Recent Review and Current Issues in Software Reliability Growth Models under Fuzzy Environment

Seema Rani
University Department of Statistic and Computer Applications,
T. M. Bhagalpur University, Bhagalpur

Ishrat Jahan Ara
University Department of Statistic and Computer Applications,
T. M. Bhagalpur University, Bhagalpur

N. Ahmad
University Department of Statistic and Computer Applications,
T. M. Bhagalpur University, Bhagalpur

Abstract- Today computer is essential part of our lives and software is heart of computer. Without software we can't imagine about computer and it is not useful for us. So it is very necessary to produce reliable software at less cost and deliver on appropriate time. In any business environment the ultimate objective is to derive a quality product. Among the various quality characteristics, reliability is considered as the most predominant characteristics. Software Reliability is one of the important factors of software quality. Before software delivered in to market it is thoroughly checked and errors are removed. Every software industry wants to develop software that should be error free and deliver product in to market on appropriate time with less cost. Software Reliability Growth Models (SRGM) helping the software industries to develop software which is error free and reliable. Software development and testing is human dependent and as human behavior is fuzzy, it is better to use fuzzy set theory for developing software reliability models and study SRGM under fuzzy environment. Many models have been utilized for assessing the quality of software using reliability in fuzzy environment. This paper presents a detailed study of existing software reliability growth models under fuzzy environment. We also discuss the recent trends and current issues in software reliability growth models with testing effort under fuzzy environment.

Keywords- Software testing, Software reliability growth model, Testing effort function, fuzzy environment, Fuzzy set theory.

I. INTRODUCTION

Software reliability is about to define the stability or the life of software system with different properties. These properties include the trustfulness of software system, software cost, execution time, software stability etc. The aspects related to these software system includes the probability of software faults, frequency of fault occurrence, criticality of fault, associated module with respective fault etc.. Software reliability can be defined as the probability of failure-free operation of computer program for a specified period of time in a specified environment (Musa et al., 1987).

SRGM is a tool that measures software reliability for the developer during the testing and operational phases of the software (Kapur et al., 2007). The models applicable to the assessment of software reliability are called SRGMs. SRGMs are useful for estimating how software reliability improves as faults are detected and repaired. It can be used to predict when particular level of reliability is likely to attain a given reliability level. SRGMs help in decision making in many software development activities such as number of initial faults, failure intensity, reliability within a specified analysis and release time etc. To achieve highly reliable software systems, many software fault detection/removal techniques can be used by programmers or testing teams. In applying these techniques, the SRGM are important, because they can provide quite useful information for developers and testers during the testing/debugging phase. SRGM is a mathematical expression of the software error occurrence and the removal process. Since the early 1970`s many SRGMs have been proposed (Musa et al., 1987). A NHPP as the stochastic process has been widely used in SRGM. In the past years, several SRGMs based on NHPP which incorporates the testing effort function (TEF) have been proposed by many authors (Ahmad et al., 2008; 2009; 2010 2011; Bokhari and Ahmad, 2006; 2014; Huang and Kuo, 2002; Yamada et al., 1986; 1993, Imam and Ahmad, 2015; Islam and Ahmad, 2015;). This testing-effort can be represented as the number of CPU hours, the number of executed test cases etc.

The quality of software depends on various factors, which are characterized by a few attributes; such as reliability, availability, safety, integrity, development cost, and release time (Dimov and Punnekkat, 2010). If software is tested for a long time before release, obviously, the quality of software increases. On the other hand, software testing cost will increase and market value may go down. There is a risk to release a software system with low quality; therefore, it is necessary to decide the optimal time of software release. Release time of software depends upon different type of costs (e.g., development cost, testing cost, etc.), reliability, efficiency, and mean number of faults present in the software system. In general, mean number of faults are estimated using software reliability growth models (SRGMs). There are different types of SRGMs e.g., probability based, fuzzy based, and neuro-fuzzy based (Yamada et al. 1993; Utkin et al. 2002). In probabilistic SRGMs, parameters (e.g., total number of faults, detection rate, total effort expenditure, etc.) are usually estimated using Maximum Likelihood Estimation (MLE), least square estimation (LSE), etc. (Pachauri et al. 2013). Chatterjee et al. (2011, 2013) discussed the application of fuzzy time-series in prediction of time between failures and faults in software reliability assessment and proposed a software reliability allocation model incorporating the user view point about various functions of software.

