Analysis of Fuzzy Software Reliability Growth Model and Optimal Release Policy with Log-logistic Testing Effort under Imperfect Debugging

Seema Rani and Nesar Ahmad

University Department of Statistics and Computer Applications, T. M. Bhagalpur University, Bhagalpur-812007, India

Summary
This paper discusses fuzzy software reliability growth model under imperfect debugging environment. Log-logistic testing effort function (LLTEF) is incorporated into software reliability growth model (SRGM). The parameters of LLTEF and SRGM are estimated by using least square and maximum likelihood methods. Also, uncertainties involved in estimated parameters of proposed SRGM due to human conduct are investigated for trapezoidal fuzzy numbers. Computational analysis is performed at different level of uncertainties to compute various measures of software reliability. The results obtained are then comparing with other existing SRGM. It is shown that the proposed SRGM with LLTEF gives reasonable prediction of software reliability under the fuzzy model. Further, under optimal release policy, software cost based on reliability criteria is calculated under the fuzzy condition.

Key words:
Software Reliability Growth Model, LLTEF, Fuzzy Number, Fuzzification, defuzzification

1. Introduction

In real world complexity arises from uncertainty and uncertainty exists wherever and whatever human beings interact with real world. It arises in any situation in which a person does not qualitatively possess the appropriate information to describe its behavior or other phenomena deterministically and numerically. In the late 19th century, the modern view of uncertainty began from transition from the traditional view. In first stage of transition, Newtonian mechanism, which involves only certainty, is replaced by probability theory. In second stage of transition, the modern view of uncertainty started in 1965 when Zadeh (1965) published a paper on fuzzy sets. Probability theory captured uncertainty of a certain type and this notion for uncertainty is challenged by Zadeh's paper. Everywhere, uncertainty exists except in idealized situations. It is not only an unavoidable and ubiquitous phenomenon, but it is also a fundamental scientific principle (Gong et al., 2014).

It is well known that the customers always need software with zero defects with specified time and minimum cost. That's why estimating reliability of software system is more importance. Software reliability growth model (SRGM) is a model which are developed or designed to predict and estimate the software reliability of a system. There are various software reliability growth models have been developed in past decades considering different assumptions and environment. There are various parameters which characterize the development of software. Each parameter contains certain level of fuzziness. So we need to introduce some degree of uncertainty involved in software development, to make the model reliable.

Software reliability growth model (SRGM) is a tool for measuring software reliability during the operational and testing phases of the software (Kapur et al., 2007). In the literature, there are many types of SRGM, like SRGM based on probability, based on fuzzy number, and based on neuro-fuzzy (Musa, 1971; Utkin et al., 2002; Kiran and Ravi, 2007). In several decades, various non-homogeneous Poisson process (NHPP) SRGMs including the optimal release policy have been investigated by the researchers (Goel and Okumoto, 1979; Musa et al., 1987; Yamada et al., 1984). Further many authors have been proposed SRGMs with testing-effort, optimal release policy, and imperfect debugging (Yamada et al., 1986; 1993; Huang and Kuo, 2002; Ahmad et al., 2008; 2009; 2010; 2011; Bokhari and Ahmad, 2006; 2014; Imam et al., 2016; Quadri et al., 2011; Khan et al., 2016;). Rafi et al. (2010) discussed SRGM with testing effort and optimal release policy under an imperfect debugging.

We know very well that there are uncertainties in software reliability estimation, cost estimation, effort estimation and risk analysis in software development process. Software testing is done by human and the human behavior is fuzzy. Therefore it would be better to use fuzzy number theory for the development of SRGMs.

Several work has been done under fuzzy environment in the field of software testing and development for the last many decades (Utkin et al., 2002; Chatterjee et al., 2011; Chawla et al., 2012; Kapur et al., 2011; Zhang et al., 2014; Singh et al., 2011; Dimov and Punekkat, 2010; Kotaih and Khan, 2013). Some of them are: Jha et al. (2011) discussed fuzzy optimization techniques to formulate a fuzzy bi-criterion release time problem. Madsen et al. (2012) illustrated the usage of intuitionistic and fuzzy number in modeling optimization algorithms for software

Manuscript received July 5, 2019
Manuscript revised July 20, 2019

reliability computing. Pachuari et al. (2013) proposed a software reliability growth models and optimal release time under fuzzy set with imperfect debugging environment. Kiruthiga and Loganathan (2014) have proposed a SRGM under fuzzy environment. Seema et al. (2016) discussed the recent development and current issues in SRGMs under fuzzy set. Measurement of the software quality is thus the key factor that has to be taken into account while developing a software system. The study of Kalayathankal et al. (2017) intends to make available a fuzzy multiple and developed fuzzy software quality model. Dwivedi et al., (2018) discussed optimal release policy in fuzzy environment under imperfect debugging. Lohmor and Sagar (2019) presented a model based on FKNN (Fuzzy K-Nearest Neighbour) and nature inspired Glowworm Swarm Optimization (GSO) to understand the relationship between the data of software failure time and the nearest n failure time.

This paper proposes a software reliability growth model under fuzzy and imperfect debugging environment. Log-logistic testing effort function (LLTEF) is integrated into fuzzy software reliability growth model (SRGM). The parameters of LLTEF and SRGM are estimated by using least square and maximum likelihood methods. Numerical data analysis is presented. Optimal release policy is also discussed with illustrated example. This paper extends the works of Pachuari et al. (2013). It also viewed as an extension of Bokhari and Ahmad (2006) and Ahmad et al. (2011) to fuzzy environment. The rest of the article is organized as: In Section 2, basic preliminaries are discussed. A review of Log-logistic testing effort function is discussed in section 3, SRGM under fuzzy environment are discussed in Section 4, Estimation of parameters is given in Section 5, Data analysis and model comparison are also given in Section 6. Optimal release policy under fuzzy set is discussed in Section 7 and finally, conclusion is given in Section 8.

2. Basic Preliminaries of Fuzzy

2.1 Fuzzy set

Fuzzy set on U is a function A: U [0, 1], where A is the membership function or grade of x in A. We also write

$$A = \{(x, A(x)) : x \in U\}. \quad (1)$$

2.2 α-cuts

The α-cut of a set A, is known as a crisp set given by A^{α} . It is expressed as:

$$(A)^{\alpha} = \{x \epsilon A | \mu_{\tilde{A}}(x) \geq \alpha\}, \ where \ \alpha \in [0,1] \quad (2)$$

2.3 Fuzzy Number

The membership function of trapezoidal fuzzy number is given as (Klirr and Yuan, 1995):

$$\mu_A(x) = \begin{cases} 0, & x < x_1 \\ \frac{x-x_1}{x_2-x_1}, & x_1, \leq x \leq x_2 \\ 1, & x_2 \leq x \leq x_3 \\ \frac{x_4-x}{x_4-x_3}, & x_3 \leq x \leq x_4 \\ 0, & x > x_4 \end{cases} \quad (3)$$

2.4 Arithmetic Operations on Fuzzy Numbers

Let $A_{\alpha} = [a_l^{\alpha}, a_r^{\alpha}]$ and $B_{\alpha} = [b_l^{\alpha}, b_r^{\alpha}]$ be the two intervals of fuzzy and their operation is given in Table 1.

