International Journal of Automation and Computing 04(4), October 2007, 406-412
DOI: 10.1007/s11633-007-0406-8

Fuzzy Logic Based Group Maturity Rating for Software Performance Prediction

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

[1] Indian Institute of Technology Bombay, Powai, Mumbai 400 076, India

[2] Larsen & Toubro Infotech Limited, Powai, Mumbai 400 072, India

Abstract: Driven by market requirements, software services organizations have adopted various software engineering process models (such as capability maturity model (CMM), capability maturity model integration (CMMI), ISO 9001:2000, *etc.*) and practice of the project management concepts defined in the project management body of knowledge. While this has definitely helped organizations to bring some methods into the software development madness, there always exists a demand for comparing various groups within the organization in terms of the practice of these defined process models. Even though there exist many metrics for comparison, considering the variety of projects in terms of technology, life cycle, *etc.*, finding a single metric that caters to this is a difficult task. This paper proposes a model for arriving at a rating on group maturity within the organization. Considering the linguistic or imprecise and uncertain nature of software measurements, fuzzy logic approach is used for the proposed model. Without the barriers like technology or life cycle difference, the proposed model helps the organization to compare different groups within it with reasonable precision.

Keywords: Group maturity rating, fuzzy logic, fuzzy sets, historical data, software projects.

1 Introduction

Nowadays, most of the systems that we are using contain a software. These systems have become an integral part of modern society and are being used in our day to day life, from home appliances to nuclear reactors, from automobiles to satellite navigation. As organizations are spending significant resources in software development, software project management is very vital for the success of this organization. With the rapid increase in size and complexity of software, the need for better prediction is becoming more and more important. Today, there are many techniques and tools available for effective project management. Techniques like constructive cost model (COCOMO)[1,2] have been created to estimate the project cost parameters before the development of the software. Capability maturity models (CMM) or capability maturity model integration (CMMI)[3,4] discusses in detail about software project management and the need for the usage of historical data for predicting the project parameters. The project management body of knowledge[5] gives better understanding on managing a project more efficiently. A detailed software metrics helps in day to day management of software projects[6]. In a recent study, an attempt was made to predict the software output metrics with the use of neural networks using historical data and the current project data[7]. However, no approach has proven to be successful in predicting the software output metrics consistently and effectively.

While predicting the software output metrics, one has to consider the environmental parameters like the maturity of the group that develops the software. Without considering this aspect, the prediction may become less accurate. There exists a model that takes the historical data and rates different groups accordingly[8].

A fuzzy logic approach is proposed to overcome the limitations of the current model. Fuzzy logic uses the expert knowledge into the fuzzy rules directly. The exciting parameters used in the current model[8] are used with modifications and the rules for setting the limits are defined. The resultant model can be interpreted easily and is generalised so that decision process is very crisp.

The paper is organized as follows. Section 2 introduces the concept of fuzzy logic. Section 3 talks about the problem overview. Section 4 discuses the parameters used for the model and the method of calculations. Section 5 presents the proposed model using fuzzy systems. Section 6 discuses the application of the proposed model on industrial data and Section 7 concludes the paper along with future work.

2 Fuzzy logic

Fuzzy logic has being used in many important investigations since its invention by Zadeh[9,10] in 1965. The fuzzy logic concept provides a natural way of dealing with problems where absence of crisply defined criteria is the main source for impreciseness. In fuzzy approach, linguistic uncertainties control the concerned phenomena in the system. A typical fuzzy system consists of a fuzzifier, fuzzy engine and a defuzzifier. The Mamdani method is the most commonly used fuzzy interference engine due to the simplicity associated with it, even though there exists many other approaches[11,12]. The internal structure of the fuzzy engines are determined by a sequence of fuzzy interface rules. A typical fuzzy system consists of four steps. In the first step, an input value is translated into linguistic terms with the usage of the membership functions. The membership function decides how much a given numerical input fits into the linguistic terms, which are under consideration. In the second step, 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. In the

Manuscript received December 26, 2006; revised April 8, 2007
*Corresponding author. E-mail address: anil.r@iitb.ac.in

third step, the derived rules are applied to the membership functions and the aggregation of the outputs of all the rules takes place. This step is performed by the fuzzy interference engine which maps the input membership function and the output membership function using the defined fuzzy rules. The final step is converting the resulting fuzzy output into a crisp number which is called defuzzification.