In software reliability, most of the models so far developed are probabilistic. All these probabilistic models have various draw backs due to different assumptions made in all these models. Software development and testing is human dependent and as the human behavior is fuzzy, it is better to use fuzzy set theory for developing software reliability models. Due to several problems like system complexity, cutthroat competitions, varied market requirements and many other reasons, the software developers are able to provide ambiguous estimates on the available resources and their aspirations, which ultimately give rise to uncertainty (fuzziness) in the problem formulation. In such situations, crisp optimization technique may not serve the purpose to quantify the parameters. Defining the problem under fuzzy environment provides a better platform to quantify these uncertainties (Jha et al., 2010).

II. RELATED WORK

The concept of fuzzy reliability has been proposed by several authors (Onisava and Kacprzyk., 1995; Cai et al., 1995). The conventional reliability is considered under the probability and binary-state assumptions. Cai el al., (1995) have given a different insight by introducing the possibility and fuzzy-state assumptions to replace the probability and binary-state assumptions. According to Cai et al., (1995) various forms of fuzzy reliability theories, including profust, posbist, and posfust reliability models, can be considered by taking new assumptions, such as possibility or fuzzy-state, instead of probability or the binary-state assumption. In the conventional systems, we always give an exactly failed or functioning probability for each component.

However, in practice, when the stress or the strength or both of them are fuzzy variables, it is very difficult to compute the exact value for each component. So, currently, by investigating the fuzzy reliability of a system, the researchers always assume that reliability of each component is a fuzzy variable (Mon et al., 1994; Utkin et al., 1995, 1996). The design of fuzzy logic system (FLS) includes the design of rule base, input scaled factors, output scaled factors, and membership functions.

Hardy (1995) presented a new approach on fuzzy reliability, based on the use of beta type distribution as membership function. Chatterjee et al., (1998) have present a methodology for software reliability estimation incorporating the concept of fuzzy test effort and learning factor.

Utkin et al. (2002) proposed a fuzzy software reliability model where the time intervals between the software failures are taken as the fuzzy variables governed by membership function.

Kapur et al. (2007) first proposed a testing efficiency model incorporating the effect of imperfect fault, debugging and error generation. Then by using this model, a software reliability growth model (SRGM) was developed to model the reliability growth of an NVP system. Singh (2007) presented a conference paper, in this paper the optimal release policies for software system are discussed maximizing expected gain and simultaneously achieving a given level of failure intensity incorporating the effect of testing effort expenditure while fixing intensity constraint.

Simona and Cristina (2009) have present fuzzy simulation in reliability analysis. Donighi and Mohammadi, (2011) defined fuzzy based reliability model for the software system. Fuzzy set based methods have been proved to be effective in handling many types of uncertainties in different fields, including reliability engineering.

Chawla et al. (2012) presents a work which is the solution of the problem of large software system with thousands of class modules. In this work software quality is estimated by using rejection method on software faults. The rejection method is applied on the basis of Fuzzy Logic in a software system.

Kapur et al. (2011) have formulated software release time decision problems under fuzzy environment and have discussed the solution methodology based on fuzzy set theory and fuzzy optimization through numerical illustrations. Jha et al. (2011) have formulated a fuzzy bi-criterion release time problem optimizing the two conflicting cost minimization and reliability maximization objectives under budget constraint. Fuzzy optimization technique is discussed to solve such problem with numerical illustration.

Madsen et al. (2012) illustrated the usage of intuitionistic, fuzzy degree approach in modeling the quality of entities in imprecise software reliability computing in order to optimize management results. Intuitionistic fuzzy optimization algorithms are proposed to be used for complex software systems reliability optimization under various constraints.