Table 1: Fuzzy number operation

Addition	$A + B = [a_l^{\alpha} + b_l^{\alpha}, a_r^{\alpha} + b_r^{\alpha}]$
Subtraction	$A - B = [a_l^{\alpha} - b_l^{\alpha}, a_r^{\alpha} - b_r^{\alpha}]$
Multiplication	$A * B = [a_l^{\alpha} * b_l^{\alpha}, a_r^{\alpha} * b_r^{\alpha}]$
Division	$A/B = [a_l^{\alpha}/b_l^{\alpha}, a_r^{\alpha}/b_r^{\alpha}]$

2.5 Fuzzification and Defuzzification Techniques

Fuzzification is the process of transforming a crisp variable into a fuzzy set where as defuzzification is the process of producing an experimental result in fuzzy logic. Trapezoidal fuzzifier is used for fuzzification and center of gravity (COG) is used for defuzzification in this paper and is given as:

$$COG_{out} = \frac{\int_{x_1}^{x_2} x \mu_{out}(x) dx}{\int_{x_1}^{x_2} \mu_{out}(x) dx} \quad (4)$$

3. Review of Log-logistic TEF

The mathematical form of cumulative LLTEF in time (0, t], (Bokhari and Ahmad, (2006); Ahmad et al. (2011)) is:

$$W(t) = \alpha \left[\frac{(\beta.t)^{\delta}}{1+(\beta.t)^{\delta}}\right], \ t > 0 \quad (5)$$

where α is the total amount of testing-effort consumption required by software testing, β is the scale parameter and δ is shape parameters.

Therefore, the current testing effort expenditure at testing time t are given by,

$$w(t) = \frac{\alpha.\beta.\delta(\beta t)^{\delta-1}}{[1+(\beta t)^{\delta}]^2} \quad t > 0, \alpha > 0, \beta > 0, \delta > 0 \quad (6)$$

4. SRGM under Fuzzy Environment

Recently Pachuari et al. (2013) proposed a software reliability growth models and optimal release time under fuzzy set with imperfect debugging environment. In this paper we first discuss the SRGM model by using

probabilistic method. An NHPP SRGM under imperfect debugging is discussed with the assumption that if deleted errors are removed, then there is a possibility that new errors with a constant rate γ are introduced (Ahmad et al., 2011).

The proposed software reliability growth model is based on following assumptions (Huang and Kuo, 2002; Yamada et al., 1986, 1993; Ahmad et al., 2010; 2011; Rafi et al., 2010).

(i) Error detection and elimination process in software testing is modeled by an NHPP.
(ii) Due to errors remaining in the software system, the software is subject to failure at random times.
(iii) Each time a failure occurs, the error that caused it is immediately removed and new errors may be introduced with probability γ.
(iv) The number of errors detected in the time interval (t, $t + \Delta$ t) to the current testing-effort is proportional to the mean number of remaining errors in the system and proportionality is constant over time.
(v) Log-logistic testing-effort function is used as testing-effort function.

Let a(t) be the number of errors to be detected and the number of new errors may be introduced in the system by time t. On the above assumption, the following differential equations is obtained:

$$\frac{dm(t)}{dt}.\frac{1}{w(t)} = r\big(a(t) - m(t)\big) \quad (7)$$

$$\frac{da(t)}{dt} = \gamma\frac{dm(t)}{dt} \quad (8)$$

where $m(t), w(t), a(t), \gamma$ and r are mean value function, testing-effort function, total number of faults, probability to introduce new faults and detection rate respectively.

Now solving the Eq. (7) and Eq. (8) with assumptions $m(0) = 0, W(0) = 0, a(0) = a$, we obtain the following mean value function

$$m(t) = \frac{a}{1-\gamma}\big(1 - e^{-r(1-\gamma)W(t)}\big) \quad (9)$$

where $w(t)$ is Log-logistic testing-effort function, a is expected number of initial error in the system, and r is the error detection rate per unit testing-effort at time t. Then the failure intensity at testing time t of NHPP is given by:

$$\lambda(t) = \frac{dm(t)}{dt} = \frac{a}{1-\gamma}.r.w(t).e^{-r(1-\gamma)W(t)} \quad (10)$$

The expected number of errors to be detected is:

$$m(\infty) = \frac{a}{1-\gamma} \quad (11)$$

And finally the conditional reliability function is:

$$R(t + \Delta t/t) = e^{-m(t+\Delta t)-m(t))} \quad (12)$$

5. Estimation of SRGM parameter

Using actual data sets, we can estimate the value of parameters such as, total amount of testing-effort expenditure (α) required by software testing, scale parameters (β), shape parameters (δ), total number of faults (a), detection rate (r), and probability to introduce new faults (γ) of SRGM with Log-logistic testing effort using the following least square estimation and maximum likelihood estimation methods:

Let n observed data pairs are in the form (t_k, W_k), where $k = 1, 2, ..., n; \quad 0 < t_1 < t_2 < .... < t_n$. Then the parameters of Log-logistic testing effort are estimated by minimizing (Bokhari and Ahmad, 2006; Ahmad et al., 2011):

$$S(\alpha, \beta, \delta) = \sum_{k=1}^{n}[W_k - W(t_k)]^2$$

Let n observed data pairs are in the form (t_k, y_k); where $k = 1, 2, ..., n; 0 < t_1 < t_2 < ..... < t_n$, and y_k is detected cumulative number of faults during $(0, t_k]$. Then the maximum likelihood estimates of the parameters a, r, and γ in the SRGM model is obtained by (Musa et al., 1987; Bokhari and Ahmad, 2006; Ahmad et al., 2011):

$$L(a, r, \gamma) = \prod_{k=1}^{n}\frac{\big[m(t_k) - m(t_{k-1})\big]^{(y_k - y_{k-1})}}{(y_k - y_{k-1})!}.e^{-[m(t_k)-m(t_{k-1})]}$$

where $t_0 \equiv 0$ and $y_0 \equiv 0$.

6. Data Analysis and Model Comparison

The proposed work has been validated on two data sets which are taken from (Ohba, 1984; Tohma et al., 1989). DS1 denotes the first data set is taken from Ohba, (1984) which consists of approximately 1,317,000 lines of code and found that after 19 weeks of testing, a total of 47.65 CPU hours were consumed to detect about 328 software errors (Ohba, 1984). The second data set (DS2) is taken from Tohma et al., (1989), reported that after 22 days of testing, a total of 86 software faults were detected and 93 CPU hours were consumed.

