Selection of Fuzzy Logic Mechanism for Qualitative Software Reliability Prediction

K. Krishna Mohan[1], A. K. Verma[2] A. Srividya[3]
[1]Reliability Engineering Group, [2]Department of Electrical Engineering, [3] Department of Civil Engineering
Indian Institute of Technology Bombay, Mumbai – 400076, INDIA
kkm@ee.iitb.ac.in, akv@ee.iitb.ac.in, asvidya@ee.iitb.ac.in

***Abstract*-The Proof of Concept (PoC) is usually considered a milestone on the way to a fully functioning prototype. A company presented with a project or proposal will often undertake internal research initially, to prove that the core ideas are workable and feasible, before going further. In general, any predictions about real time implementation are solely based on the prototype studies. A unique take on strengthening the role of the protoype itself, without actually realizing it, would be to arrive at predictions using historical information from similar PoCs or the permeating experience of those involved in projects of comparable nature. Abundance of soft computing techniques should make this crucial bypassing feasible. The purpose of the this work is to demonstrate the same. Validation of this approach could be obtained by comparing the results with the ones obtained on realized prototypes at module level. In a work of the first of its kind involving studies at the PoC level, qualititave predictions for the metric 'number of defects' are obtained using a generic Fuzzy Logic based modeling. A sound mathematical base for the calculation of slopes of various Fuzzy membership functions employed is explained in detail for the case studies considered. This framework is applicable to any of the process oriented developmental systems like RUP.**

Keywords: Proof of Concept (PoC), Rational Unified Process (RUP), Prototype, Software Development Life Cycle (SDLC), Fuzzy Slopes.

I. INTRODUCTION

Reliability is one of the most important aspects of software quality; quality has become a critical concern for the software development organizations and software users. The software system is said to be "reliable "if it performs its specified functions correctly over a long period of time or for a wide variety of operations under a usage environment similar to that of its target customers. Today's business model requires time-to-market products / projects with shorter development life cycles. This results in aggressive schedules for delivery to meet the customer requirements. Reliability prediction has to be done early in the Software Development Life Cycle (SDLC) at the prototype level before the actual development process.

Quality of a software product should be tracked during the software lifecycle right from the architectural phase to its operational phase. Heterogeneous systems consist of several globally distributed components, thus rendering their reliability evaluation more complex with respect to the conventional methods. In this context, performance enhancement for reliability of software process oriented systems assumes prime importance. There is a need to obtain a comprehensive framework to address these core issues.

II. PROPOSED FRAME WORK

The framework proposed consists of the key steps of Concept Development, Software Development and Orientation, which are elaborated upon in the following explanation. A pictorial representation is given in Fig.1. This framework is applicable to any of the process oriented developmental systems like RUP [1, 2].

A. Concept Development

The Concept Development at application level includes:

- Developing concepts of alternatives to meet the goal specifications at application level.
- Enhancement of creativity and innovation.
- A service level agreement (frequently abbreviated as SLA) is a part of a service contract where the level of service is formally defined.
- Service level objectives (SLOs) are a key element of SLA between a service provider and a customer, which are agreed as a means of measuring the performance of the Service Provider and are outlined as a way of avoiding disputes between the two parties due to any possible misunderstandings.
- Developing the marketing and engineering details such as target populace, cost benefit analysis and modes of producing applications.

B. Software Development

- Prediction at component - module level: Finding expected number of defects in the software prior to the results obtained from the PoC (Proof-of–Concept) through the prototype at module level. The predicted results are then compared with the practical output of the prototype in the PoC.
- Based on the prototype, quantitative analysis is then estimated prior to actual implementation of the application development.

C. Orientation

Test measurements are obtained in the form of CPU throughput, response time, % utilization and Disk mean idle time. Load testing for examining the system behavior and performance, could be done by artificially generating actual load, using the intensification tool such as load runner. Load Runner can emulate hundreds or thousands of concurrent users to put the application through the rigors of real-life user loads, while collecting information from key infrastructure components (Web servers, database servers etc). The results can then be analyzed in detail, to explore the reasons for particular behavior.