Pachauri et al. (2013) calculated the reliability, and optimal time of software release based on cost-reliability criterion using fuzzy set theory. Pandey and Goyal (2013) discussed an approach for early prediction of number of faults in software and evaluation of reliability based on fuzzy expert system. Kiruthiga and Loganathan, (2014) presents a case study for evaluating software reliability on the OFBiz project software package used on business routines. They also explain a new model that can have more accurate analysis and forecast to software assumptions. Jha et al. (2000) presented interactive approach to release time problem of software under fuzzy environment.

Besides the above described SRGM under fuzzy environment, there are many SRGMs have been discussed over the last two decades. Analogy-based estimation with fuzzy numbers have been integrated in order to improve the performance of software projects during the early stages of a software development life-cycle, communication skills, personality, mood, and capability of team members has been developed under fuzzy paradigm (Kazemifard et al. 2011).

III. SOFTWARE RELIABILITY MODELS AND OPTIMAL RELEASE POLICY UNDER FUZZY ENVIRONMENT

1. Fuzzy set: The concept of a fuzzy set is an extension of the concept of a crisp set. Just as a crisp set on a universal set *U* is defined by its characteristic function from *U* to {0,1}, a fuzzy set on a domain *U* is defined by its membership function from *U* to [0,1].

Let *U* be a non-empty set, to be called the universal set or the universe of discourse or simply a domain. Then, by a fuzzy set on *U* is meant a function $\mu_{\tilde{A}} : X \to [0,1]$. $\mu_{\tilde{A}}$ is called the membership function, $\mu_{\tilde{A}}(x)$ is called membership grade of *x* in $\mu_{\tilde{A}}$. We also write

$$\mu_{\tilde{A}} = \{(x, \mu_{\tilde{A}}(x)) : x \in U\}$$

We may sometimes represent the unit interval [0, 1] by *I*. The following example illustrates the concept of fuzzy sets. Example:

Consider *U* = {*a, b, c, d*} and *A: U*→ *I* defined by *A*(*a*) = 0, *A*(*b*) = 0.7, *A*(*c*) = 0.4 and *A(d)* = 1. Then *A* is fuzzy set on *U*. This fuzzy set can also be written as follows

$$A = \{(a, 0), (b, 0.7), (c, 0.4), (d, 1)\}$$

2. *α*-cuts*:* In fuzzy set theory the concept of α-cut is the most significant and widely used. The α-cut of a fuzzy set *X*, is a crisp set denoted by X^{α} in which all the elements x of a universe of discourse *U* have the membership value greater than or equal to α, mathematically it is expressed as:

$$X^{\alpha} = \{x \in X | \mu_x(x) \geq \alpha\}, \quad (1)$$

3. *Fuzzy numbers and arithmetic operations on fuzzy numbers:*

A fuzzy set $\tilde{A}$, defined on the universal set of real number *R*, is said to be a fuzzy number if its membership function has the following characteristics:

(i) $\tilde{A}$ is convex i.e., $\mu_{\tilde{A}}(\lambda x_1 + (1-\lambda)x_2) \geq \min(\mu_{\tilde{A}}(x_1), \mu_{\tilde{A}}(x_2)) \ \forall x_1, x_2 \in R, \ \forall \lambda \in [0,1]$
(ii) $\tilde{A}$ is normal i.e., $\exists x_0 \in R$ such that $\mu_{\tilde{A}}(x_0) = 1$
(iii) $\mu_{\tilde{A}}$ is piecewise continuous.

A **fuzzy number** is a quantity whose value is imprecise, rather than exact as is the case with "ordinary" (single-valued) numbers.

There exist various types of fuzzy numbers in literature, e.g., triangular fuzzy number (TFN), trapezoidal fuzzy numbers, etc.

4. Triangular Fuzzy Number: A fuzzy number $\tilde{A} = (p, q, r)$ is said to be a triangular fuzzy number if its membership function is given by, where $p \leq q \leq r$.

$$\mu_{\tilde{A}}(x) = \begin{cases} 0; & x < p, \\ \frac{x-p}{q-p}; & p \leq x \leq q, \\ \frac{r-x}{r-q}; & q \leq x \leq r, \\ 0; & x > r, \end{cases} \tag{2}$$

Fuzzy arithmetic operations on two fuzzy numbers depend on two basic properties of fuzzy numbers, which enables us to define arithmetic operations on fuzzy numbers in terms of arithmetic operations on their α-cuts (Klirr and Yuan 1995).

a. Each fuzzy number can fully and uniquely be represented by its α-cuts.
b. α-cuts of each fuzzy numbers are closed intervals of real numbers for all $\alpha \in (0,1]$.

