Fuzzy Logic Based Revised Defect Rating for Software Lifecycle Performance Prediction Using GMR

A. K. Verma[1], Anil R[2] and **Dr. Om Prakash Jain[3]**

Abstract - Software service organizations have adopted various software engineering process models and are practicing it earnestly. Even though this has helped the organizations to improve the quality and the profit margins; there exists a need to compare different groups within it so as to concentrate on the weaker sections. In this paper, the authors propose a revised model for defect rating that can be used for calculating group maturity within the organization. Fuzzy logic approach is used for the proposed model considering the linguistic or imprecise nature of the software measurements. The output of this model can be used as one of the parameter for predicting different software parameters within the software lifecycle.

.

Index Terms - Defect rating, Fuzzy logic, Historical data.

1. INTRODUCTION

Today, there exists many software reliability models [1],[2], [3], [4], [5], [6], [7], [8], [9] which predicts the defect density early in the life cycle. However, these models use the current trend of the defects for prediction. Most of these models are developed using some large software projects where the size of the source code is the range of many hundred thousand lines of codes. It is easy to develop and apply these models on large software projects because of the significant duration and effort spent. However, for industrial software projects that are of size less than a hundred thousand lines of code and being executed in less than six months, it is difficult to use these models for early prediction of the software defect density. In these projects, the average duration of testing may last only a couple of weeks.

Historical information from the past projects also needs to be used for the prediction of the defect density of the new projects [10]. The people and the maturity of the organization are playing an important role in the quality of the software being developed. One can not ignore these facts while predicting the quality of the software under development. In this paper, the authors propose a model that uses the historical information from the past projects and gives a rating for the present project which can be used along with other project parameters to predict the defect density of the project. Fuzzy logic approach is used for the developing the proposed model considering the advantages of fuzzy logic in converting the experts knowledge into fuzzy rules. The paper is organized in the following way. Section II introduces the concept of Fuzzy logic. Section III talks about the problem overview, Section IV talks about the parameters used for the model and the method of calculations, Section V talks about the proposed model using Fuzzy systems, section VI discuses about the application of the proposed model on industrial data and section VII concludes the paper along with future work.

[1]*Indian Institute of Technology, Mumbai*
[2,3]*L&T Infotech, Mumbai*

2. FUZZY LOGIC

Fuzzy logic is invented by Zadeh in 1965. [11] [12]. It is being used in many important investigations since its invention. This concept provides a natural way of dealing with problems where the main source for impreciseness is the absence of crisply defined criteria. In fuzzy approach, the concerned phenomenon in the system is controlled by linguistic uncertainties. A typical fuzzy system consists of a fuzzifier, fuzzy engine and a defuzzifier. Due to the simplicity associated with it, Mamdani method is the most commonly used fuzzy interference engine even though there exists many other approaches [13], [14]. A sequence of fuzzy interface rules determines internal structure of the fuzzy engines. A typical fuzzy system consists of four steps.

1. Using membership functions, an input value is translated into linguistic terms. How much a given numerical input, which is under consideration, fits into the linguistic terms, is decided by the membership function.
2. Fuzzy rules are evolved by considering the different permissible combinations of input and output membership functions. The rules are defined with the use of experts' knowledge in the field under consideration.
3. The derived rules are applied to the membership functions and the aggregation of the outputs of the all rules takes place. This is performed by the fuzzy interference engine which maps the input membership function and the output membership function using the defined fuzzy rules.
4. Converting the resultant fuzzy output into a crisp number which is called as defuzzification.

3. PROBLEM OVERVIEW

Quality of the software being developed in an organization depends not only on the present project conditions, but also on the past performance of the group which develops the software. Considering this, a rating based on the historical data was developed using fuzzy logic technique. Group maturity rating (GMR) [15] is defined for predicting the software performance of a group with in a typical software organization of high maturity. This rating uses five parameters such as, schedule variance, effort variance, customer satisfaction index, process compliance index and defect rating. Since the parameters used for arriving at the model are either linguistic or data is uncertain or vague, fuzzy logic approach is considered as the best approach. Group maturity rating is being used as one of the environmental parameter apart from the project metrics for better prediction using Fuzzy-neuro approach.