3 Problem overview

Organizations are spending a lot of time and money in developing metrics and collecting metrics data of the projects they undertake. Metrics gives a very good overview about the status of the projects and the projects can be tracked very effectively. Most of these organizations have a very good database about past projects. Some of the data are being used for effort estimation and prediction. Most of the organizations, especially, CMM/CMMI level 4 or 5 organizations, publish the baseline on a periodic basis and take actions to improve the baseline figures. However, most of the time, the usage of the historical data is limited only to this. Some organizations use the metrics data to arrive at the performance of the various project groups. This helps the organizations to track the groups quantitatively. It has been observed that the data used for arriving at the performance are often linguistic in nature[13,14]. Even the quantitative data are being converted into linguistic data and are being used for arriving at the performance values.

A rating exists which uses the historical data to rate different groups within the organization[8]. The traditional rating is based on five parameters. The steps for arriving at the traditional rating are as follows.

Step 1. Obtain the baseline mean and standard deviation value from the historical data.

Step 2. Calculate the range from baseline mean and standard deviation. Split range into defined intervals based on expert opinion and give rating as 0, 3, 6, 9 based on the importance of the values.

Step 3. Calculate the rating for each project for each of the parameters mentioned and calculate the average for each of the parameter for the entire group.

Step 4. Calculate the business unit (BU) rating using the equation,

$$\text{BU rating} = \frac{\sum \text{Average rating obtained}}{\text{Rating}_{\max} \cdot \text{Number of parameters}} \%.$$

In this case, the maximum attainable rating $9 \cdot 5 = 45$. The output rating is translated into final rating of A, B or C.

However, the main limitations discovered in this model are

1) Relationship between software input metrics and output metrics are of nonlinear complex nature.
2) Simultaneous application of the numerical data both from software projects and the knowledge of experts is difficult in this model.
3) Software measurements are imprecise and uncertain due to cognitive nature.

4 Parameters under consideration

Considering the above limitations, there is a need for arriving at a revised model which will help the organizations to evaluate various project groups under them. Five major parameters from the existing and closed projects are considered for the proposed model. These parameters are the same as that of the traditional model. They are introduced as follows.

4.1 Schedule variance (SV)

SV is percentage of variance of the actual duration for an activity to the planned duration. It is a measure of variation in meeting the software project′s planned deadline date. It can be calculated using the formula

$$\text{SV} = \frac{(\delta_{\text{actual}} - \delta_{\text{planned}})}{\delta_{\text{planned}}} \cdot 100\%$$

where δ is the duration.

Improved project management and control lead to reduction in SV.

4.2 Effort variance (EV)

EV is the percentage variance of the actual effort with respect to the planned effort. It is a measure of how effectively the estimation and planning was conducted for a software project. It can be calculated using the formula

$$\text{EV} = \frac{(\varepsilon_{\text{actual}} - \varepsilon_{\text{planned}})}{\varepsilon_{\text{planned}}} \cdot 100\%$$

where ε is the effort.

Reduction in EV value is achieved by continuously improving project estimation and planning practices.

4.3 Customer satisfaction index (CSI)

There is a strong relationship between customer satisfaction and customer retention. Customer′s perception of quality and service of software product delivered will determine the success of the organization in the market. A clear understanding of customers′ perceptions helps the software organizations to determine the actions required to meet the customers′ needs. Customer satisfaction measurement helps to focus more on customer outcomes and stimulate actions for improvements in the work practices and processes used within the organization. They can identify their own strengths and weaknesses, where they stand in comparison to their competitors, chart out path for future progress and improvement.

CSI represents the overall satisfaction level of that customer as a number in a scale of 1 to 5, where 1 is the minimum and 5 is the maximum. Sixteen questions in the area of project execution, quality of the service and the communication with the customer are given to the customer to rate. The linguistic terms for rating "not applicable", "dissatisfied", "satisfied partially", "satisfied completely", and "delighted" are given a rating of 1 to 5, respectively. Each question is assigned with a weight. CSI is calculated using the formula

$$\mathrm{CSI} = \left(\frac{\sum\limits_{i=1}^{n} S_i}{\sum\limits_{i=1}^{n} S_{\max}} \right) \cdot 5$$

here

$$S_i = \begin{cases} 0 & , \ r_i = 1 \\ w_i r_i & , \ r_i = 2, 3, 4, 5 \end{cases}$$

and

$$S_{\max} = \begin{cases} 0 & , \ r_i = 1 \\ w_i \cdot 5 & , \ r_i = 2, 3, 4, 5 \end{cases}$$

where w_i is the weight associated with question i, r_i is the rating of question i, and n is the number of questions.