Let $P_\alpha = [p_l^\alpha, p_r^\alpha]$ and $Q_\alpha = [q_l^\alpha, q_r^\alpha]$ be any two fuzzy intervals then arithmetic operations on them are shown in table 1.

Table 1: Fuzzy number operations.

Addition	$P + Q = [p_l^\alpha + q_l^\alpha, \ p_r^\alpha + q_r^\alpha]$
Subtraction	$P - Q = [p_l^\alpha - q_l^\alpha, \ p_r^\alpha - q_r^\alpha]$
Multiplication	$P * Q = [p_l^\alpha * q_l^\alpha, \ p_r^\alpha * q_r^\alpha]$
Division	$P / Q = [p_l^\alpha / q_l^\alpha, \ p_r^\alpha / q_r^\alpha]$

5. *Fuzzification:* The process of transforming a crisp variable into a fuzzy set is called fuzzification. In general, fuzzification of a real-valued variable is done by intuition, experience and analysis of the set of rules and conditions associated with the input variables. There are generally three types of fuzzifiers which are used for fuzzification, namely, (a) singleton fuzzifier, (b) Gaussin fuzzifier, and (c) trapezoidal or triangular fuzzifier.

6. *Defuzzification:* Defuzzification is the process of producing a quantifiable result in fuzzy logic, given fuzzy sets and corresponding membership degrees. There are several types of defuzzification techniques, constraint decision defuzzification, center of gravity, mean of maxima, weighted fuzzy mean, center of sum, and center of largest area which are used, depending on the application, Center of gravity method for defuzzification.

IV. SOFTWARE RELIABILITY MODEL WITH TESTING EFFORT UNDER FUZZY ENVIRONMENT

Pachauri et al. (2013) has discussed a software optimal release time with cost-reliability criteria in an imperfect debugging environment with Generalized Modified Weibull distribution. The motive of their study is to model uncertainty involved in estimated parameters of the software reliability growth model (SRGM). Method: Initially the reliability parameters of SRGM are estimated using least square estimation (LSE). Considering the uncertainty involved in the estimated parameters due to human behavior being subjective in nature and the dynamism of the testing environment, the concept of fuzzy set theory is applicable in developing SRGM. Finally, using arithmetic operations on fuzzy numbers, the reliability and total software cost are calculated. Result: Various reliability measures have been computed at different levels of uncertainties, and a comparison has been made with the existing results reported in the literature. Conclusion: It is evident from the results that a better prediction of reliability measures, namely, software reliability and total software cost can be made under the fuzzy paradigm.

Pachauri et al.'s work is motivated from Verma et al. (2007). The work of Verma et al. is discussed below in algorithmic form:

- Used probabilistic method for formulating the model.
- Estimated the involved parameters and collect data.
- Used triangular fuzzy numbers to fuzzify the parameters.
- To find out the fuzzy reliability apply arithmetic operation on fuzzy numbers.
- Defuzzify the results.

Step by step details of the work is as follows:

Step 1: Model formulation
The mathematical formulation of the model is given as:

$$\frac{dm(t)}{dt}\cdot\frac{1}{w(t)} = r\big(a(t) - m(t)\big), \quad (3)$$

$$\frac{da(t)}{dt} = \gamma\frac{dm(t)}{dt}, \quad (4)$$

where $m(t), w(t), a(t), r$ and γ are mean value function, testing effort function, total number of faults, detection rate and probability of introducing the new faults respectively. After solving the above equations, we get the mean value function *m(t)*:

$$m(t) = \frac{a}{1-\gamma}\left(1 - e^{-r(1-\gamma)W(t)}\right), \quad (5)$$

where a, r, γ and w(t) are total number of faults, detection rate, probability to introduce new fault and testing effort function respectively. Then the failure

$$\lambda(t) = \frac{dm(t)}{dt} = \frac{a r w(t) e^{-r(1-\gamma)W(t)}}{1-\gamma}. \quad (6)$$