The defect rating used in the first version of GMR consists of two parameters, defect density and residual defect density and was defined using fuzzy logic approach. Even though, this model gives a good rating on the maturity of the groups under consideration, the defect rating can be refined by incorporating the review effectiveness as the third parameter, considering the fact that quality of the software under consideration depends of the effectiveness of the review which is being carried out. Also in some cases, the relative error with the existing model is on the higher side that can be reduced.

4. PARAMETERS UNDER CONSIDERATION

a. Defect Density

Defect density is one of the important metrics of software organizations and gives a picture of the quality of the projects of that organization. Defect density is defined as the defects per unit size of the software entity being measured. Low value of defect density is better, however, the same needs to be investigated, since ineffective review and testing also leads to low defect density. Defect density can be correlated with many parameters like the project management practices and processes followed by the project team, the technical knowledge of the organization, and on the competency of the people. Due to these factors, the historical information about the defect density of projects will always help the organization to decide on the time required for review and testing and stoppage rules of testing. Even though the defects found out during the review and testing are resolved before shipping, it takes a significant effort and time from the project. This will directly affect the profit of the organization. The membership functions for defect density are decided using the expert opinion and the historical baseline metrics. Trapezoidal membership functions are considered for defect density. The membership functions of defect density are decided as "Excellent", "Very good", "Good" and "Poor". The elements of the fuzzy sets are determined using the historical baseline mean and the control limits. Table I illustrates the formulae used to find out the membership values of defect density.

Membership function	Membership values
Very good	$0, \mu\text{-}\frac{9\sigma}{2}, \mu\text{-}\frac{7\sigma}{2}, \mu\text{-}\frac{5\sigma}{2}$
Good	$\mu\text{-}\frac{7\sigma}{2}, \mu\text{-}\frac{5\sigma}{2}, \mu-\frac{3\sigma}{2}, \mu-\frac{\sigma}{2}$
Poor	$\mu-\frac{3\sigma}{2}, \mu-\frac{\sigma}{2}, \mu+\frac{\sigma}{2}, \mu+\frac{3\sigma}{2}$
Very poor	$\mu, \mu+\sigma, \mu+2\sigma, \infty$

Table 1: Membership Values for Defect Density

b. Residual Defect density

Residual defect density shows the quality of the projects delivered by an organization and this is also one of the important defect metrics for an organization. Residual defect density (RDD) is the measure of the unresolved defects after release of the software entity per unit size. This number indicates the number of defects passed on to the customers after completing the in-house testing. RDD plays a crucial role in the customer satisfaction since it directly affects the customer whereas; DD defines in the quality of the in-house development. The membership functions for residual defect density are decided using the expert opinion and the historical baseline metrics. Trapezoidal membership functions are also considered for residual defect density. The membership functions of residual defect density are decided as "Excellent", "Very good", "Good" and "Poor". The elements of the fuzzy sets are determined using the historical baseline mean and the control limits. Table II illustrates the formulae used to find out the membership values of residual defect density.

Membership function	Membership values
Very good	$0, 0, \mu\text{-}\frac{3\sigma}{2}, \mu-\sigma$
Good	$\mu\text{-}\frac{3\sigma}{2}, \mu-\sigma, \mu+\frac{3\sigma}{4}, \mu+\frac{5\sigma}{4}$
Poor	$\mu+\frac{3\sigma}{4}, \mu+\sigma, \mu+\frac{13\sigma}{4}, \mu+\frac{15\sigma}{4}$
Very poor	$\mu+3\sigma, \mu+\frac{7\sigma}{2}, \mu+\frac{9\sigma}{2}, \infty$

Table 2: Membership Values for Residual Defect Density

c. Review Effectiveness

During software development, there exist a lot of opportunities for errors. Even though, in ideal conditions, one expects no defects are injected during the development process, the same is an impossible target. In this scenario, the best possible method is to remove the maximum possible error injected as soon as possible. The first possible chance for finding out the errors while developing software is the review process.