We estimate the SRGM parameters through SPSS. Then we calculate crisp reliability by using estimated value of parameters. We fuzzify the estimated value of parameters by using trapezoidal Fuzzy numbers. After applying arithmetic operation on fuzzy numbers, we get upper and lower limit of parameter. By using these values we calculate upper and lower limit of failure intensity and

fuzzy reliability. Finally defuzzify the reliability at different testing time and spreads with different presumption level (0 to 1). The estimated value of parameters is given in following Table 2 and Table 3 for different data set.

Table 2: Estimated values of parameters for DS1.

Parameters	α	β	δ	a	r	γ
Estimated value	1.45E3	0.003	1.12	563.26	0.02	0.03

Table 3: Estimated values of parameters for DS2.

Parameters	α	β	δ	a	r	γ
Estimated value	177.02	0.048	1.973	133.1	0.02	0.26

6.1 Fuzzify the estimated value of parameters, by using TrFN

Trapezoidal fuzzy numbers (TrFN) have been used to minimize the uncertainty involved in the estimated parameters. For example, assume that the crisp value of parameter 'a' be 563.263 and is called the initial number of errors. Also assume that there is 1% uncertainty in this value. Add and subtract 1% of value from original value. We obtain three numbers, and for getting fourth number we subtract 1% of a from the result of subtracted value of 1% of a from a, that is

$$a_1 = \{563.263 - \left(1 * \left(\frac{563.263}{100}\right)\right\} - \left(1 * \left(\frac{563.263}{100}\right)\right),$$
$$a_2 = 563.263 - \left(1 * \left(\frac{563.263}{100}\right)\right),$$
$$a_3 = 563.263,$$
$$a_4 = 563.263 + (1 * \frac{563.263}{100})$$

Now, a trapezoidal fuzzy number may be given as follows:

$$u_A(x) = \begin{cases} 0, & x < 551.99774 \\ \frac{x-551.99774}{557.63037-551.99774}, & 551.99774, \le x \le 557.63037 \\ 1, & 557.63037 \le x \le 563.263 \\ \frac{568.89563-x}{568.89563-563.263}, & 563.263 \le x \le 568.89563 \\ 0, & x > 568.89563 \end{cases}$$

Same as +/-1% we can fuzzyfy the parameters at +/-2% and +/-3% spreads only by calculating 2% and 3% at the place of 1%. Similarly, the other parameters are converted in the form of trapezoidal fuzzy numbers. Fuzzy representations of the parameters are shown in following table.

Table 4: Fuzzy representation of parameters using trapezoidal fuzzy number for DS1 at +/1% spread.

Parameters	Fuzzy representation
α (1452.068)	$(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ (1423.0266,1437.5474,1452.068 1466.5887)
β (0.003)	$(\beta_1, \beta_2, \beta_3, \beta_4)$ (0.00294, 0.00297, 0.003, 0.00303)
δ (1.11602)	$(\delta_1, \delta_2, \delta_3, \delta_4)$ (1.09369, 1.10486, 1.11602, 1.12718)
a (563.263)	(a_1, a_2, a_3, a_4) (551.9977, 557.63037, 563.263, 568.8956)
r (0.02)	(r_1, r_2, r_3, r_4) (0.0196, 0.0198, 0.02, 0.0202)
γ (0.029)	$(\gamma_1, \gamma_2, \gamma_3, \gamma_4)$ (0.02842, 0.02871, 0.029, 0.02929)

Table 5: Fuzzy representation of parameters using trapezoidal fuzzy number for DS2 at +/1% spread.

Parameters	Fuzzy representation
α (177.02)	$(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ (175.2498, 175.2452, 177.02 178.7902)
β (0.048)	$(\beta_1, \beta_2, \beta_3, \beta_4)$ (0.04704, 0.04752, 0.048, 0.04848)
δ (1.973)	$(\delta_1, \delta_2, \delta_3, \delta_4)$ (1.93354, 1.95327, 1.973, 1.99273)
a (133.1)	(a_1, a_2, a_3, a_4) (130.438, 131.769, 133.1, 134.431)
r (0.016)	(r_1, r_2, r_3, r_4) (0.01568, 0.01584, 0.016, 0.01616)
γ (0.265)	$(\gamma_1, \gamma_2, \gamma_3, \gamma_4)$ (0.2597, 0.2623, 0.265, 0.2676)

In Table 4 and Table 5, we calculate 1% from the value of parameters then add and subtract this value to crisp value of parameters. We get three values for one parameter. We calculate fourth one by subtract 1% of parameter from its previous value. For example, let a = 563.263 and its 1% is 5.63263, now we add this value to 563.263 and get $a_4 = 563.263 + 5.63263 = 568.8956$.We subtract this value from 563.263 and get as $a_2 = 563.263 - 5.63263 = 557.63037$. Now we have three values, such as 557.63037, 563.263, 568.8956. For calculating fourth value we subtract 1% of 563.263 from leftmost value i.e. 557.63037 and we get $a_1 = 551.9977$.

Similarly, we fuzzify all parameters at +/-2% and +/-3% spreads. In these cases we find out 2% and 3% of crisp value of parameters, rest work is same as 1% spreads.

6.2 Arithmetic Operations

Let $p = [l^\alpha, r^\alpha]$
$= [l + (m - l)\alpha, r - (r - n)\alpha], \quad \forall \alpha \in [0, 1]$ (14)

We use the above formula to calculate Lower and Upper limit of all fuzzified parameters at different presumption level (0 to 1). After putting the upper and lower limit of estimated parameter a and r and current testing effort $(w(t))$, cumulative testing effort $(W(t))$ corresponding to different value of t in Eq. (10) and Eq. (13), we calculate upper and lower limit of failure intensity for DS1 and DS2 at different presumption levels with +/-1%, +/-2% and +/-3% spreads for trapezoidal fuzzy numbers, coupled with α-cuts. Lower and upper limits of failure intensity for DS1 are shown in Table 6 at t = 10.

