ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 3 March 2014 Page No. 4081-4086

Fuzzy Technique for software Quality Measurement

Vishal Choudhary and Dr.P.K Bharti
(Research Scholars)
Department of Computer Engineering
Ranchi University Jharkhand-(India)
E-Mail:vishalhim@yahoo.com,Pa_dutt@yahoo.com

Abstract— The software quality prediction is a major issue these days. in order to develop software quality prediction model, one must first identify the factors that strongly influence software quality and the number of residual errors.unfortunatly,it is extremely difficult, to accurately identify relevant quality factors. that is although exact and discrete metric data are used, inference rules used may be fuzzy in nature. the benefits of inspections, originally indicated by Fagan**[1],**have been re-confirmed by other practitioners. Software inspection is considered as an essential practice to develop high quality software. if it is possible to identify potentially error -prone modules with relatively high degree of accuracy at a little or no extra cost by analyzing the present inspection data. this paper purpose a fuzzy logic based precise approach to quantify quality of software modules based on inspection rate and error density to predict quality factor such as whether a component is fault prone or not.

Keywords: Fuzzy logic, inspection rate, complexity density, Defuzzification

1. Introduction

There are various important metrics for quality prediction of software, Inspection metrics, inspection rate and error density are the most important metrics for checking the quality of a software. past researchers focused their attention on empirically validating cost effectiveness of inspection methods[2,3]

.Barnard[4] identified nine key metrics used in planning ,monitoring, controlling and improving inspection processes.Ebenau[5] was the first who employed inspection metrics to identify modules that are likely to be error prone. Experts are likely to differ in their opinion as to whether or not the inspection rate exceeded the industrial decision boundary is not well defined. Suppose for example ,that an inspection team reported an inspection rate over 380 LOC/h whereas typical inspection rate range from 150 to 200 LOC/h[6].one can convincingly argue that such inspection rate significantly exceeds the reported average from industrial applications, and experts will most likely agree unanimously with the conclusion. however such assessment is fuzzy because the term significantly cannot precisely quantified. This paper is divided in to segments .in segment 2 purposed model is given segment 3 deals with fuzzifcation segment 4 deals with defuzzification segment 5 deals with experimental study and evaluation of quality grades. segment 6 concludes the paper and describes promising topics worthy for further research.

2. Fuzzy for Software Quality Measurement

It is extremely difficult, to identify relevant quality factors accurately.furthermore,the nature of the degree of influence is imprecise. That is, although exact and discrete metric data are used, inference rules used may be fuzzy in nature.int this segment, a fuzzy logic based quality prediction model is proposed using inspection rate (I) and error density€ as metrics for quality assessment.

Inspection Rate and Error density are categorized in to three levels of complexity.i.e Low, Medium, High. The complexity and Co-efficient matrix for inspection Rate and Error density is given in Table 1. And Table 2. respectively.

Table1.Complexity and Co-efficient matrix of Inspection Rate(I)

I	Complexity	Co-efficient(R_i)
0-100	Low	2
100-250	Medium	10
250 or more	High	2

Table2.Complexity and Co-efficient matrix for Error Density(E)

E	Complexity	Co-efficient(R_e)
0-25	Low	5
25-35	Medium	15
35 or more	High	20

3. Fuzzification:

The complexity attributes low, medium and high of the two metrics Inspection Rate(I) and error Density(E) are taken as triangular number(TFN).

Figure 1 and figure 2 represents memberships functions for I and E.

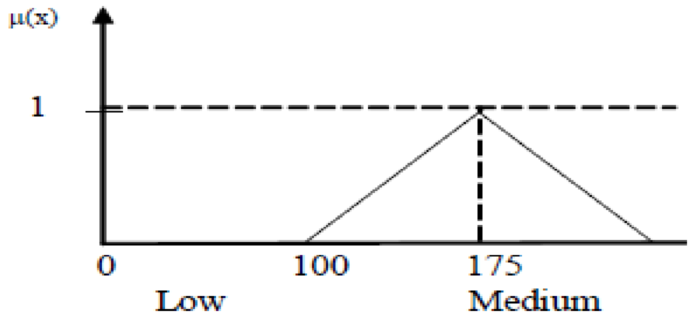


Figure1:Fuzzy Pictorial Representation of I

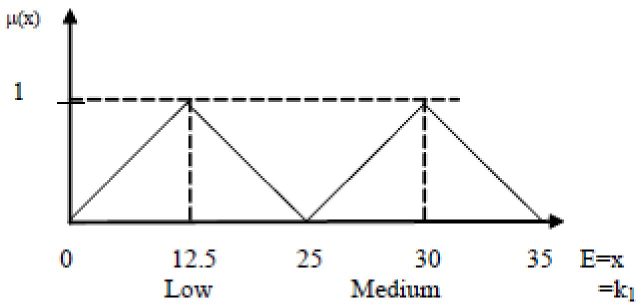


Figure2:Fuzzy pictorial Representation of E

4. Defuzzification:

Defuzzification rule for Ri and Re are defined using table1 and table 2 and TFNs for inspection Rate (I) and Error Density equation 1 and equation 2.