Review effectiveness (RE) is the measure of the efficiency of the review process. It is the ratio of total defects found during reviews to the total no of defects found during the entire life cycle. This can be expressed as,

$$\text{RE}=\frac{\text{Number of defects found during review}}{\text{Number of defects found during lifecycle}}\times 100\%$$

The membership functions for review effectiveness also are decided using the expert opinion and the historical baseline metrics. For this parameter also, Trapezoidal membership functions are considered. The membership functions of residual defect density are decided as "Very Poor", "Poor", "Good" and "Very good". The elements of the fuzzy sets are determined using the

Membership function	Membership values
Very good	$0, 0, \mu-\frac{9\sigma}{4}, \mu-\frac{3\sigma}{2}$
Good	$\mu-\frac{9\sigma}{4}, \mu-\frac{7\sigma}{4}, \mu-\frac{3\sigma}{2}, \mu-\frac{3\sigma}{4}$
Poor	$\mu-\frac{3\sigma}{2}, \mu-\frac{3\sigma}{4}, \mu-\frac{\sigma}{4}, \mu+\sigma$
Very poor	$\mu+\frac{\sigma}{4}, \mu+\frac{3\sigma}{4}, 100, 100$

Table 3: Membership Values for Review Effectiveness

historical baseline mean and the control limits. Table III illustrates the formulae used to find out the membership values of review effectiveness.

The output elements are selected as rating "A", "B", "C" and "D", where "A" is the best rating and "D" is the worst rating. These are chosen carefully with the help of experts in the field and converted into defect rules as stated in the next session.

5. DEFECT RATING

There exists a unique relationship between the parameters mentioned in the previous section. Considering this DD, RDD and RE are to be treated together. Low DD and low RDD is the best. When RDD is more and DD is less, it implies to the ineffective in-house testing and review. Here the influence of review effectiveness comes into picture. An effective review will definitely helps the defect densities to come down, but may not be in a linear scale. Considering these, a new parameter called Defect rating (DR) is developed using the different combinations of DD, RDD and RE. This will help the organization to know the health of the project. It also avoids the problem of comparing projects in different technologies since DD and RDD are correlated to the technology and review effectiveness is independent of technology.

A fuzzy logic model was created for defect rating. Sixty four different rules were created based on the input – output combination and fed to the fuzzy engine. Some of the example

- Rule 1: if (DD is **Poor**) and (RDD is **Very Poor**) and (RE is **Very Poor**)Then (Defect rating is **B**)
- Rule 11: if (DD is **Poor**) and (RDD is **Good**) and (RE is **Good**)Then (Defect rating is **C**)
- Rule 33: if (DD is **Very Good**) and (RDD is **Very Poor**) and (RE is **Very Poor**)Then (Defect rating is **D**)
- Rule 22: if (DD is **Good**) and (RDD is **Poor**) and (RE is **Poor**)Then (Defect rating is **C**)
- Rule 36: if (DD is **Very Good**) and (RDD is **Very Poor**) and (RE is **Very Good**)Then (Defect rating is **B**)
- Rule 48: if (DD is **Very Good**) and (RDD is **Very Good**) and (RE is **Very Good**)Then (Defect rating is **A**)
- Rule 52: if (DD is **Excellen**t) and (RDD is **Very Poor**) and (RE is **Very Good**)Then (Defect rating is **B**)
- Rule 61: if (DD is **Excellent**) and (RDD is **Very Good**) and (RE is **Very Poor**)Then (Defect rating is **B**)

