Journal of Information Technology Management

Print ISSN: 2008-5893

Online ISSN: 2423-5059

Multi-Objective Fuzzy Software Release Problem with Learning Capacities for Fault Detection and Correction Processes

Deepak Kumar*

*Corresponding Author, Professor, Amity Institute of Information Technology, Amity University Uttar Pradesh, Noida, India. E-mail: deepakgupta_du@redifmail.com

Pankaj Gupta

Assistant Professor, Department of Computer Science, Birla Institute of Technology, Mesra, India. E-mail: pgupta@bitmesra.ac.in

P. K. Kapur

Professor & Director, Amity Center for Interdisciplinary Research, Amity University Uttar Pradesh, Noida, India. E-mail: pkkapur1@gmail.com

Abstract

Without the utilization of computers and related technology, modern day's life cannot be headway. It has also transformed into an incredibly troublesome task. The genuine challenges included are shorter life cycles, cost-effectiveness, and higher software quality goals. Despite these challenges, the software developers have started to give cautious thought to the procedure to develop software, testing and reliability investigation of software, and to reinforce the method. Developers' most fundamental decisions related to the perfect release time of Software. The software development method incorporates a piece of vulnerabilities and ambiguities. We have proposed a multi-objective software release time issue under a fuzzy environment using a software reliability growth model to overcome such vulnerabilities and ambiguities. Further, we have discussed the fuzzy environment framework to deal with the issue. Considering the model and issue, we can especially address the issue of when to release software under these conditions. Results are illustrated numerically.

Keywords: Software Reliability, Software Reliability Growth Models, Fuzzy, Release Time Problem, Software Development Life Cycle

Journal of Information Technology Management, 2023, Vol. 15, Issue 3, pp. 17-30
Published by the University of Tehran, Faculty of Management
doi: https://doi.org/ 10.22059/jitm.2023.93621
Article Type: Research Paper

Received: April 03, 2023
Received in revised form: June 13, 2023
Accepted: July 20, 2023
Published online: August 26, 2023

Introduction

In recent years, numerous Software Reliability Growth Models (SRGMs) have been proposed for the estimation of the reliability of software products. As personal computers and their related software turn out to be more refined, whilst turning out to be more essential in our lives, developers are progressively worried about their reliability and quality. Software reliability offers the most far-reaching cutting-edge procedures to measure the software quality and testing approaches. Software improvement forms normally concentrate on errors, recognizing and redressing software development procedures that do happen at any period of software development.

In the previous four decades' numerous software reliability-based SRGM models (Goel & Okumoto, 1979; Kapur & Garg, 1980; Kumar et al., 2012) have been proposed in writing to gauge and anticipate the software reliability and number of faults staying in the software. Non-Homogeneous Poisson Process (NHPP) model is one of the SRGMs, created by Goel & Okumoto (1979).

A standout amongst the most critical utilization of SRGMs is to focus on the software release time. Most software engineers and managers constantly need to know the date on which the software reliability objective will be met.

For the most part, the detection of faults included in enhancing the software reliability of software comprises an exceptionally prolonged and costly testing procedure. It is accounted for that much of the time, more than a large portion of the time and cost is spent in testing when creating software. Subsequently, one of the imperative issues in this testing is the point at which we ought to quit testing and be prepared to release the software.

A wide range of methodologies have been proposed to focus on the ideal release time of software, because of distinctive SRGM. It couldn't be any more obvious, for case, (Goel & Okumoto, 1979; Yamada & Osaki, 1987; Jha et al., 2006).

So one of the significant concerns in the software development procedure is choosing when to quit testing and release the software. This release issue is known as the ideal software release and has been mulled over broadly. Goel & Okumoto (1979) tended to a cost for ideal software release arrangement that minimizes the aggregate expected software cost. Yamada & Osaki (1987) considered the ideal software release issue utilizing two cases: when the planned software release time is consistent and when it is an irregular variable with a self-assertive appropriation. Yamada & Osaki (1987) presented a cost-reliability software release strategy that minimizes the aggregate expected cost and fuzzy software reliability prerequisite.

Software release time issue has additionally been planned and comprehended by numerous scientists in the software field in distinctive ways (Huang & Lyu, 2005; Jha et al,

2006; Kapur et al., 1994; Kapur & Garg, 1990; Kapur et al., 2006). The greater part of the software release time issues talked about in literature consider the minimization of the cost of testing or maximize the software reliability subject to budgetary limitation and/or reliability level to be accomplished by the release time (Kapur et al., 1994; Kapur & Garg, 1990). Goel & Okumoto (1979) were the first to talk about the software release time approach from the cost-saving advantage perspective. A couple of researchers have additionally talked about the Bi-criteria release approach (Jha et al., 2006; Kapur & Garg, 1990) at the same time augmenting reliability and minimizing cost subject to reliability necessity and testing budget accessibility limitations. Optimization strategies, for example, methods of calculus, Mathematical Programming, and so forth are embraced to take care of these issues.