4.4 Process compliance index (PCI)

Most of the organizations have developed a well defined process for the project execution. However, even if an organization defines the best available processes, it will not yield any result unless the same is being followed in the organization with its right spirit. PCI is the quantitative representation of the process being executed as the standards defined within the organization. This is calculated by providing ratings for the compliance of each activities.

PCI can be calculated using the formula

$$\mathrm{PCI} = \left(\frac{\sum\limits_{j=1}^{m} \Phi_j}{\sum\limits_{j=1}^{m} \Phi_{\max}} \right) \cdot 100\%$$

here

$$\Phi_j = \begin{cases} \Psi_j \rho_j & , \ \rho_j = 0, 0.5, 1 \\ 0 & , \ \text{otherwise} \end{cases}$$

and

$$\Phi_{\max} = \begin{cases} \Psi_j & , \ \rho_j = 0, 0.5, 1 \\ 0 & , \ \text{otherwise} \end{cases}$$

where Ψ_j is the weight associated with activity j, ρ_j is the rating of activity j, and m is the total number of activities.

4.5 Defect rating (DR)

Defect density (DD) and residual defect density (RDD) are the two important defect metrics that shows the quality of the projects of an organization. DD is the measure of defects per unit size of the software entity being measured. A low value of DD is better; however, the same needs to be investigated, since ineffective review and testing also leads to low DD. DD can be correlated with the technical knowledge of the organization, the project management practices and processes followed by the project team and on the competency of the people. Hence, historical information about the DD of projects will always help the organization to decide the duration of review and testing and stoppage rules of testing.

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. RDD plays a crucial role in the customer satisfaction since it directly affects the customer. Whereas DD plays an important role in the quality of the in-house development. 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.

Considering these factors, DD and RDD are to be treated in pair. A low DD and low RDD is the best. When RDD is more and DD is less, it implies an ineffective in-house testing and review. A new parameter called DR is developed using the different combinations DD and RDD. 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.

A fuzzy logic model was created for DR. The inputs taken are DD and RDD. The membership functions for both DD and RDD are decided using the expert opinion and the historical baseline metrics. Trapezoidal membership functions are considered for both DD and RDD. The membership functions of DD are "excellent", "very good", "good", and "poor". The elements of the fuzzy sets are determined using the historical baseline mean and the control limits. Table 1 illustrates the formulae used to find out the membership values of DD and RDD.

Table 1 Membership values for density

(a) Defect density

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

(b) Residual defect density

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

The output elements are selected as rating "A", "B", "C", and "D", where "A" is the best rating and "D" is the worst rating. Sixteen different rules were cerated based on the input-output combination and fed to the fuzzy engine. Some of the example rules are shown below.

- **Rule 5** : if (DD is very good) and (RDD is very good) then (DR is A).

- **Rule 10** : if (DD is poor) and (RDD is good) then (DR is D).

- **Rule 14** : if (DD is good) and (RDD is good) then (DR is C).

The Mamdani method is used as the fuzzy interference engine. Defuzzified crisp output is taken as the input to the group maturity rating (GMR). The rules are created using the fuzzy system editor contained in the fuzzy logic toolbox of Matlab 7.0. The control surface of DR based on fuzzy rules is illustrated in Fig. 1.

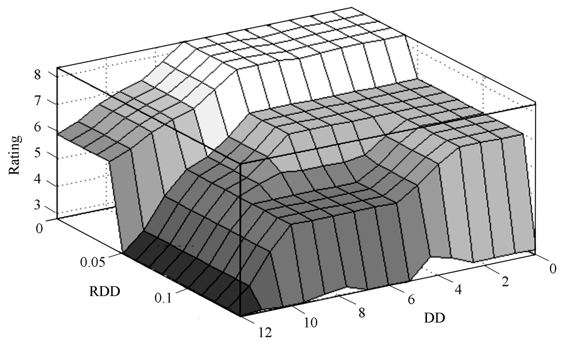

Fig. 1 Control surface for DR fuzzy logic application

The fuzzy inference diagram shown in Fig. 2 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 output variable. One can use the fuzzy inference diagram to change the inputs and to find out the corresponding outputs.