Table 6: Failure intensity at different presumption level at t = 10 for DS1

Presumption Level	At +/-1% spread		At +/-2% spread		At +/-3% spread	
	Lower limit	Upper limit	Lower limit	Upper limit	Lower limit	Upper limit
0	20.05	21.15	19.31	21.51	18.58	21.88
0.1	19.19	20.16	19.39	21.44	18.69	21.77
0.2	19.23	20.12	19.46	21.37	18.80	21.66
0.3	19.26	20.09	19.53	21.29	18.91	21.55
0.4	19.30	20.05	19.61	21.22	19.02	21.44
0.5	19.33	20.02	19.68	21.15	19.13	21.33
0.6	19.36	19.98	19.75	21.07	19.24	21.22
0.7	19.40	19.95	19.83	21.00	19.35	21.11
0.8	19.43	19.92	19.90	20.93	19.46	21.00
0.9	19.47	19.88	19.97	20.85	19.57	20.89
1	19.50	19.85	20.05	20.78	19.68	20.78

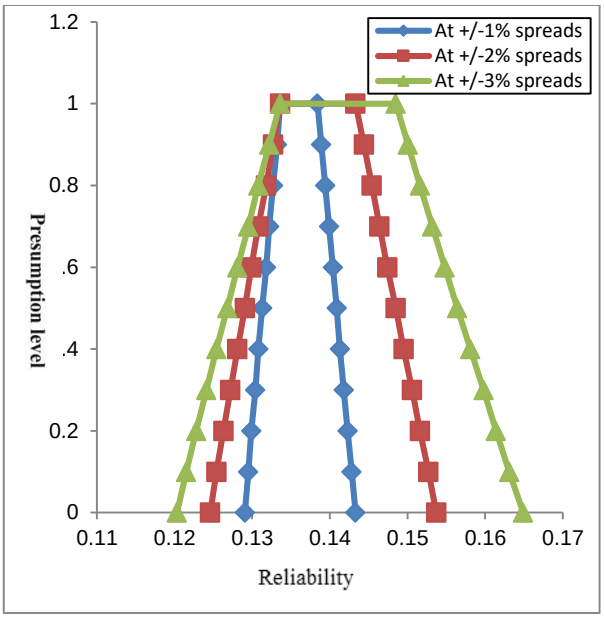
From the Table 6, we can say that failure intensity is high at the top of upper limit column and increases in next row. We get lowest failure intensity at the bottom of upper limit column and failure intensity at the top of the lower limit column and decreases in next row. We get higher failure intensity at the bottom of the lower limit column. It means that at the top of lower limit column, we get minimum failure intensity and maximum failure intensity at the top of upper limit. Also we get a middle value of failure intensity at the bottom of lower limit column and the other middle value of failure intensity at the bottom of upper limit column. This is due to proposed SRGM considering trapezoidal fuzzy number whereas for triangular fuzzy number, highest value is at the top of upper limit and lowest value is at the top of lower limit but there are same values at the bottom of upper and lower limit.

Similarly, we calculate the upper and lower limit of failure intensity at +/-1%, +/-2% and +/-3% spreads for DS2. Lower and upper limits of failure intensity are shown in Table 7 at t = 10.

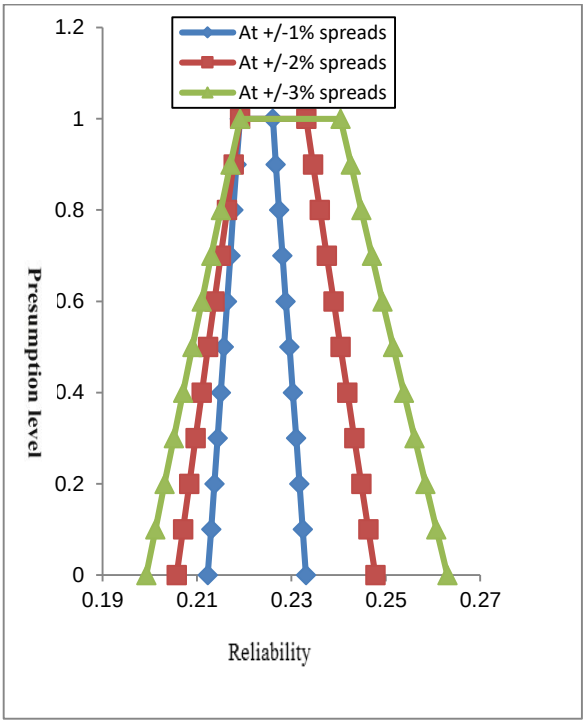
Table 7: Failure intensity at different presumption level at t = 10 for DS2

Presumption Level	At +/-1% spread		At +/-2% spread		At +/-3% spread	
	Lower limit	Upper limit	Lower limit	Upper limit	Lower limit	Upper limit
0	9.71	10.91	8.95	11.32	8.37	11.76
0.1	9.75	10.87	9.02	11.24	8.48	11.63
0.2	9.78	10.83	9.10	11.15	8.58	11.50
0.3	9.82	10.78	9.17	11.07	8.69	11.37
0.4	9.86	10.74	9.25	10.98	8.80	11.24
0.5	9.90	10.70	9.32	10.90	8.91	11.11
0.6	9.93	10.66	9.40	10.82	9.02	10.99
0.7	9.97	10.62	9.48	10.74	9.13	10.86
0.8	10.01	10.57	9.55	10.65	9.24	10.74
0.9	10.05	10.53	9.63	10.57	9.35	10.61
1	10.09	10.49	9.71	10.49	9.46	10.49

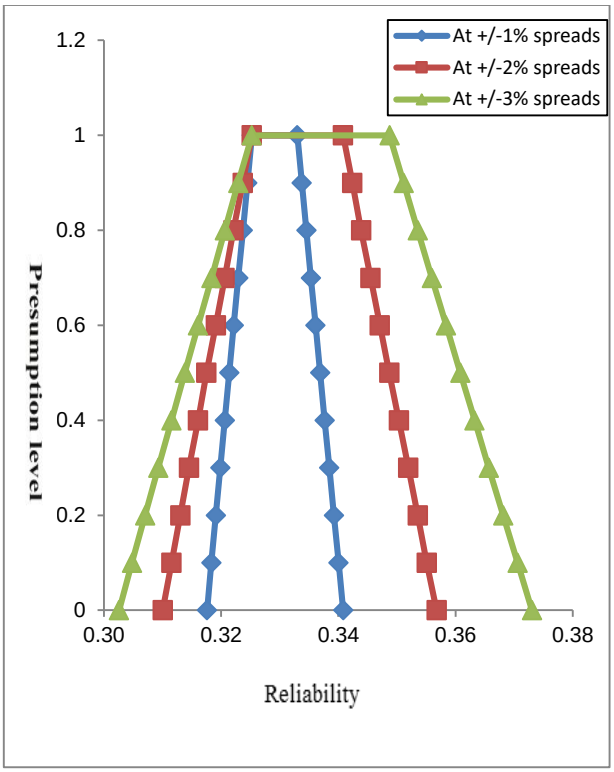
The membership functions of reliability at testing time 10, 15, 20 and 25 weeks for DS1 in imperfect debugging, are shown in following Figs. 4a-4d .