$$Ri = \begin{cases} 2 \quad , & 0 \le I \le 100 \\ \mu i * 10 + (1 - \mu i) * 2 & 1000 \le I \le 250 \\ 2, & 100 \le I \le 250 \end{cases}$$
------------------------------1

$$Re = \begin{cases} \mathrm{\mu e} * 5 \,, & 0 \le \mathrm{E} \le 12. \\ \mathrm{\mu e} * 5 + (1 - \mathrm{\mu e}) * 10, & 12.5 \le \mathrm{E} \le 25 \\ \mathrm{\mu e} * 15 + (1 - \mathrm{\mu e}) * 10, & 25 \le \mathrm{E} \le 30 \\ \mathrm{\mu e} * 15 + (1 - \mathrm{\mu e}) * 20, & 30 \le \mathrm{E} \le 35 \\ 20 & \mathrm{E} \ge 35, \end{cases}$$

-------------------------------------2

Here , μ_i and e are membership functions for I and E respectively. Quality grades are given by equation 3,which shows that higher the quality grade, higher the quality.

A threshold values for quality is selected to distinguish error-prone components from error - free components' based on domain knowledge,

$$Grade = \begin{cases} \left(\frac{Ri}{Re}\right), Re \ne 0 \\ 10 \quad , Re = 0 \end{cases} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots .3$$

5. Experimental Study and Data Analysis:

We have used the dataset composed of 25 detail design and code inspection [SUNs02].these work products were enhancements to the software feature of a local telephone switching system, table 3 provides a listing of the 25 PBX200 inspection records. The overall performance of the inspection shows that there were 23 new inspections and 2 re-inspections.

Table3:Particular of 15 Modules under study.

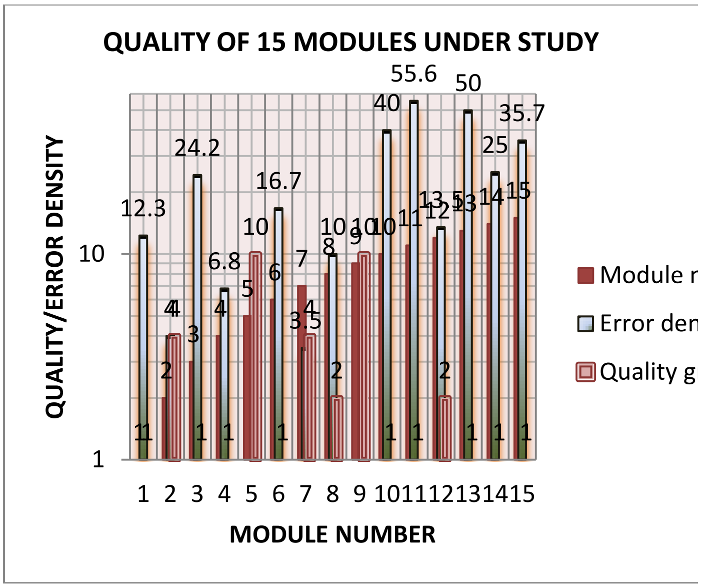
Module no	Components	Meet	Lines	Duration (Hrs)	Efforts	Inspection rate	Defects	Error Density
1	Ops	I_i	81	1	5.0	81	1	12.3
2	Status DMA	I_i	330	2.5	20.5	132	8	24.2
3	LCD-user	I_i	250	1	5.0	250	1	4.0
4	Status-Bus	R	440	1	19.5	440	3	6.8
5	Wait display	I_i	350	1.5	6.5	233	0	0.0
6	Active display	I_i	240	1	5.5	240	4	16.7
7	Status disk	I_i	570	2.5	19.0	228	2	3.5
8	Trans Type4	I_i	400	3	11.3	133	4	10.0
9	Audible Alert	I_i	80	0.5	2.8	160	0	0.0
10	Chan Handler	I_i	200	1.8	20.7	111	8	40.0
11	Dial 0	I_i	36	0.5	4.2	72	2	55.6
13	Directed Recal	I_i	370	2	15.0	185	5	13.5
13	DMA Handler	I_i	40	0.3	2.4	133	2	50.0
14	Alert/Con	I_i	120	1	6.5	120	3	25.0
15	DND	I_i	140	0.8	4.9	175	5	35.7

Table 4 describes the evaluation of the quality grades of various modules. A threshold value is selected to distinguish error -prone components from error -free components based on domain knowledge. such a decision is bound to be subjective.