If testing stops too early, then there will be an excess of deficiencies left in the software, which will bring about unreasonable software failure during operation, and lead to critical misfortunes resulting into failure punishment or client dissatisfaction. Then again, spending an excess of time duration in testing may bring about a high testing cost, and postponement the release of the software into the commercial market.

In the current examination, we identified the SRGM for fault detection and correction with the learning capacities of developer. Software release time choice it is expected that every one of the parameters of the issue are known accurately. Parameters of the SRGM used to portray the failure occurrences are assessed from the detected faults. Different goals and limitations are fixed by the management and cost parameters included in the total cost are resolved in light of experience and all are altered constants. Practically speaking it is conceivable that the management is not ready to situate exact estimations of the different cost parameters and targets to be met by the release time. It might likewise be conceivable that the management itself doesn't set exact qualities keeping in mind the end goal to give some flexibility on these parameters because of focused inspections. This prompts vulnerability (fuzziness) in the problem. A fuzzy optimization approach to deal with these issues gives a solution to evaluate these vulnerabilities. The fuzzy set theory and fuzzy optimization procedures can be utilized as a part of such a circumstance. In this paper, we have solved an optimization problem for the optimal release time for software by minimizing cost capacity and maximizing reliability subject to budgetary constraint under a fuzzy domain (fuzzy objective, fuzzy inequalities in the limitations, and problem parameters being fuzzy numbers) and tackled by utilizing fuzzy optimization method.

The rest of the paper is composed as follows: First in section 2.1 we have examined the SRGM used to depict the useful relationship between the failure occurrence and time. In section 2.2 we have talked about the expense show and defined the issue in section 2.3. In area 3.1 we have talked about the essential ideas of fuzzy sets and given a calculation to explain the fuzzy optimization problem. Further in section 3.2 arrangement method is shown with a numerical case. At last, we conclude the paper in section 4.

Problem Formulation

Software Reliability Growth Models

For planning the release time of software, first, we characterize a cost capacity depicting the aggregate expected testing and investigating expense during testing and debugging stages. The cost acquired during the testing stage incorporates cost per unit testing time. Notwithstanding these expenses, the cost capacity can likewise incorporate punishment or opportunity misfortune cost because of postponed delivery, risk expenses, and so on. In this paper, we consider just the cost of testing and debugging. Software reliability growth models give a functional relationship between the error introduction and time to remove it and number of faults staying in the software. A few classes of SRGMs have been proposed and accepted on the literature. Among these Non Homogeneous Poisson Process (NHPP) based SRGM have been broadly considered and utilized as a part of this paper. NHPP-based SRGM considers the failure and debugging process as a counting procedure $\{N(t), t \geq 0\}$, and are shown by a mean number of faults (Kapur & Garg, 1990; Kumar et al., 2012).

In this section, we will demonstrate SRGM to incorporate the impact of fault detection, correction process, and learning capacities of the testing group into improving software reliability. During the testing procedure faults are distinguished on a failure by the fault detection group. The following list of symbols is used in this paper:

m(t): Mean value function of faults remaining in the NHPP model, with m(0) = 0.
a: Initial number of faults in the software when testing of software begins.
$m_d(t)$: Expected number of faults detected by time t.
$m_c(t)$: Expected number of faults corrected by time t.
b(t): Time-dependent rate of fault detection/correction per remaining faults.
α,β: Constant parameter in the learning capacities of the developer.
R(x|T): Pr{no disappointment happens amid (T,T+x)|testing stops at T }
Co2: Cost acquired during testing before release of the software.
Co3: Cost acquired during testing after the release of software.
Co1: Testing cost per unit time.
$C_0(t)$: Total cost spending during the development process by time t.
Tw: Warranty period.
T: Release time of the product.
T*: Optimal release time.
R_0: Desired level of software reliability at release time $(0 < R_o < 1)$.
$C_B(T)$: Budget available by time t.
B: Total budget available to the developer.

Assumption:
(1) The expected number of failures take after an NHPP with mean value function, m(t),
(2) The software fault detection rate is proportional to the expected number of undetected faults,

(3) The proportionality may change with time,
(4) The number of faults in each of the individual intervals is autonomous,
(5) Each time a failure happens; the fault that brought on it is perfectly fixed, and
(6) No new fault is created.

We had expected that the testing stage would be a two-stage process. For the first phase of the testing process, the mean number of shortcoming recognition md(t), is propositional to the mean number of undetected issues staying in the product and can be communicated by taking after differential mathematical statement:

$$m'_d(t) = b(t)\left(a - m_d(t)\right) \quad \ldots (2.1)$$

where,

$$b(t) = \frac{\alpha + \beta t}{1 + bt}$$

Understanding comparison (2.1) with beginning condition md(0)= 0 we get:

$$m_d(t) = a\left(1 - (1+bt)^{\left(\frac{\beta}{b^2} - \frac{\alpha}{b}\right)} e^{-\left(\frac{\beta}{b}\right)t}\right) \quad \ldots (2.2)$$

It can be watched that as $t \to \infty$, $b(t) \to \frac{\beta}{b}$.