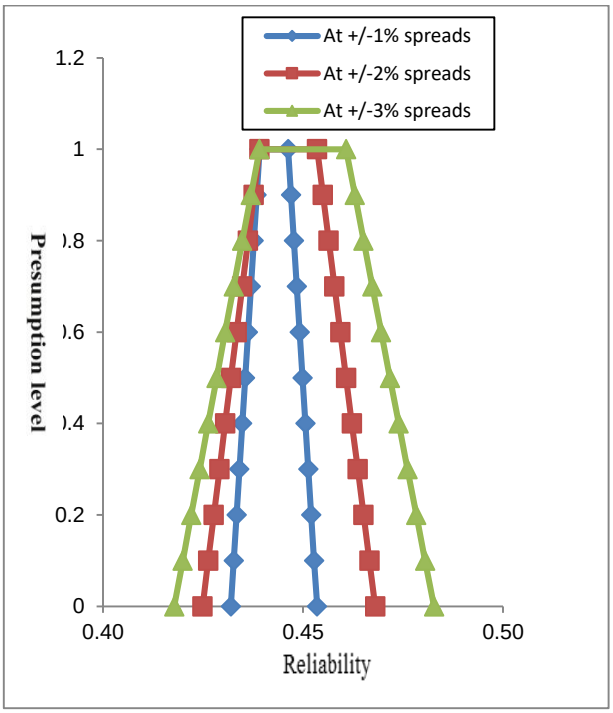
4a. After 10 weeks



4b. After 15 weeks



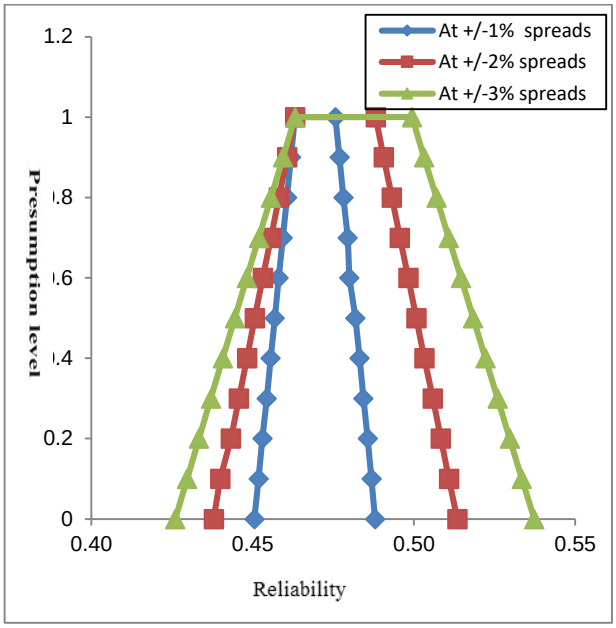
4c. After 20 weeks



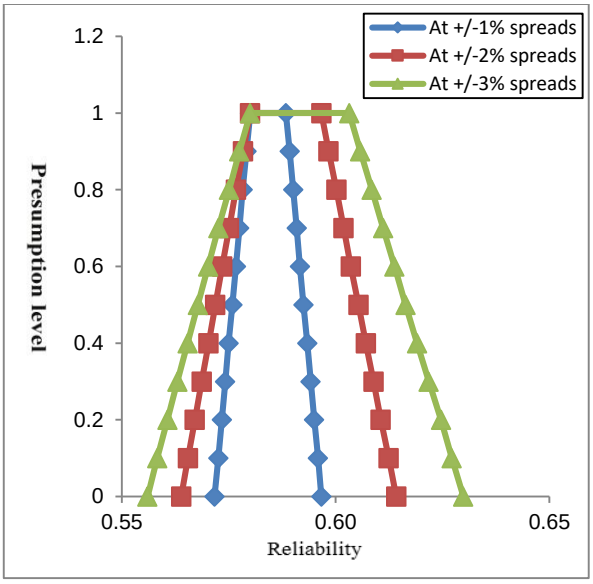
4d. After 25 week

Fig. 4a-4d. Spreads in reliability at different time for DS1

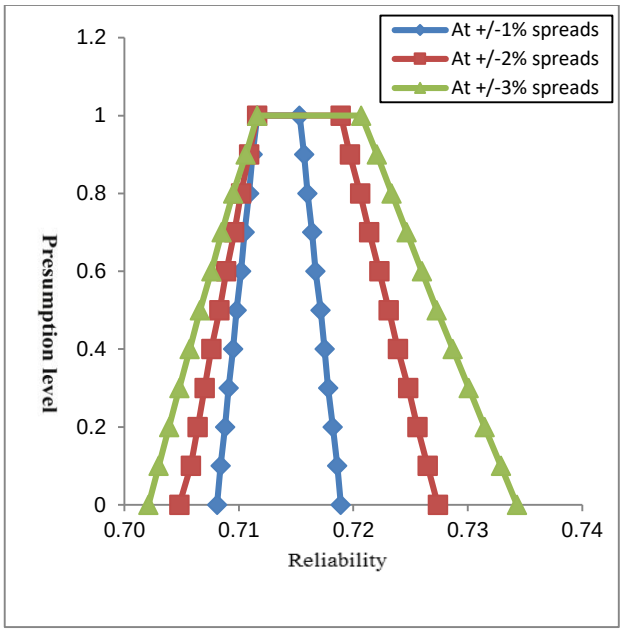
The membership functions of reliability at testing time 6, 12, 18 and 22 weeks for DS2 in imperfect debugging, are shown in following Figs. 5a-5d.



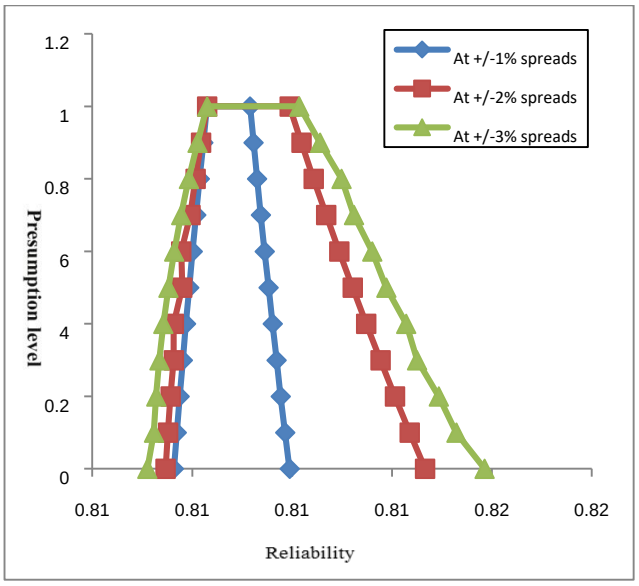
5a. After 10 days



5b. After 15 days



5c. After 20 days



5d. After 25 days

Fig. 5a-5d. Spreads in reliability at different time for DS2

6.3 Compare the result of our proposed model with previous models

The defuzzified values of reliability at +/-1%, +/-2% and +/-3% spreads for different testing time have been calculated along with their crisp values for DS1 and DS2. It is shown in following Table 8 and Table 9.