The expected number of errors detected at the long run is given as:

$$m(\infty) = \frac{a}{1-\gamma}. \quad (7)$$

And finally the conditional reliability function is:

$$R(t + \Delta t|t) = e^{-(m(t+\Delta t)-m(t))}. \quad (8)$$

Step 2: Data collection and parameter estimation

To verify the model, they used the secondary data which have been collected from literature. The parameters, namely, total effort expenditure, total number of faults, detection rate, probability of introducing new faults, accelerating factor, scale and shape parameters of Generalized Modified Weibull distribution, are estimated by using nonlinear least square estimation through SPSS.

Step 3: Fuzzification of parameters
To minimize the uncertainty involved in estimated parameters, triangular fuzzy numbers have been used. The fuzzification process can be understood by an example given below:

The other parameters are converted in the form of triangular fuzzy numbers. Fuzzy representation of the parameters is shown in Table 2.

Table 2. Fuzzy representation of parameters using triangular fuzzy numbers.

Parameters	Fuzzy Representation
s	(s_1, s_2, s_3)
β	$(\beta_1, \beta_2, \beta_3)$
λ	$(\lambda_1, \lambda_2, \lambda_3)$
m	(m_1, m_2, m_3)
θ	$(\theta_1, \theta_2, \theta_3)$
a	(a_1, a_2, a_3)
γ	$(\gamma_1, \gamma_2, \gamma_3)$
r	(r_1, r_2, r_3)

Step 4: Basic operations

Let $\tilde{A} = [p, q, r]$ be any triangular fuzzy number, then the α-cut concept says that it can be characterized by the interval of confidence at the α-cut level such that:

$$A_\alpha = [p^\alpha, r^\alpha] = [p + (q - p)\alpha, r - (r - q)\alpha], \quad \forall \alpha \in [0, 1]. \tag{9}$$

By using basic operations on triangular fuzzy numbers, coupled with α-cuts, the failure intensity for data set (Musa et al., 1987) is considered, for testing time $t = 10$ weeks, at different presumption levels.

Step 5: Defuzzification

Since, most of the decisions implemented by the researchers are in binary form, therefore, there is a need to defuzzify the fuzzy outputs. There exist several defuzzification techniques in literature, e.g., adaptive integration, basic defuzzification distributions, constraint decision defuzzification, center of gravity, mean of maxima, weighted fuzzy mean, center of sum, and center of largest area which are used, depending on the application. Center of gravity method for defuzzification is widely used and is suitable for our study, which is defined as:

$$COG_{out} = \frac{\int_{x_1}^{x_2} x\mu_{out}(x)\,dx}{\int_{x_1}^{x_2} \mu_{out}(x)\,dx}, \tag{10}$$

V. CURRENT ISSUES AND FUTURE WORKS

The following may be the areas of current issue and future works:

1) Pachauri et al. (2013) has developed the software reliability growth model and total software cost estimation by using triangular fuzzy number with Generalized Modified Weibull distribution, trapezoidal fuzzy number may be used to develop software reliability and total software cost estimation.
2) Exponetiated Weibull Software Reliability Growth Model (Ahmad et al. 2008) may be extended under fuzzy environment by using triangular and trapezoidal fuzzy numbers.

Trapezoidal Fuzzy Number may be formulated as:

A fuzzy number $\tilde{A} = (p, q, r, s)$ is said to be a trapezoidal fuzzy number if its membership function is given by, where $p \le q \le r \le s$.

$$\mu_A(x) = \begin{cases} 0; & x \le p, \\ \dfrac{x-p}{q-p}; & p \le x \le q, \\ 1; & q \le x \le r, \\ \dfrac{s-x}{s-r}; & r \le x \le s, \\ 0; & x \ge s, \end{cases}$$

Fuzzification of parameters: To minimize the uncertainty involved in estimated parameters, trapezoidal fuzzy numbers have been used. The fuzzification process can be understood by an example given below:

Example: Let the crisp value of a parameter '*a*' be 407.506 which is called the initial number of faults. By intuition, let there be 1% uncertainty in this value, and then, we add and subtract 1% of this value from the original value. For calculating fourth value which is right of the original value, we add 1% of original value to its just previous value (i.e. 411.58106 + (1 * 407.506 / 100) = 415.5065).

