–900.
[12] M Brown and CJ Harris, *Neurofuzzy Adaptive Modeling and Control*, Hemel Hempstead: Prentice Hall, **1994**.
[13] B Kosko, Fuzzy Systems as Universal Approximators, *Proceeding of International Conference on Fuzzy System*, **1998,** 1153–1162.
[14] B Peter, N Stephan and W Roman, A Brief Course in Fuzzy Logic and Fuzzy Control, Department of Knowledge-based Mathematical Systems, *www.esru.strath.ac.uk/Reference/concepts/fuzzy/fuzzy.htm*, **1996**.
[15] Ranjan Srivastava, Praveen Kumar, Sirish Singh, Ajit Pratap and G Raghurama, Software Testing Effort: An Assessment Through Fuzzy Criteria Approach, *Journal of Uncertain Systems*, **2011**, 5 (3), 183-201.