III. Qualitative Reliability Prediction using Fuzzy logic

This section explains the fuzzy logic system [3] for predicting the defects before taking up a project implementation and well before the beginning the project. Financial services application software case study has been discussed. The output variable refers to the expected number of defects in the software before the beginning of the project and the input variables considered are Requirements, Design, Coding, Unit testing and IST Testing. In all, the input parameters which are dependent on the key influencing factors like techno-complexity (technology + complexity), practitioners' level (experience + product familiarity), creation effort, and review effort are considered.The assignment of linguistic descriptors such as high, medium, low and very low to these factors to translate them into fuzzy variables are shown in Table 1.

TABLE I. Assignment of the linguistic descriptors

Requirements	Low, Medium, High
Design	Low, Medium, High
Coding	Low, Medium, High
Unit Testing	Low, Medium, High
IST Testing	Low, Medium, High
No. of defects	Very Less, Less, Medium, High, Very High(o/p)

A. *Application to the Case Study*

In this proposed model, an financial services application software case study at module level has been discussed. The fuzzy inputs sets are requirements, design, coding, unit testing, and IST support. The output parameter of the fuzzy system is the number of defects, which is defined based on its past experience in the organization for various development projects at module level. Fuzzy sets for the expected number of defects and the input factors are drawn based on the previous experience and knowledge. Requirements, design and coding are measured subjectively on the scale of 0 to 150 as a percentage of the ideal time. Unit testing and IST testing are measured on the scale of 0 to 250 as a percentage of the ideal time. Ideal time is the effort spent in terms of the person hours involved. The output variable is measured on the scale of 0 to 500 defects. As mentioned earlier, historical information about the projects under a group is available with the organization.The input parameters mentioned above are used for creating a qualitative reliability prediction model. Fuzzy approach is selected since the parameters are either linguistic or fuzzy in nature. The membership values for input parameters and output parameters are depicted in the table 2.

B. *Calculation of slopes of Fuzzy membership functions*

A realistic way of incorporating practical factors into the consideration of fuzzy slopes has been put forward in [4] with respect to calculations involved in power system reliability. Taking cue from this, for software reliability prediction purposes, the following has been computed.

A sample mean of effort in person hours can be calculated by a direct average of effort in person hours in various phases of various cycles as follows.

$$\bar{r} = \frac{1}{n}\sum_{i=1}^{1} r_i \tag{1}$$

where $\bar{r}$ is the point estimate of effort, r_i is the i^{th} effort time, and n is the number of data sets. The confidence interval of the expected effort can be estimated using the t-distribution [5]. The estimation method is as follows.

Assume that μ represents the real expected effort and s is the sample standard deviation of effort. According to statistics theory, for a given significant level, it can be affirmed that the random variable $(\bar{r}-\mu)\sqrt{n}/s$ is located between $-t_{\alpha/2}(n-1)$ and $t_{\alpha/2}(n-1)$ with the probability of 1-α, where $t_{\alpha/2}(n-1)$ is such a value that the integral of the t-distribution density function with the *n-1* degree of freedom from $t_{\alpha/2}(n-1)$ to ∞ equals α/2 . Therefore, we have

$$-t_{\alpha/2}(n-1) \le \frac{\bar{r}-\mu}{s/\sqrt{n}} \le t_{\alpha/2}(n-1) \tag{2}$$

Equation (2) can be equivalently expressed as

$$r_4 = \bar{r} - t_{\alpha/2}(n-1)\frac{s}{\sqrt{n}} \le \mu \le \bar{r} + t_{\alpha/2}(n-1)\frac{s}{\sqrt{n}} = r_5 \tag{3}$$