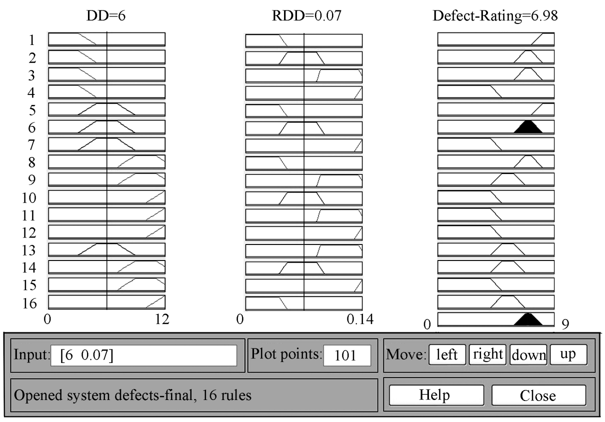

Fig. 2 Fuzzy inference diagram for DR

5 Group maturity rating (GMR)

As mentioned earlier, historical information about the projects under a group is available with the organization. Input parameters mentioned in Section 4 are used for creating the GMR model. Fuzzy approach is selected since the parameters are either linguistic in nature or they are fuzzy in nature. In the proposed model, the fuzzy input sets are SV, EV, PCI, CSI, and DR, whereas, the output parameter of the fuzzy system is GMR which is defined as the rating given to each project group in the organization based on its past performance. The mapping of input of the fuzzy system into appropriate membership functions for SV, EV, PCI, CSI and DR is illustrated in Fig. 3. The membership functions of SV are "very low", "low", "appropriate", "high", "very high", and "extremely high". The membership functions of other input parameters are as follows, EV {"low","appropriate", "high", "very high", "extremely high"}, PCI {"very poor", "poor", "good", "very good"}, CSI {"dissatisfied completely", "dissatisfied partially", "satisfied", "delighted"}, and DR {"poor", "good", "very good", "excellent"}.

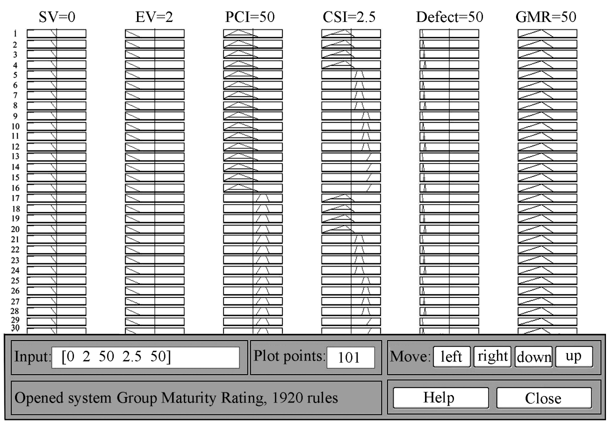

Fig. 3 Fuzzy inference diagram for GMR

Weighted mean and specification limits from the baseline of the historical data is considered for arriving at the membership values of the SV and EV. Using the mean and specification limits, the different membership values are defined with the help of experts in the given field. While defining the membership values, the practical issues are also considered. For example, even if the mean EV is far from zero, we have to give a good rating for those projects whose variance is near to zero. Table 2 illustrates the formulae used to find out the membership values of SV and EV.

Table 2 Membership values for variance

(a) Schedule variance

Membership function	Membership values
Very low	$-\infty, \mu-3\sigma, \mu-\frac{3\sigma}{2}$
Low	$\mu-\frac{13\sigma}{4}, \mu-\frac{11\sigma}{4}, \mu-2\sigma, \mu-\frac{3\sigma}{2}$
Appropriate	$\mu-2\sigma, \mu-\frac{3\sigma}{2}, \mu+\frac{\sigma}{3}, \mu+\sigma$
High	$\mu+\frac{\sigma}{3}, \mu+\sigma, \mu+\frac{3\sigma}{2}, \mu+\frac{9\sigma}{4}$
Very high	$\mu+\frac{3\sigma}{2}, \mu+\frac{9\sigma}{4}, \mu+\frac{5\sigma}{2}, \mu+\frac{7\sigma}{2}$
Extremely high	$\mu+\frac{5\sigma}{2}, \mu+\frac{7\sigma}{2}, \infty$

(b) Effort variance

Membership function	Membership values
Low	$-\infty, \mu-5\sigma, \mu-4\sigma, \mu-2\sigma$
Appropriate	$\mu-4\sigma, \mu-2\sigma, \mu-\frac{5\sigma}{4}, \mu-\frac{3\sigma}{4}$
High	$\mu-\frac{5\sigma}{4}, \mu-\frac{3\sigma}{4}, \mu+\frac{3\sigma}{4}, \mu+\frac{5\sigma}{4}$
Very high	$\mu+\frac{3\sigma}{4}, \mu+\frac{5\sigma}{4}, \mu+\frac{5\sigma}{2}, \mu+\frac{7\sigma}{2}$
Extremely high	$\mu+3\sigma, \mu+4\sigma, \infty$

The input variable DR is output of the fuzzy system mentioned in Section 4. For CSI and PCI, the membership values are also defined.