Now, a trapezoidal fuzzy number *X* may be defined as follows, based on aforesaid procedure:

$$\mu_A(x) = \begin{cases} 0, & x \le 403.43094 \\ \dfrac{x - 403.43094}{407.506 - 403.43094}, & 403.43094 \le x \le 407.506 \\ 1, & 407.506 \le x \le 411.58106 \\ \dfrac{415.65612 - x}{415.65612 - 411.58106}, & 411.58106 \le x \le 415.65612 \\ 0, & x \ge 415.65612 \end{cases}$$

The other parameters of the model are converted in the form of trapezoidal fuzzy numbers. Fuzzy representation of the parameters is shown in Table 3.

Table 3. Fuzzy representation of parameters using trapezoidal fuzzy numbers.

Parameters	Fuzzy Representation
s	(s_1, s_2, s_3, s_4)
β	$(\beta_1, \beta_2, \beta_3, \beta_4)$
λ	$(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$
m	(m_1, m_2, m_3, m_4)
θ	$(\theta_1, \theta_2, \theta_3, \theta_4)$
a	(a_1, a_2, a_3, a_4)
γ	$(\gamma_1, \gamma_2, \gamma_3, \gamma_4)$
r	(r_1, r_2, r_3, r_4)

Basic operations

Let $A = [a, b, c, d]$ be any triangular fuzzy number, then the α-cut concept says that it can be characterized by the interval of confidence at the α-cut level such that:

$$A_\alpha = [a^\alpha, d^\alpha] = [a + (b-a)\alpha, c + (c-b)\alpha, \; d + (d-c)\alpha], \quad \forall \alpha \in [0, 1].$$

3) Some studies have shown that fuzzy logic systems (FLS) performance is more dependent on membership function design than rule base design (Cordon et al., 2000). The future work can be done how the performance of FLS is depend on membership function design and how much depend on rule base design separately.
4) Researcher can also discuss the dependency of FLS performance on membership function that how much it is greater than the dependency of FLS performance on rule base design in percentage or in average.

5) Pachauri et al. (2013) has discussed a software optimal release time with cost-reliability criteria in an imperfect debugging environment with generalized modified Weibull testing effort function. This work may be extended by incorporating other testing effort function such as log-logistic, Burr type XII or generalized exponential in future.
6) Future research may involve various other techniques such as neural networks, machine learning techniques, to predict software reliability growth models and optimal release policy under fuzzy environment.
7) Also the model developed by Utkin et al. (2002) can be extended in many directions.

VI. CONCLUSION

In this paper we have investigated the software reliability growth models and optimal release policy under fuzzy environment. Since human judgment is naturally indefinite and the software development is performed by humans, so this process is inherently associated with uncertainty in the estimation of effort or cost requirement for software testing. This paper has suggested many current issues in software reliability growth models under fuzzy environment. It also suggested many extensions on the previous work in terms of software reliability modeling, reliability estimation, optimal release policy, testing effort functions and software testing under fuzzy environment. This study may provide new directions to the researcher in these areas.