Table 8: Comparison of Defuzzified values after different testing time for DS1

Testing time (weeks) t	Reliability					
	Crisp Reliability	Defuzzified value			Rafi et al. (2010)	Pachauri et al. (2013)
		spread +/-1%	**spread +/-2%**	**spread +/-3%**		
10	0.1337	0.1362	0.1389	0.1419	0.1502	0.1488
15	0.2192	0.2227	0.2265	0.2306	0.2324	0.2268
20	0.3252	0.3292	0.3333	0.3375	0.5131	0.5209
25	0.4391	0.4427	0.4464	0.4502	0.9462	0.9640

From the above table we can say that reliability is reasonable as compare to the previous model at t = 10, 15, 20 and 25.

Table 9: Comparison of Defuzzified values after different testing time for DS2

Testing time (weeks) t	Reliability					
	crisp Reliability	Defuzzified value			Rafi et al. (2010)	Pachauri et al. (2013)
		spread +/-1%	spread +/-2%	spread +/-3%		
10	0.4632	0.4695	0.4759	0.4816	0.656	0.565
15	0.5801	0.5843	0.5888	0.5924	0.722	0.748
20	0.7116	0.7135	0.7159	0.7557	0.777	0.912
25	0.8103	0.8108	0.8117	0.8119	0.823	0.984

From the above table we can say that the reliability increase when spread value is increases. We observe that value of fuzzy reliability increases when we fuzzify the parameters by using trapezoidal fuzzy number, also observe that reliability increases faster as compare with previous used triangular fuzzy number. It means fuzzification of parameters by using trapezoidal fuzzy number gives more efficient reliability than fuzzify the parameters by using triangular fuzzy number.

7. Optimal Release Policy under Fuzzy

Optimal release policy is one of most important application of reliability models. In optimal release policy we study the appropriate time to release software into market, because it is not important for customer. So release software on appropriate time is a very important task for developer. In this section, we discuss the cost model and release policy based on the cost-reliability criterion using fuzzy numbers. Similar process has been used to calculate the optimal release time. Cost function can be redefined as (Bokhari and Ahmad, 2006; Pachuari et al., 2013; Dwivedi, 2018):

$$C(t) = C^1 m(t) + C^2[m(t_{LC}) - m(t)] + C^3 \int_0^t w(x)dx \quad (15)$$

where $C^1 = (C_1^1, C_2^1, C_3^1, C_4^1)$ the cost of correcting an error during is testing, $C^2 = (C_1^2, C_2^2, C_3^2, C_4^2)$ is the cost of correcting an error during operation, $C^2 > C^1, C^3 =$

$(C_1^3, C_2^3, C_3^3, C_4^3)$ is the cost of per unit testing-effort expenditures and t_{LC} is the software life-cycle length. The conditional reliability of software system for a given period of time is:

$$R(t + \Delta t/t) = e^{-(m(t+\Delta t)-m(t)} \quad (16)$$

where $m(t)$ is the mean value function. First we calculate the reliability time to reach the desired reliability as described in Eq. (16). In the next step, total software cost has been calculated using fuzzy number for DS2.

By differentiating Eq. (15) with respect to T and equating to zero, gives:

$$\frac{dC(T)}{dT} = C^1 \frac{dm(T)}{dT} - C^2 \frac{dm(T)}{dT} + C^3 w(T) = 0,$$

$$\text{or, } (C^1 - C^2)\frac{dm(T)}{dT} = -C^3 w(T)$$

$$\frac{\frac{dm(T)}{dT}}{w(T)} = \frac{C^3}{C^2 - C^1}$$

$$\frac{\lambda(T)}{w(T)} = \frac{C_3}{C_2 - C_1}$$

where

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

$$= \frac{a.r.w(T).e^{-r.W(t)}}{w(T)} = \frac{C_3}{C_2 - C_1}$$

$$= a.r.e^{-r.W(t)} = \frac{C_3}{C_2 - C_1}$$

$$= r.\big(a - m(T)\big) = \frac{C_3}{C_2 - C_1} \quad (17)$$

where $m(T) = a.(1 - e^{r.W(t)})$

when $T = 0$, then $M(0) = 0$ and $\frac{\lambda(T)}{w(T)} = a.r$

When $T = \infty$the $m(\infty) = a.(1 - e^{-r.\alpha})$

and $\frac{\lambda(t)}{w(t)} = a.r.e^{-r.\alpha}$.

$\frac{\lambda(T)}{w(T)}$ is monotonically decreasing in T.

Illustration: We calculate cost for DS2. According to Bokhari and Ahmad (2006), let us assume that $C^1 = 1, C^2 = 60, C^3 = 30$, to get more than 85% reliable software by cost-reliability criterion, we need minimum 24.42781 weeks for testing. To calculate software cost at $t = 24.42781$, first of all we defuzzify C^1, C^2 and C^3at +/-1%, +/-2% and +/-3% spreads. The defuzzified value of C^1, C^2, and C^3 are shown in Table 10. Now we calculate lower and upper limit for C^1, C^2, and C^3. Lower and upper limit of C^1, C^2,and $C^3$are shown in Tables 11-13, respectively. We calculate $m(T)$ and $W(T)$ at $t =$ 24.42781. By putting the lower and upper limit of C^1, C^2, C^3, $m(T)$ and $W(T)$ in Eq. (15). Total cost at different presumption levels at $t = 24.42781$ is given Table 14 and the defuzzified values are 4383.4211, 4884.720611 and 4953.636127 units for +/-1%, +/-2% and +/-3% spreads, respectively, which is more than the actual cost.

Table 10: Fuzzy representation of different cost at +/-1%, +/-2% and +/-3% spreads.

Cost	Fuzzy representation (+/-1%)	Fuzzy representation (+/-2%)	Fuzzy representation (+/-3%)
C^1 (1)	$(C_1^1, C_2^1, C_3^1, C_4^1)$ (0.98, 0.99, 1, 1.01)	$(C_1^1, C_2^1, C_3^1, C_4^1)$ (0.96, 0.98, 1, .02)	$(C_1^1, C_2^1, C_3^1, C_4^1)$ (0.94, 0.97, 1, 1.03)
C^2 (60)	$(C_1^2, C_2^2, C_3^2, C_4^2)$ (58.8, 59.4, 60, 60.6)	$(C_1^2, C_2^2, C_3^2, C_4^2)$ (57.6, 58.8, 60, 1.2)	$(C_1^2, C_2^2, C_3^2, C_4^2)$ (56.4, 58.2, 60,61.8)
C^3 (30)	$(C_1^3, C_2^3, C_3^3, C_4^3)$ (29.4, 29.7, 30, 30.3)	$(C_1^3, C_2^3, C_3^3, C_4^3)$ (28.8, 29.4, 30, 0.6)	$(C_1^3, C_2^3, C_3^3, C_4^3)$ (28.2, 29.1, 30,30.9)

Table 11: Lower and upper limit of C^1at +/-1%, +/-2% and +/-3% spreads.