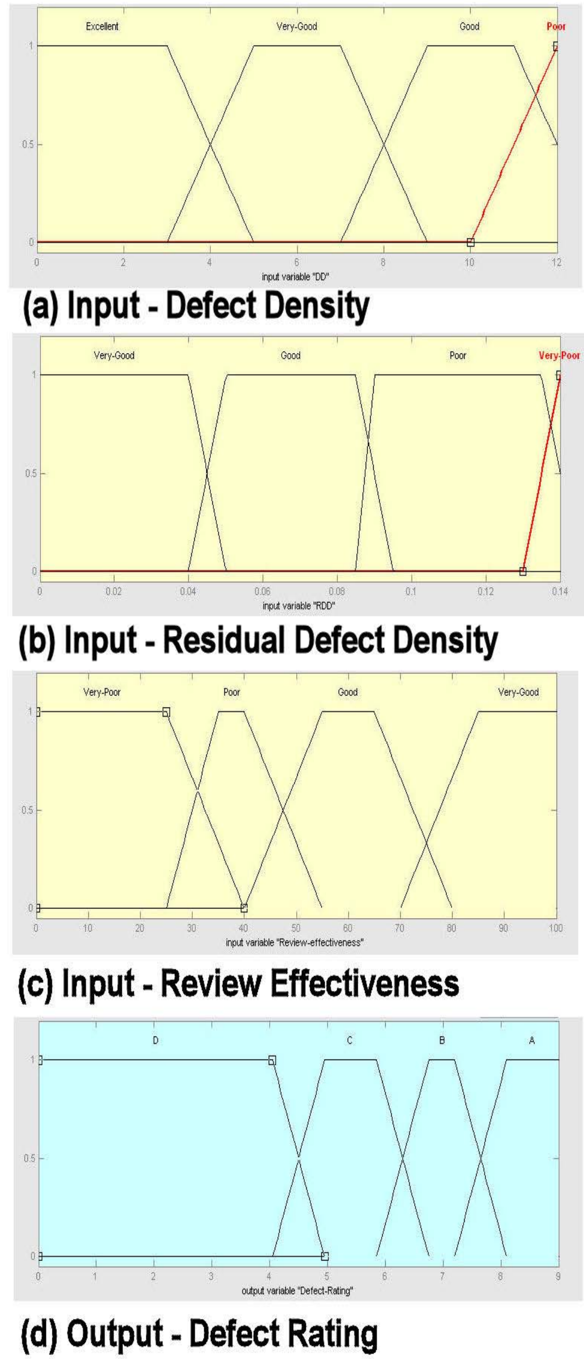

Figure 1: Membership Functions for Inputs and Output of Defect Rating

Mamdani method is used as the fuzzy interference engine. Defuzzified crisp output it taken as the input to the defect rating. Fig. 1 illustrates the mapping of inputs of the fuzzy logic into appropriate membership functions. The rules are created using the fuzzy system editor contained in the Fuzzy Logic Toolbox of Matlab 7.0. Control surface of Defect rating based on fuzzy rules is illustrated in Fig. 2 and Fig. 3. The fuzzy inference diagram in Fig. 4 displays all parts of the fuzzy inference process from inputs to outputs. Each row of plots corresponds to one rule, and each column of plots corresponds to either an input variable or an input variable. One can use the fuzzy inference diagram to change the inputs and to find out the corresponding outputs.

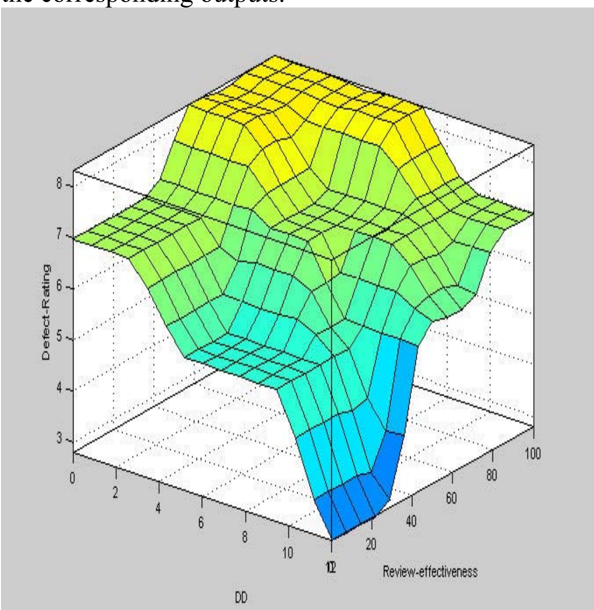

Figure 2: - Control surface for Defect rating fuzzy logic application – DD Vs RE

6. CASE STUDY

In order to validate the revised model of rating, new model is checked with the same set of industrial project data. The case study was employed with the data from six different groups from a typical software organization. The data set consists of data from 140 projects in the recent one year, which is filtered from a larger set of data to get a range of output. Outliers, which are the abnormal project data with large noise, are removed from the selected set of project data to arrive at best results. The project data is pre processed and removed five