REFERENCES

[1] P. C. Jha, Indumati Sigh, P. K. Kapur, Interactive Approach to Releae time problem of Software under Fuzzy environment, Communication in Dependability and quality Management, Vol. 13, No. 3, pp. 61-75, 2000.
[2] N. Ahmad, M. U. Bokhari, S. M. K, Quadri, and M. G. M. Khan, The Exponetiated Weibull Software Reliability Growth Model with Various Testing-Efforts and Optimal Release Policy: A Performance Analysis, *International Journal of Quality and Reliability Management*, Vol. 25 (2), pp. 211 – 235, 2008.
[3] N. Ahmad, M. G. M. Khan, S. M. K, Quadri, and M. Kumar, Modeling and Analysis of Software Reliability with Burr type X testing-effort and release-time determination, Journal of Modeling in Management, Vol. 4, No. 1, pp. 28-54, 2009.
[4] N. Ahmad, M. G. M. Khan, and L. S. Rafi, A Study of Testing-Effort Dependent Inflection S-Shaped Software Reliability Growth Models with Imperfect Debugging, International Journal of Quality and Reliability Management, Vol. 27 (1), pp. 89 – 110, 2010.
[5] N. Ahmad, M. G. M. Khan, and L. S. Rafi, Analysis of an Inflection S-shaped Software Reliability Model Considering Log-logistic Testing-Effort and Imperfect Debugging, International Journal of Computer Science and Network Security, Vol. 11 (1), pp. 161 – 171, 2011.
[6] M. U. Bokhari, N. Ahmad, Analysis of a Software Reliability Growth Models: the case of log-logistic test-effort function, Proceedings of the 17th IASTED International Conference on Modeling and Simulation, pp. 540-545, 2006.
[7] M. U. Bokhari, N. Ahmad, Incorporating Burr Type XII Testing-efforts into Software Reliability Growth Modeling and Actual Data Analysis with Applications", *Journal of Software,* Vol. 9 (6), pp. 1389-1400, 2014.
[8] M. Z. Imam and N. Ahmad, "Modeling and Analysis of Fault Detection and Correction Processes in Software Reliability Growth with Exponentiated Weibull Testing Effort Function", Computer Science and Engineering: Recent Trends (ISBN 978-81-8487-391-7) Editors: R. Rajesh and P. Ranjan, Narosa Publishing House, New Delhi, pp. 55-63, 2015.
[9] S. F. Islam, and N. Ahmad, "Incorporating Change Point into Exponentiated Weibull Software Reliability Growth Model and Actual Data Analysis", Computer Science and Engineering: Recent Trends (ISBN 978-81-8487-391-7) Editors: R. Rajesh and P. Ranjan, Narosa Publishing House, New Delhi, pp. 229-237, 2015.
[10] C. Y. Huang, S. Y. Kuo, Analysis of incorporating logistic testing – effort function into software reliability modeling, IEEE Transactions on Reliability, Vol. 51, No. 3, pp. 261 – 270, 2002.
[11] J. D. Musa, A. Lannino, K. Okumoto, Software Reliability Measurement, Prediction and Application, McGraw – Hill, 1987.
[12] S. Yamada, J. Hishitani, S. Osaki, Software Reliability growth model with Weilbull testing – effort : a model and application, IEEE Transactions on Reliability, Vol. R-42, pp. 100-105, 1993.
[13] S. Yamada, H. Ohtera, H. Norihisa, Software Reliability Growth Model with testing – effort, IEEE Transactions on Reliability, Vol. R-35, No. 1, pp. 19-23, 1986.
[14] Bhoopendra Pachauri, Ajay Kumar, Joydip Dhar, Modeling Optimal release policy under fuzzy paradigm in imperfect debugging environment, Information and Software Technology, Vol. 55, Issue 11, pp. 1974-1980, 2013.
[15] M. Kiruthiga, C. Loganathan, Software Reliability Modeling in Fuzzy Environment, Progress in Nonlinear Dynamics and Chaos, Vol. 2, No. 1, 2014.
[16] P. K. Kapur, Anshu Gupta, P. C. Jha, Reliability Growth Modeling and Optimal Release Policy under Fuzzy Environment of N-version Programming System Incorporating the Effect of Fault Removal Efficiency, International Journal of Automation and Computing, Vol. 04, No. 4, pp. 369-379, 2007.
[17] Lev V. Utkin, Sergey V. Gurov, Maxim I. Shubinky, A Fuzzy software reliability model with multiple-error introduction and removal, International Journal of Rel. Qual. Sof. Eng., Vol. 09, Issue 3, page 215, 2002.
[18] S. Chatterjee, S. Nigam, J. B. Singh, L. N. Upadhyaya, Application of fuzzy time series in prediction of time between failures & faults in software reliability assessment, Fuzzy information and engineering, Vol. 3, Issue 3, pp. 293-309, 2011.
[19] S. Chatterjee, Jeetendra B. Singh, Arunava Roy, A structure-based software reliability allocation using fuzzy analytic hierarchy process, International Journal of Systems Science, Vol. 46, No. 3, pp. 513-525, 2013.
[20] P. C. Jha, Shivani Bali, U. Dinesh Kumar, A Fuzzy approach for optimal selection of COTS components for modular software system under consensus recovery block scheme incorporating execution time, Turkish Journal of Fuzzy Systems, Vol. 2, No. 1, pp. 45-63, 2011.