Table 4: evaluation of quality Grades for different Modules

Module no	Size(KLOC)	Inspection rate	Error density	Ri	Re	Quality grade
1	0.081	81	12.3	2	4.92	1
2	0.330	132	4	5.41	1.6	4
3	0.250	250	24.2	2	9.68	1
4	0.440	440	6.8	2	2.72	1
5	0.350	233	0	3.81	0	10
6	0.240	240	16.7	3.07	6.68	1
7	0.570	228	3.5	4.35	1.5	4
8	0.400	133	10	5.52	4	2
9	0.080	160	0	8.4	0	10
10	0.200	111	40	3.17	20	1
11	0.036	72	55.6	2	20	1
12	0.370	185	13.5	8.93	5.4	2
13	0.040	133	50	5.52	20	1
14	0.120	120	25	4.13	10	1
15	0.140	175	35.7	16	20	1

From the case study we found that the module number 1,3,4,6,10,11,12,13,14,15 is most error prone.



6. Conclusion and future work:

in this paper a fuzzy logic based precise approach is given to quantify the quality of software module based on inspection rate and error density. Software is graded on the quality grade basis in ten grades .modules having quality grades one are supposed to be most error prone while those having quality grade more than ten are considered satisfactory. Evaluation of model is done based on published inspection data. the approach of this paper is very helpful in decision making regarding quality of software. There are several enhancement that are worthy of further research. For example effectiveness of the proposed model needs to be further validated empirically. Such validation would provide useful insights as how fuzzified membership functions can be enhanced.

7. References:

1.M.E Fagan,Design and Code inspection to reduce error in program development,IBM systems J.15,182-211,1976.

2. G.W.Russell,Experience with inspection in ultra large-scale developments,IEEE software 8(19191) 25-31.

3.Doolan,E.P,Experience with Fagan's Inspection method, software Practice Experience 22,173-182,1992.

4. Barnard J.,&Price A. Managing code inspection information,IEEE software 59-69.(1994)

5.Ebenau,R.G,Predictive quality control with software inspection, in software inspection: an industry best practice,IEEE computer society Press,silver Spring,MD,pp.147-156,1996.

6.D.A Wheeler,B.Brykczynski,R.N Messon Jr.,Software inspection:an industry best Practice,IEEE computer society Press,silver spring MD 1996.

7. E.P. Doolan, Experience with Fagan's inspection method, Software Practice experience 22 (1992) 173–182.

8. Emilia Mendes, Nile Mosley, "Web Cost Estimation: An Introduction, Web engineering: principles and techniques", Ch 8, 2005.

9. Harish Mittal, Pradeep Bhatia, "Optimization Criterion for Effort Estimation using Fuzzy Technique". CLEI EJ, Vol. 10 Num. 1 Pap. 2, 2007.

10. Musílek, P., Pedrycz, W., Succi, G., & Reformat, M., "Software Cost Estimation with Fuzzy Models". ACM SIGAPP Applied Computing Review, 8(2), 24-29, 2000.

11. A. J. Albrecht, "Measuring application development productivity", SHARE/GUIDE IBM Application development Symposium.

12.V. R. Basili, K. Freburger, "Programming Measurement and Estimation in the Software Engineering Laboratory", Journal of System and Software, 2, 47-57, 1981.

13. M.E. Fagan, Design and code inspections to reduce errors in program develo1pment, IBM Systems J. 15 (1976) 182–211.

14. B. W. Boehm et al., "Software Cost Estimation with COCOMO II", Prentice Hall, (2000).

15. Lima O.S.J., Farias, P.P.M. Farias and Belchor, A.D., "A Fuzzy Model for Function Point Analysis to Development and Enhancement Project Assessments", CLEI EJ 5 (2), 2002.

Author Profile

Vishal choudhary received his MSc. and M.Tech. degrees in Computer Engineering from Guru Jambeshwar university of science and technology Hisar(India) in 2000 and 2003, respectively. During 2002-2003, he stayed in National Physical Laboratory-CSIR-New Delhi, to study Simulation of moirés signal for mask and wafer alignment in Integrated circuits, from 2003 he is working as Associate Professor In Accurate college of Engineering Greater Noida. he is persuing his Phd in Computer Science and Engineering from Ranchi university where he is working on Fuzzy technique in software Engineering

Prof.P.K.Bharti is presently working in the United college of Engineering as HOD in CSE. he done his Phd in Computer Science and Engineering from Ranchi University,M.Tech from NIT Kurukeshtra and BTech from **MJP** Rohilkhand university.Prof.Bharti published number of papers in international journal including Books on switching theory, database system.professor Bharti has around 13 years expereinece in research and teaching.