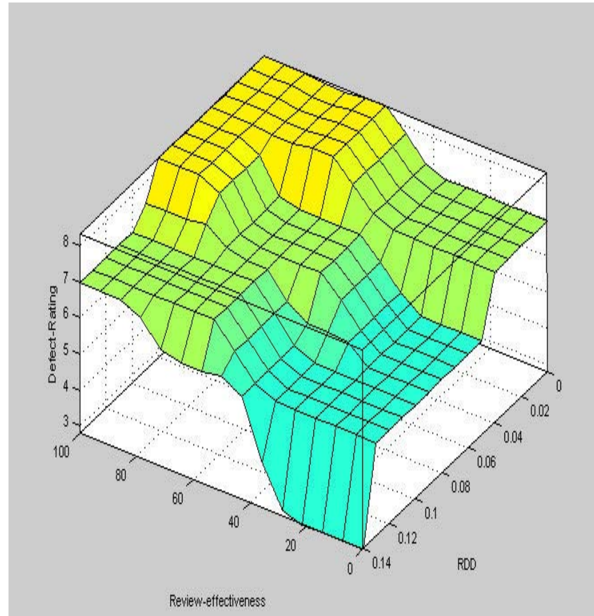

Figure 3: Control surface for Defect rating fuzzy logic application – RDD Vs RE

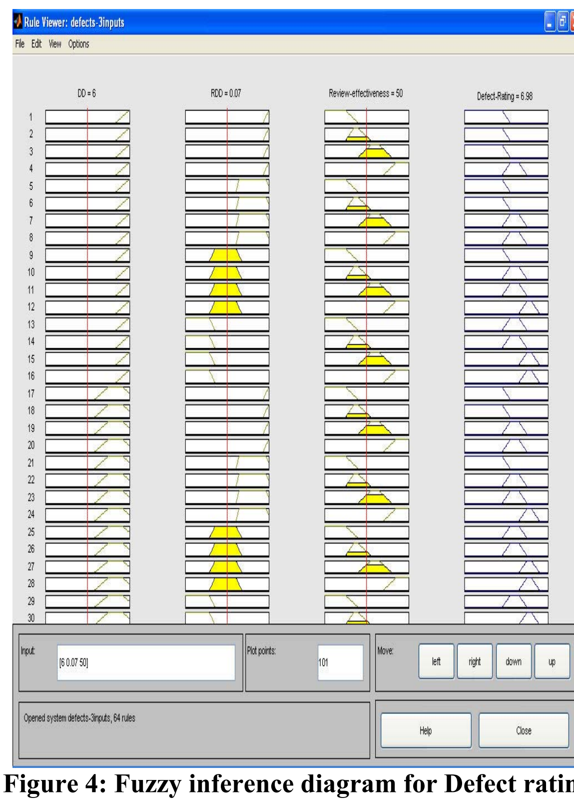

Figure 4: Fuzzy inference diagram for Defect rating

projects from the original database. The database is divided into three sets, based on the period of execution of these projects. Defect rating was calculated separately using the industrial data and the crisp output arrived from the defect rating model is fed as input to the Group maturity model. The output of the revised GMR was compared with the output of the earlier GMR and the same is produced in the table.

(a) Quarter 1

Group	Trad. rating	GMR		GMR Revised	
		Rating	MRE in %	Rating	MRE in %
Group 1	66.67	52.79	20.82	70.00	5.00
Group 2	66.67	70.00	5.00	73.98	10.97
Group 3	46.67	32.83	29.65	32.83	29.65
Group 4	40.00	32.34	19.14	32.34	19.14
Group 5	66.67	70.00	5.00	70.00	5.00
Group 6	53.33	46.33	13.13	46.33	13.14

(b) Quarter 2

Group	Trad. rating	GMR		GMR Revised	
		Rating	MRE in %	Rating	MRE in %
Group 1	60.00	43.02	28.30	70.00	16.67
Group 2	60.00	70.00	16.67	70.00	16.67
Group 3	33.33	31.63	5.10	31.64	5.09
Group 4	46.67	41.90	10.22	41.90	10.22
Group 5	66.67	53.05	20.43	53.01	20.49
Group 6	46.67	44.95	3.67	44.95	3.69

(c) Quarter 3

Group	Trad. rating	GMR		GMR Revised	
		Rating	MRE in %	Rating	MRE in %
Group 1	80.00	71.95	10.06	85.00	6.25
Group 2	60.00	70.00	16.67	70.00	16.67
Group 3	40.00	50.00	25.00	50.00	25.00
Group 4	60.00	55.96	6.74	55.97	6.71
Group 5	53.33	44.90	15.82	44.90	15.82
Group 6	26.67	31.82	19.31	31.81	19.28

Table 4: Comparison of Models

a. Evaluation Criteria

The criterion, Magnitude of relative error (MRE) is employed to asses and compare the performance of the model with respect to the existing model. It can be defined as

$$MRE = \frac{|\text{Excisting Rating-Group Maturity Rating}|}{\text{Excisting Rating}}$$

MRE value is calculated for each group i whose rating is to be determined.