[21] Henrik Madsen, Grigore Albeanu, Florin Popentiu Vladicescu, An Intuitionistic Fuzzy Methodology for Component-Based Software Reliability Optimization, International Journal of Performability Engineering, Vol. 8, No. 1, pp. 67-76, 2012.
[22] P. C. Jha, Indumati, Ompal Singh, Deepti Gupta, Bi-Criterion Release Time Problem for a Discrete SRGM under Fuzzy Environment, International Journal of Mathematics in Operational Research, Vol. 3, No. 6, pp. 680-696, 2011.
[23] A. K. Pandey, N. K. Goyal, A fuzzy model for early software fault prediction using process maturity and software metrics, International journal of Electronics Engineering, /vol. 1, No. 2, page 239-245, 2009.
[24] P. K. Kapur, Hoang Pham, Anshu Gupta, P. C. Jha, Optimal Release Policy under Fuzzy Environment, International Journal of System Assurance Engineering and Management, Vol. 2, No. 1, pp. 48-58, 2011.
[25] Ashok Kumar Singh, Optimal Release Policy under Fuzzy Environment International Conference on Advances in Interdisciplinary Statistics and Combinatorics, 2007.
[26] Terry L. Hardy, Multi-Objective Decision-making under Uncertainty: Fuzzy Logic Methods, National Aeronautics and Space Administration, Lewis Research Center, pages 16, 1994.
[27] S. Chatterjee, R. B. Misra, S. S. Alam, Analysis of software reliability using fuzzy Test-Effort, Journal of Fuzzy Mathematics, Vol. 6, No. 1, pp. 395-404, 1998.
[28] Gitika Chawla, Sanjay Kumar, Deepak Goyal, Pankaj Gupta, An Improved Fuzzy Model to Predict Software Reliability, International Journal of Computer Science & Management Studies, Vol. 12, Issue 03, pp. 44-48, 2012.
[29] S. Sardar Donighi, S. Khan Mohammadi, A fuzzy reliability model for series-parallel systems, International Journal of Ind. Eng.Vol. 7, No. 12, pp. 10-18, 2011.
[30] A. Dimov, S. Punekkat, Fuzzy Reliability model for component based software systems in software engineering and advanced applications, IEEE, pp. 39-46, 2010.
[31] G. J. Klirr, B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, prentice-Hall of India Private limited, New Delhi, 1995.
[32] A. K. Verma, A. Srividya, R. S. P. Goankar, Fuzzy-Reliability Engineering, Concepts and Applications, Narosa Publishing house, 2007.
[33] P. C. Jha, shivani Bali, P. K. Kapur, BVICAM's International Journal of Information Technology, Vol. 3, No. 1, 2010.
[34] K. Y. Cai, C. Y. Wen, M. L. Zhang, Postbist Reliability behavior of faultolerant sysyems, Microelectron Reliab, Vol 35, pp 49-56, 1995.
[35] M. D. L. Mon, C. H. Cheng, Fuzzy System Reliability analysis for components with different membership functions, Fuzzy Sets System, Vol. 6, pp. 145-157, 1994.
[36] T. Onisawa, J. Kacprzyk, Reliability and safety analysis under fuzziness, Heidelberg: Physica-Verlag, 1995.
[37] L. V. Utkin, S. V. Gurov, A general formal approach for fuzzy reliability analysis in the possibility aontext, Fuzzy Sets Systems, Vol. 83, pp. 203-213, 1996.
[38] L. V. Utkin, S. V. Gurov, I. B. Shubinsky, A method to solve fuzzy reliability optimization problem, Microelectron Reliab, Vol. 35, No. 2, pp. 171-181, 1995.