The output for the fuzzy system is linguistic variable GMR and is defined as {"A", "B", "C"}. Based on the input-output combinations, 1920 rules are created using the fuzzy system editor contained in the fuzzy logic toolbox of Matlab. These rules are fed to the fuzzy engine. Some of the major rules are given below.

- **Rule 1904** : if (SV is appropriate) and (EV is appropriate) and (PCI is good) and (CSI is delighted) and (DR is excellent), then (GMR is A).

- **Rule 1206** : if (SV is low) and (EV is high) and (PCI is very good) and (CSI is dissatisfied partially) and (DR is good), then (GMR is C).

- **Rule 875** : if (SV is very high) and (EV is high) and (PCI is good) and (CSI is satisfied) and (DR is very good), then (GMR is B).

The fuzzy interface diagram in Fig. 3 illustrates the input and output of the GMR model. By changing the input, we can find out the corresponding output. Using the organization′s historical data the maturity rating of the different groups can be obtained. This will be a single measurement unit for the organization to assess different groups since most of the groups will be working on different domains, different technology and on different type of projects, and it will be difficult to compare them without a single measurement unit.

6 Case study

To validate the model of GMR, a 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 year, which are filtered from a larger set of data to get a range output. To arrive at best results we need to remove the outliers, which are the abnormal project data with large noise. The project data is pre-processed and 5 projects are removed from the original database. The database is divided into three based on the period of execution of these projects. The parameter DR was calculated separately using the fuzzy logic model defined in Section 4. The crisp output arrived from the DR model is fed as input to the group maturity model.

6.1 Evaluation criteria

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

$$\mathrm{MRE} = \frac{|\text{Excisting rating} - \text{Group maturity rating}|}{\text{Excisting rating}}.$$

The value of MRE 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 (MMRE) is also determined, which can be calculated as

$$\mathrm{MMRE} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{MRE}_i.$$

The results of the evaluation is shown in Table 3. The MMRE for the entire data set consisting of the data from all three quarters is 15.04%. Considering the vagueness and uncertain data and linguistic parameters, this error is well within the acceptable limit.

Table 3 Comparison of models

(a) Quarter 1

Group	Traditional rating	GMR	MRE (%)
Group 1	66.67	52.79	20.82
Group 2	66.67	70.00	5.00
Group 3	46.67	32.83	29.65
Group 4	40.00	32.34	19.14
Group 5	66.67	70.00	5.00
Group 6	53.33	46.33	13.13

(b) Quarter 2

Group	Traditional rating	GMR	MRE (%)
Group 1	60.00	43.02	28.30
Group 2	60.00	70.00	16.67
Group 3	33.33	31.63	5.10
Group 4	46.67	41.90	10.22
Group 5	66.67	53.05	20.43
Group 6	46.67	44.95	3.67

(c) Quarter 3

Group	Traditional rating	GMR	MRE (%)
Group 1	80.00	71.95	10.06
Group 2	60.00	70.00	16.67
Group 3	40.00	50.00	25.00
Group 4	60.00	55.96	6.74
Group 5	53.33	44.90	15.82
Group 6	26.67	31.82	19.31

7 Conclusions

In this paper, a model is developed to compare the different groups within the organization using fuzzy logic approach. 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 over the traditional approach.

From the results, it has been observed that the MMRE of the proposed model is 15.04% compared with the traditional model. This is within the acceptable limits while considering the vagueness and imprecise nature of the data and the presence of linguistic parameters. The GMR model can be used to compare the different groups within the organization without the barriers like projects with technological and/or life-cycle differences. The GMR model can also be used to compare the groups with respect to the implementation of process models such as CMM/CMMI and ISO 9001:2000. GMR can also be used as one of the parameters for the prediction of software projects.

This paper offers some instances based on the research into the aspect of using historical data for predicting various parameters of the software project throughout the development life cycle. GMR will be used as one of the environmental parameters apart from the project metrics for better prediction using fuzzy-neuro approach. GMR can be used by the organizations to evaluate the maturity of the different groups within it so as to concentrate on and improve the weaker groups.