Presumption Level	At +/-1% spread		At +/-2% spread		At +/-3% spread	
	Lower limit	Upper limit	Lower limit	Upper limit	Lower limit	Upper limit
0	0.98	1.01	0.96	1.02	0.94	1.03
0.1	0.98	1.01	0.96	1.02	0.94	1.03
0.2	0.98	1.01	0.96	1.02	0.95	1.02
0.3	0.98	1.01	0.97	1.01	0.95	1.02
0.4	0.98	1.01	0.97	1.01	0.95	1.02
0.5	0.99	1.01	0.97	1.01	0.96	1.02
0.6	0.99	1.00	0.97	1.01	0.96	1.01
0.7	0.99	1.00	0.97	1.01	0.96	1.01
0.8	0.99	1.00	0.98	1.00	0.96	1.01
0.9	0.99	1.00	0.98	1.00	0.97	1.00
1	0.99	1.00	0.98	1.00	0.97	1.00

Table 12: Lower and upper limit of C^2at +/-1%, +/-2% and +/-3% spreads.

Presumption Level	At +/-1% spread		At +/-2% spread		At +/-3% spread	
	Lower limit	Upper limit	Lower limit	Upper limit	Lower limit	Upper limit
0	58.80	60.60	57.60	61.20	56.40	61.80
0.1	58.86	60.54	57.72	61.08	56.58	61.62
0.2	58.92	60.48	57.84	60.96	56.76	61.44
0.3	58.98	60.42	57.96	60.84	56.94	61.26
0.4	59.05	60.36	58.08	60.72	57.12	61.08
0.5	59.10	60.30	58.20	60.60	57.30	60.90
0.6	59.16	60.24	58.32	60.48	57.48	60.72
0.7	59.22	60.18	58.44	60.36	57.66	60.54
0.8	59.28	60.12	58.56	60.24	57.84	60.36
0.9	59.34	60.06	58.68	60.12	58.02	60.18
1	59.40	60.00	58.80	60.00	58.20	60.00

Table 13: Lower and upper limit of C^3at +/-1%, +/-2% and +/-3% spreads.

Presumption Level	At +/-1% spread		At +/-2% spread		At +/-3% spread	
	Lower limit	Upper limit	Lower limit	Upper limit	Lower limit	Upper limit
0	29.40	30.30	28.80	30.60	28.20	30.90
0.1	29.43	30.27	28.86	30.54	28.29	30.81
0.2	29.46	30.24	28.92	30.48	28.38	30.72
0.3	29.49	30.21	28.98	30.42	28.47	30.63
0.4	29.52	30.18	29.04	30.36	28.56	30.54
0.5	29.55	30.15	29.10	30.30	28.65	30.45
0.6	29.58	30.12	29.16	30.24	28.74	30.36
0.7	29.61	30.09	29.22	30.18	28.83	30.27
0.8	29.64	30.06	29.28	30.12	28.92	30.18
0.9	29.67	30.03	29.34	30.06	29.01	30.09
1	29.70	30.00	29.40	30.00	29.10	30.00

Table 14: Lower and upper limit of cost at +/-1%, +/-2% and +/-3% spreads.

Presumption Level	At +/-1% spread		At +/-2% spread		At +/-3% spread	
	Lower limit	Upper limit	Lower limit	Upper limit	Lower limit	Upper limit
0	4795.9	5067.3	4612.8	5155.3	4458.9	5247.5
0.1	3047.2	5058.1	4631.1	5137.2	4486.4	5220.2
0.2	3056.4	5048.9	4649.4	5119.2	4513.8	5192.9
0.3	3065.6	5039.7	4667.7	5101.3	4541.3	5165.7
0.4	3074.8	5030.5	4686.0	5083.2	4568.7	5138.4
0.5	3084.1	5021.3	4704.3	5065.3	4596.1	5111.2
0.6	3093.3	5012.2	4722.6	5047.3	4623.5	5084.1
0.7	3102.6	5002.9	4740.9	5029.3	4650.9	5056.9
0.8	3111.9	4993.8	4759.3	5011.4	4678.2	5029.7
0.9	3121.2	4984.6	4777.6	4993.4	4705.6	5002.6
1	3130.6	4975.4	4795.9	4975.4	4732.9	4975.4

The main threat to validity of the proposed approach is it's sensitivity to the choice of spreads. A wrong choice of spread may lead to wrong conclusion. Hence, it is essential for the system programmer to choose a most suitable spread while fuzzifying the crisp values.

8. Conclusion

In this paper, a software reliability growth model and optimal release policy has been developed under fuzzy and imperfect debugging environment. We have incorporated log-logistic testing effort function (LLTEF) into software reliability growth model (SRGM). We have estimated the parameters of LLTEF and SRGM by using methods least square estimation and maximum likelihood estimation. We have also considered the uncertainty in SRGM estimated parameters. Fuzzy software reliability and software cost has been obtained for proposed SRGM model. The results have been compared with the other previous SRGM model. It is concluded that the variation in factors such as reliability can be modeled efficiently in the fuzzy paradigm. The proposed fuzzy SRGM is helpful to make reasonable prediction of software reliability and software cost measures. Further this work can be extended in many directions such as incorporating other testing effort function, change point concept, and dynamic error detection rate.

Acknowledgment

The authors would like to thank the editor and three referees for their valuable comments and suggestions.

References

[1] L. A. Zadeh , "Fuzzy sets", Information and Computation, Vol. 8,pp. 338-353, 1965.

[2] Z. Gong, S. Hai, "The Interval-Valued Trapezoidal Approximation of Inter-Value Fuzzy Numbers and Its Application in Fuzzy Risk Analysis", Journal of Applied Mathematics, Vol. 2014, pp. 22, 2014.

[3] P. K. Kapur, A. 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.

[4] J. D. Musa, "A theory of software reliability and its application", IEEE Trans. Software Eng., Vol. SE-1, pp. 312-327, 1971.

[5] L. V. Utkin, S. V. Gurov, M. I. Shibinsky, "A fuzzy software reliability model with multiple-error introduction and removal", International Journal of Reliability, Quality, and Safety Engineering, Vol. 9, No. 3, pp. 215-227, 2002.

[6] N. Raj Kiran, V. Ravi, "Software reliability prediction by soft computing techniques", J. Syst. Software, doi:10.1016/j.jss.2007.05.005, 2007.

[7] N. Ahmad, M. U. Bokhari, S. M. K. Quadri, M. G. M. Khan, "The exponentiated Weibull software reliability growth model with various testing-efforts and optimal release policy", International Journal of Quality & reliability Management, Vol. 25, No 2, 2008.