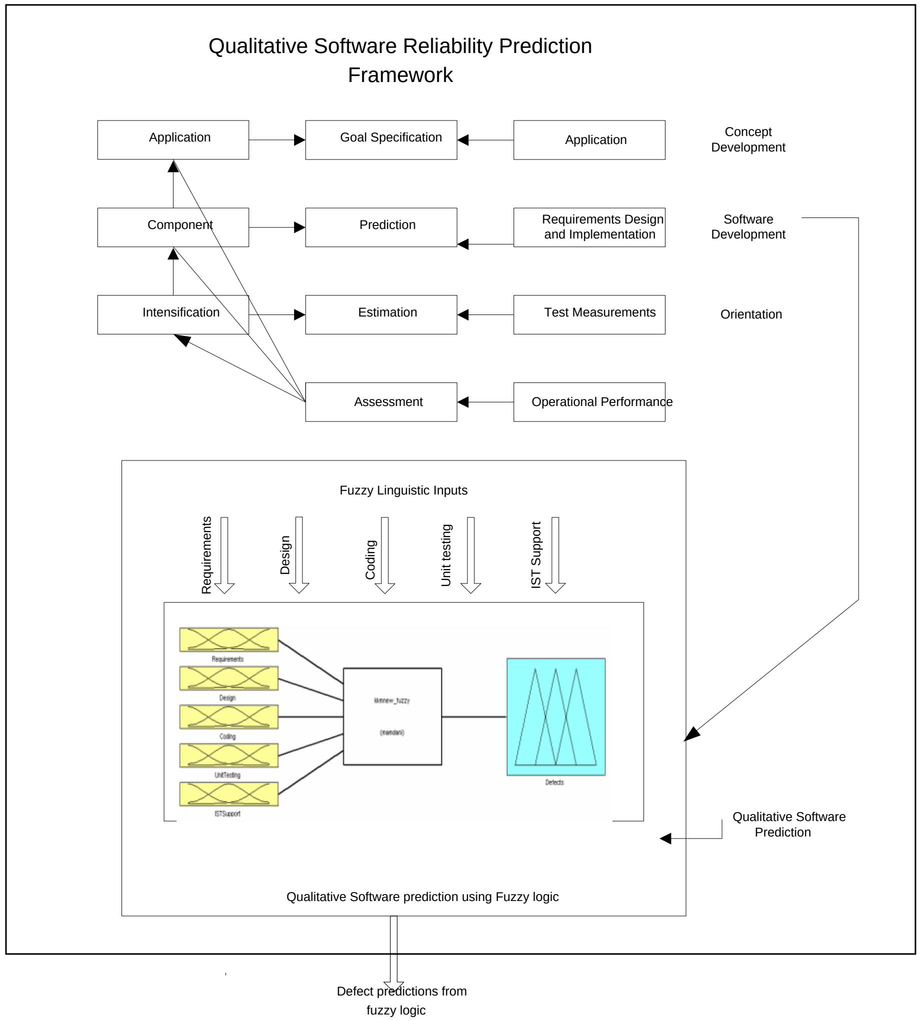

Figure1. Framework for qualitative software reliability prediction

Equation (3) indicates that the real expected effort is located between the lower and higher bounds which are determined by sampled efforts.

The judgements regarding effort of different modules in various phases are listed in table 3. The computed mean and standard deviation of the efforts for the three linguistic descriptors namely “low”, “medium” and “high” are also indicated in the table. From the t distribution for degree of freedom of 5 corresponding to confidence level of 0.2, $t_{0.2}(5)$ is obtained as 2.015. The value of r in the membership function corresponding to 0.2 can be computed as

$$r = 31.67 \pm 2.015 * 26.98/\sqrt{6} = 54$$

Similarly for confidence level of 0.4, $t_{0.4}(5)$ is obtained as 1.476 and the value of r is obtained as 45. Then membership function can be plotted by joining the points using a straight line and extending the line in the interval corresponding to the confidence levels of 0 and 1 as shown in fig.2

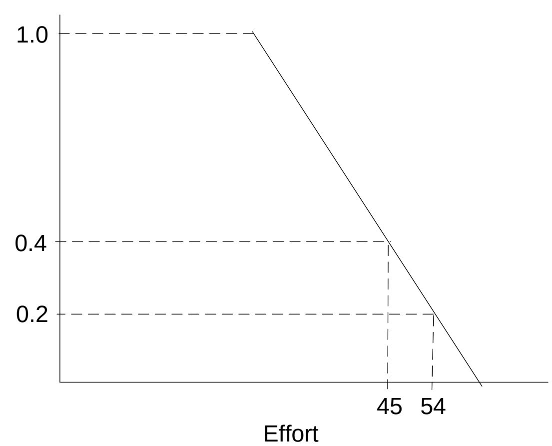

Figure 2. Slope of fuzzy membership function

TABLE II. MEMBERSHIP VALUES FOR VARIOUS INPUT AND OUTPUT FUZZY SET

Membership values for requirements input fuzzy set

Membership function	Membership values
Low	$0,0, \mu-(9\sigma/4), \mu-(3\sigma/4)$
Medium	$\mu-(9\sigma/4), \mu-(3\sigma/4), \mu, \mu+(3\sigma/2)$
High	$\mu, \mu+(3\sigma/2),150,150$