Acknowledgement

The authors would like to thank Larsen & Toubro Infotech Limited, especially Mr. Sudharma Rao for providing support for the research work.

References

[1] W. B. Boehm, *Software Engineering Economics*, Prentice Hall, NJ, USA, 1981.

[2] W. B. Boehm. *Software Cost Estimation with COCOMO II*, Prentice Hall, NJ, USA, 2000.

[3] CMMI Product Development Team. *Capability Maturity Model Integration (CMMI)*, Version 1.1, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 2002.

[4] CMMI Product Development Team. *CMMI for Development*, Version 1.2, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 2006.

[5] Project Management Institute. *A Guide to the Project Management Body of Knowledge*, 3rd Edition, Project Management Institute, Pennsylvania, USA, 2004.

[6] N. E. Fenton, S. L. Pfleeger. *Software Metrics: A Rigorous and Practical Approach*, PWS Publishing Co., Boston, USA, 1997.

[7] P. N. Mehta. Integration of Product and Process Attributes for Quantiative Modeling in Software, Ph.D. dissertation, Indian Institute of Technology, 2006.

[8] Quality Team. Business Unit Rating, Version 1.0, Internal document, Larsen & Toubro Infotech Limited, Mumbai, India, 2007.

[9] L. A. Zadeh. Fuzzy sets. *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[10] L. A. Zadeh. The Concept of a Linguistic Variable and its Application to Approximate Reasoning, Part I. *Information Sciences*, vol. 8, no. 3, pp. 199–249, 1975.

[11] J. K. George, T. A. Folger. *Fuzzy Sets, Uncertainity and Information*, Prentice-Hall, NJ, USA, 1988.

[12] P. Liu, H. X. Li. *Fuzzy Neural Network Theory and Application*, World Scientific Publishing Co., NJ, USA, 2004.

[13] A. Idri, A. Abran, T. M. Khoshgoftaar. Estimating Software Project Effort by Analogy Based on Linguistic Values. In *Proceedings of the 8th IEEE Symposium on Software Metrics*, Ottawa, Canada, pp. 21–30, 2002.

[14] A. Idri, A. Abran. A Fuzzy Logic Based Measures For Software Project Similarity: Validation and Possible Improvements. In *Proceedings of the 7th International Symposium on Software Metrics*, IEEE, England, pp. 85–96, 2001.

A. K. Verma received the B.Tech (Hons) and Ph.D. (Engg.) degrees from Department of Electrical Engineering, IIT Kharagpur. He has been with IIT Bombay as a faculty since 1988. He is currently a professor in reliability engineering, Department of Electrical Engineering at IIT Bombay. He has over 130 research papers to his credit and has supervised eighteen Ph.D. theses and seventy Master's theses at IIT Bombay. He has been a guest editor of special issues on *Quality Management of Electronics*, *Communications & IT of IETE Technical Review*, *International Journal of Performability Engineering*, *International Journal of Reliability, Quality and Safety Engineering*, and is on the editorial board of various journals. He has been a conference chairman of various international conferences, *ICQRC 2001*, *ICMD 2002*, *ICQRIT 2003*, *ICQRIT 2006* and a patron of *ICRESH 2005*. He has authored a book on *Fuzzy Reliability Engineering: Concepts and Applications*. He is a senior member of IEEE and life fellow of IETE.

His research interests on reliability engineering include interdisciplinary applications in software engineering, computing, maintenance and power system.

Anil R received the B.Tech degree in mechanical engineering from University of Calicut, Kerala, India, in 1995. He is currently working as quality manager with Larsen & Toubro Infotech Limited and pursuing his Ph.D. degree at IIT Bombay, Mumbai, India. His research interests include software reliability and software prediction using neuro-fuzzy approach.

Om Prakash Jain received the Ph.D. degree in electrical engineering from McGill University, Montreal, Canada. He has also studied at premier Indian institutions like Indian Institute of Technology, Kharagpur, and Birla Institute of Technology, Mesra, Ranchi. He currently heads Process Consulting Practice at Larsen & Toubro Infotech Limited. He is a certified lead auditor for ISO 9001:2000, a certified examiner for quality management and a qualified Six Sigma Black Belt. He has published several technical papers in various international journals like IEEE Transactions.

His research interests include experimental software engineering, software prediction and changing paradigm on software quality for software professionals.