To find out the mean error of the model, mean magnitude of the relative error is also determined, which can be calculated as

$$MMRE = \frac{1}{N}\sum_{i=1}^{N} MRE_i$$

The result of the evaluation is shown in the table IV. The mean magnitude of the relative error (MMRE) for the entire data set consisting of the data from all three quarters is reduced to 13.64% from 15.04% which was reported by the earlier model. Considering the vagueness and uncertain data and linguistic parameters, this error is well within the acceptable limit and the revised defect rating is performing better than the previous model.

7. CONCLUSION AND FUTURE WORK

In this paper, a revised model is proposed for rating the different groups within an organization based on the defect density, residual defect density and the review effectiveness of the historical projects. A fuzzy logic approach is used for defining the model. The model is used for calculating the group maturity rating for a typical software organization of high maturity. The model is then compared with the existing model and the results were discussed. It is observed that while applying the revised model, the man magnitude of the relative error is reduced to 13.64% from 15.04% reported by the previous model.

This paper offers some instances based on the current research into the aspect of using the historical data for predicting the various parameters of the software project throughout the development life cycle. Defect rating will be used as one of the environmental parameter apart from the project metrics for better prediction of software projects using fuzzy-neuro approach.

REFERENCES

[1] A. L. Goel and K. Okumoto, Time-dependent fault-detection rate model for software and other performance measures, IEEE Transactions on Reliability, 28, 1979, p 206-211

[2] H. Pham, A software cost model with imperfect debugging, random life cycle and penalty cost, International Journal of Systems Science, vol. 27, Number 5, 1996, p 455-463

[3] M. Ohba and S. Yamada, S-shaped software reliability growth models, Proc. 4th Int. Conf. Reliability and Maintainability, 1984, pp 430-436
[4] X. Teng and H. Pham,"A software cost model for quantifying the gain with considerations of random field environments", IEEE Transactions on Computers, vol 53, no. 3, 2004
[5] H. Pham, L. Nordmann, and X. Zhang, A General imperfect software debugging model with s-shaped fault detection rate, IEEE Transactions on Reliability, vol. 48, no. 2, 1999, p 169-175
[6] H. Pham and X. Zhang, An NHPP software reliability models and its comparison, International Journal of Reliability, Quality and Safety Engineering, vol.4, no. 3, 1997, p 269-282
[7] S. Yamada, M. Ohba, and S. Osaki, S-shaped reliability growth modeling for software fault detection, IEEE Transactions on Reliability, 12,1983, p 475-484
[8] S. Yamada and S. Osaki,"Software reliability growth modeling: models and applications", IEEE Transactions on Software Engineering, 11, 1985, p 1431-1437
[9] S. Yamada, K. Tokuno, and S. Osaki, Imperfect debugging models with fault introduction rate for software reliability assessment, International Journal of Systems Science, vol. 23, no. 12, 1992
[10] Business Unit Rating Ver. 5.0, Internal document, Larsen and Toubro Infotech limited, 2007.
[11] Lotfi A Zadeh, Fuzzy sets, Information and control, vol 8, pp 338-353, 1965.
[12] Lotfi A Zadeh, The concept of a linguistic variable and its application to approximate reasoning–I, Information Sciences, Volume 8, Issue 3, pp 199-249, 1975
[13] George K Klier and Tina A Folger, Fuzzy sets, Uncertainity and Information, Prentice Hall, 1988.
[14] Puyin Liu and Hongxing Li, Fuzzy Neural Network theory and Application, World Scientific, 2004.
[15] Ajit Kumar Verma, Anil R and Om Prakash Jain, "Fuzzy logic based Group Maturity Rating for Software Performance Prediction" International Journal of Automation and Computing, volume 4, issue 4, 406-412, October 2007