Membership values for design input fuzzy set

Membership function	Membership values
Low	$0,0, \mu-(9\sigma/4), \mu-(3\sigma/4)$
Medium	$\mu-(9\sigma/4), \mu-(3\sigma/4), \mu+(\sigma/4), \mu+(3\sigma/2)$
High	$\mu+(3\sigma/4), \mu+(3\sigma/2),150,150$

Membership values for coding input fuzzy set

Membership function	Membership values
Low	$0,0, \mu-(9\sigma/4), \mu-(6\sigma/4)$
Medium	$\mu-(9\sigma/4), \mu-(6\sigma/4), \mu+(3\sigma/4), \mu+(3\sigma/2)$
High	$\mu+(3\sigma/4), \mu+(3\sigma/2),150,150$

Membership values for unit testing input fuzzy set

Membership function	Membership values
Low	$0,0, \mu-(15\sigma/8), \mu-(5\sigma/8)$
Medium	$\mu-(15\sigma/8), \mu-(5\sigma/8), \mu, \mu+(5\sigma/4)$
High	$\mu+ \sigma$ (9 $\mu+(5\sigma/4)$,250,250

Membership values for IST input fuzzy set

Membership function	Membership values
Low	$0,0, \mu-(3\sigma/2), \mu$
Medium	$\mu-(15\sigma/8), \mu-(5\sigma/8), \mu+(3\sigma/8), \mu+(5\sigma/4)$
High	$\mu+(5\sigma/8), \mu+(9\sigma/8),250,250$

Membership values for number of defects output fuzzy set

Membership function	Membership values
Very Less	$0, 0, \mu-(21\sigma/8), \mu-(20\sigma/8)$
Less	$\mu-(21\sigma/8), \mu-(17\sigma/8), \mu-2\sigma, \mu-(5\sigma/4)$
Medium	$\mu-(15\sigma/8), \mu-(5\sigma/4), \mu-(5\sigma/8), \mu$
High	$\mu-(5\sigma/8), \mu, \mu+(5\sigma/8), \mu+(5\sigma/4)$
Very High	$\mu+(5\sigma/8), \mu+(5\sigma/4)$, 500,500

C. Fuzzy analysis of defect prediction

Based on the input-output combinations, fuzzy rules [6] are created for each of the cycle specifications and using the fuzzy system editor contained in the Fuzzy Logic Tool-box of Matlab. These rules are fed to the fuzzy engine.

The various specifications for various fuzzy input phases in terms of effort measured in person hours and membership values for the key influencing parameters are estimated from the fuzzy sets for cycle 1: requirements = 125% and the corresponding membership values being medium = 0, high = 1; design = 80% and corresponding membership values being medium = 1, high = 0; coding = 70% and corresponding membership values being medium = 1, high = 0, unit testing = 125% and corresponding membership values being medium = 1, high = 0, IST support = 160% and corresponding membership values being low = 0,medium = 1.

For cycle 2: requirements = 75% and the corresponding membership values being medium = 1, high = 0; design = 40% and corresponding membership values being low = 1, medium = 0; coding = 35% and corresponding membership values being low = 1 medium = 0, unit testing = 60% and corresponding membership values being low = 1, medium = 0, IST support = 70% and corresponding membership values being low = 1, medium = 0.

Similarly for cycle 3: requirements = 50% and the corresponding membership values being medium = 1, high = 0; design = 25% and corresponding membership values being low = 1, medium = 0; coding = 30% and corresponding membership values being low = 1 medium = 0, unit testing = 45% and corresponding membership values being low = 1, medium = 0, IST support = 50% and corresponding membership values being low = 1, medium = 0.