[8] N. Ahmad, M. G. M. Khan, S. M. K. Quadri, 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.

[9] M. U. Bhokhari, 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.

[10] S. M. K. Quadri, N. Ahmad, Sheikh Umar Farooq, "Software reliability growth modeling with generalized exponential testing effort and optimal release policy", Global Journal of Computer Science and Technology, Vol. 11, 2011.

[11] 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.

[12] N. Ahmad, M. G. M. Khan, 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.

[13] N. Ahmad, M. G. M. Khan, 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.
[14] M. Z. Imam, I. J. Ara, N. Ahmad, "Analysis of Software Fault Detection and Correction Processes with Log-logistic Testing-Effort", Recent Advances in Mathematics, Statistics and Computer Science, ISBN 978-981-4696-16-6, pp. 549-560, 2016.
[15] M. G. M. Khan, N. Ahmad, L. S. Rafi, "Determining the Optimal Allocation of Testing Resource for Modular Software System using Dynamic Programming", Communications in Statistics – Theory and Methods, Vol. 45(3), pp. 670-694, 2016.
[16] A. L. Goel, K. Okumoto, "Time dependent error detection measure", IEEE Transaction Reliability, R – 288(3), pp. 206-211, 1979.
[17] 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.
[18] J. D. Musa, A. Lannino, K. Okumoto, Software Reliability Measurement, Prediction and Application, McGraw – Hill, 1987.
[19] 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.
[20] 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.
[21] B. Pachauri, A. Kumar, J. Dhar, "Modeling Optimal release policy under fuzzy paradigm in imperfect debugging environment", Information and Software Technology, Vol. 55, Issue 11, pp. 1974-1980, 2013.
[22] M. Kiruthiga, C. Loganathan, "Software Reliability Modeling in Fuzzy Environment", Progress in Nonlinear Dynamics and Chaos, Vol. 2, No. 1, 2014.
[23] Seema, R., I. J. Ara, and N. Ahmad, "Recent Review and Current Issues in Software Reliability Growth Models under Fuzzy Environment", International Journal of Latest Trends in Engineering and Technology, Vol. 6 (4), pp. 550-558, 2016.
[24] L. Zhang, P. Guo, S. Fong, M. Li, "Monthly optimal reservoirs operation for multicrop deficit irrigation under fuzzy stochastic uncertainties", Journal of Applied Mathematics, Vol. 14, Page 11, 2014.
[25] 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.
[26] S. Chatterjee, J. 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, 2015.
[27] H. Madsen, G. Albeanu, F. P. Vladicescu, "An Intuitionistic Fuzzy Methodology for Component-Based Software Reliability Optimization", International Journal of Performability Engineering, Vol. 8, No. 1, pp. 67-76, 2012.
[28] P. K. Kapur, H. Pham, A. 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.
[29] G. Chawla, S. Kumar, D. Goyal, P. Gupta, "An Improved Fuzzy Model to Predict Software Reliability", International Journal of Computer Science & Management Studies, Vol. 12, Issue 03, pp. 44-48, 2012.
[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] B. Kotaih, R. A. Khan, "Software Reliability Assessment by using Neural Networks with Fuzzy Logic based Systems", in: Proceeding of International Conference on Advances in Computer Science, AETACS, 2013.
[33] M. Ohba, "Software reliability analysis models", IBM Journal of Research Development, Vol. 28, No 4, pp. 428-443, 1984.
[34] S. M. Rafi, K. N. Rao, S. Akhtar, "Incorporating generalized modified Weibull TEF in to software reliability growth model and analysis of optimal release policy", Computer and Information Science, Vol. 3, No. 2, pp 145-162, 2010.
[35] Y. Singh, P. K. Bhatia, O. Sangwan, "Software reusability assessment using soft computing techniques", ACM SIGSOF Software Engineering Notes, Vol. 36, No. 1, pp. 1-7, 2011.
[36] Shailee Lohmor, B. Sagar, "Enhancing Software Reliability prediction based on Hybrid Fuzzy K-Nearest Neighbor with Glowworm Swarm optimization (FKNN-GSO) algorithm", International Journal of Recent Technology and Engineering, Vol. 7, Issue 6, 2019.
[37] Sunny Joselph Kalayathankal, John T. Abraham, Joselph Varghese Kureethara, "An ordered ideal intuitioniatic fuzzy software quality model", International Journal of Mechanical Engineering and Technology, Vol. 8, Issue 10, pp. 535-546, 2017.
[38] P. C. Jha, Indumati, Ompal Singh, Deepali Gupta, "Bi-criterion release time problem for a discrete SRGM under fuzzy environment", Vol. 3, Issue 6, 2011.
[39] Asit Dwivedi, "Optimal Release time of software under fuzzy environment with testing effort", International Journal of Business Analytics and Intelligence, Vol. 6, Issue 1, 2018.
[40] Y. Tohma, R. Jacoby, Y. Murata, and M. Yamamoto, "Hyper-Geometric Distribution Model to Estimate the Number of Residual Software Fault," Proceeding of COMPSAC-89, IEEE CS Press, Orlando, pp. 610-617, 1989.

Seema Rani is a research scholar in the University Department of Statistics and Computer Applications at Tilka Manjhi Bhagalpur University, Bhagalpur, India. She received the BCA and MCA degrees, from T.M. Bhagalpur University in 2006 and 2009, respectively. Her research interest includes statistical modeling, software testing, software reliability

engineering, and their application. She has published several papers in journals, and conferences in these areas.

Nesar Ahmad is a Professor in the University Department of Statistics and Computer Applications at Tilka Manjhi Bhagalpur University, Bhagalpur, India. He received the B. Sc. degree in Mathematics from Bihar University, Muzaffarpur, in 1984 and the M. Sc., M. Phil., and Ph. D. degree in Statistics from Aligarh Muslim University, Aligarh, in 1987, 1990, and 1993, respectively. He joined the University Department of Statistics and Computer Applications at Tilka Manjhi Bhagalpur University, Bhagalpur, in 1996. From 1995 to 1996 he was a post doctoral fellow of UGC/CSIR at Aligarh Muslim University, Aligarh. He has been a Lecturer at the University of the South Pacific, Suva, Fiji Islands from January 2006 to December 2009. He has about 25 years of experience in teaching and research. His research interests include life testing, statistical modeling, reliability analysis, software testing, software reliability engineering and optimization Techniques. He has published more than 80 papers in journals, and conferences in these areas.

Ahmad is an executive member of Indian Society of Information Theory and Its Applications (ISITA) and life member of Indian Science Congress. He is in the editorial board of International Journal of Scientific and Statistical Computing (IJSSC) and Journal of Convergence Information Technology (JCIT).