In the initial prediction of cycle 1 for the effort spent in terms of the person hours for inputs, the defects predicted are 94 (rounded to the nearest integer).

In the next prediction of cycle 2, for the effort spent in terms of the person hours for inputs, the defects predicted are 47 (rounded to the nearest integer).

In the last prediction of cycle 3, for the effort spent in terms of the person hours for inputs, the defects predicted are 21.

The total number of defects predicted from fuzzy logic for the various cycles are depicted in the table 4.

TABLE III. MEAN AND STANDARD DEVIATION LINGUISTIC VARIABLES

Effort in person-hours	Mean (ŕ)	S.D (s)	Max	Min
Requirements (LOW)	31.66667	26.979	60	0
Requirements (medium)	58.5	27.23784	105	30
Requirements (high)	92.16667	13.81907	150	75
Effort in person-hours	**Mean (ŕ)**	**S.D (s)**	**Max**	**Min**
Design (LOW)	31.66667	27.12686	60	0
Design (medium)	68	37.08908	105	30
Design (high)	104	13.54991	150	90
Effort in person-hours	**Mean (ŕ)**	**S.D (s)**	**Max**	**Min**
Coding (LOW)	30.83333	13.40771	45	0
Coding (medium)	44.83333	13.71739	105	30
Coding (high)	104	13.88524	150	90
Effort in person-hours	**Mean (ŕ)**	**S.D (s)**	**Max**	**Min**
Unit testing (LOW)	52.5	45.58399	100	0
Unit testing (medium)	97.33333	45.28429	175	50
Unit testing (high)	162.3333	45.52655	250	125
Effort in person-hours	**Mean (ŕ)**	**S.D (s)**	**Max**	**Min**
IST (LOW)	68.16667	54.01265	125	0
IST (medium)	97.5	45.38171	175	50
IST (high)	173.6667	22.55364	250	150
Number of defects	**Mean (ŕ)**	**S.D (s)**	**Max**	**Min**
Output(Very Less)	40.33333	9.003703	50	0
Output (Less)	82.83333	40.06204	150	40
Output (medium)	147.8333	45.50787	250	100
Output (high)	247.3333	45.63186	350	200
Output (Very high)	347.5	45.43897	500	300

TABLE IV. DEFECTS PREDICTION FROM FUZZY LOGIC

Cycle	Total number of Defects – Predicted from fuzzy logic
Cycle 1	93
Cycle 2	47
Cycle 3	21

IV. CONCLUSIONS:

Usage of fuzzy logic system acts as a conclusive weighing factor in deciding upon the taking up of a project implementation at module level. In effect, a feasibility study has been conducted, which acts as a significant pointer towards the capture of qualitative reliability (in the form of expected number of defects) well before the beginning of the project. A procedure for taking into account realistic membership values employed in the fuzzy set analysis has been outlined in this paper. This paves the way for building into applications the all important QoS considerations. The proposed approach is portable and can be applied to any of the process oriented software development in the parlance of both heterogeneous and non heterogeneous landscapes. Validation of the proposed approach has also been obtained by comparing the predicted results with the ones obtained on realized prototypes at module level.

REFERENCES:

[1] Rational Unified Process: http://www.ibm.com

[2] Hans, Westerheim., Geir Kjetil, Hanssen.,2005. “The Introduction and Use of a Tailored Unified Process”, IEEE Software, pp. 196-203.

[3] Bezdek, J, D. “Editorial: Fuzzy Models – What are They, and Why?” 1993. IEEE Transactions on Fuzzy Systems, 1(1)
[4] Wenyuan Li, Jiaqi Zhou, Kaigui Xie, and Xiaofu Xiong, 2008. “Power System Risk Assessment Using a Hybrid Method of Fuzzy Set and Monte Carlo Simulation”, IEEE transactions on power systems, 23(2) pp.336-343.
[5] J. E. Freund, Mathematical Statistics. Englewood Cliffs, NJ: Prentice-Hall, 1962.
[6] Riza C. Berkan, Sheldon L. Trubatch, 1977. “Fuzzy Systems Design Principles”, Building Fuzzy IF-THEN Rules Bases; IEEE PRESS, pp. 70.