In the second stage, the shortcoming amendment rate is relative to the mean number of flaws distinguished however not yet revised deficiencies stay in the framework. In this stage shortcoming amendment rate is accepted as a logistic learning capacity and it can be communicated as far as the differential comparison as:

$$\frac{dm_c(t)}{dt} = b(t)\left(m_d(t) - m_c(t)\right) \quad \ldots (2.3)$$

Where

$$b(t) = \frac{\left(\frac{\beta}{b}\right)}{1 + c\, e^{-\left(\frac{\beta}{b}\right)t}}$$

Fathoming mathematical statement (2.3) with the introductory condition mc(0)= 0 the mean number of shortcomings revised is given by:

What's more, the reliability measure of the software is given as:

$$m_c(t) = \frac{a}{1 + ce^{-\left(\frac{\beta}{b}\right)t}}\left(1 - \left(1 + \frac{(1+bt)^{\left(\frac{\beta}{b^2} - \frac{\alpha}{b} + 1\right)} - 1}{\left(\frac{\beta}{b^2} - \frac{\alpha}{b} + 1\right)}\left(\frac{\beta}{b^2}\right)\right) e^{-\left(\frac{\beta}{b}\right)t}\right) \quad \ldots (2.4)$$

This model is due to Kapur et al. (2009).

The Cost Model

The total cost incorporates the expense of testing, debugging costs during testing and expense of failure and debugging of faults during the operational stage. Testing is performed under a controlled environment. It is normal that the cost of fixing a bug in the testing stage to be not as much as the expense of altering the same in operational stage. The expense of failure and fixing of a fault during an altered warrantee period after the release of the software is incorporated.

In this manner total cost of testing accompanying fuzzy cost capacity can be defined to depict the aggregate expected cost of testing and debugging.

$$\tilde{C}o(T) = \tilde{C}o_1T + \tilde{C}o_2m_c(T) + \tilde{C}o_3\left(m_c(T+T_w) - m_c(T)\right) \quad \ldots (2.5)$$

Symbol ~ marked on the parameters are fuzzy numbers. Here we accept these parameters to be Triangular Fuzzy Numbers (TFN). Also, Reliability is described as:

$$R((T+Ts)|T) = e^{-(mc(T+Ts)-mc(T))} \quad \ldots\ldots\ldots\ldots\ldots\ldots (2.6)$$

The optimization model is given as:

$\widetilde{\text{Minimize}}$ $\tilde{C}o(T)$

Maximize $\tilde{R}(Tw|T)$

Subject to

CB(T) $\tilde{\leq}$ B

$T \geq 0$ ………………………….(P1)

Our aim to minimize the cost and maximize the reliability subject to budget constraint given by management to developer. The two objective functions are fuzzy in nature. The value of the budget constraint assumes budget should be less than the budget allotted by the management. The symbol $\tilde{\leq}$ is called "fuzzy less than or equal to" which defines that the value of budget constraint is not precise but it varies depending on several other factors like debugging team size, team experience etc. The management determines the value of these fuzzy constraints.

Solution of Multi-Objective Fuzzy Model

In the literature,, many researchers have worked on the optimization model of the release time of software as reliability, cost, and budget constraint. The objective function is to minimize the cost or maximize the reliability with the given constraints and to find the optimal time of release of the software. The researchers have worked on the crisp inputs wherein they have considered the crisp values. Since these inputs are not crisp and depend on various factors and can be taken as fuzzy inputs. The fuzzy optimization problems have been studied and

modelled to find the optimal release time. In this paper, we have formulated a fuzzy multi-objective approach to optimal decision "release time" for software system. A numerical example is used to illustrate the proposed fuzzy bi-criteria optimization problem. Crisp optimization techniques cannot be used to deal with the uncertainties in the problem formulation. Hence fuzzy optimization is used to solve problems.

The algorithm below specifies the steps to solve the fuzzy problem formulated in Section 2 and described in equation (P1) (Kapur & Garg, 1990).

Algorithm:

1. Convert the fuzzy optimization problem into crisp problem by using a defuzzification function,

2. Include the objective function of the fuzzifiermax and min as a fuzzy constraint with a restriction or aspiration level,

3. Problem P1 can be restated as:

Find T
s.t $Co(T) \leq C_0$
$R(T_s \mid T) \geq R_0$
$C_B(T) \leq B$
$T \geq 0$ ……(P2)

4. Membership functions are defined for all the fuzzy inequalities. The membership function for the fuzzy less than or equal to and greater than or equal to type are given as:

$$\mu_1(T) = \begin{cases} 1 & ; P \leq P_0 \\ \dfrac{P^* - P}{P^* - P_0} & ; P_0 < P \leq P^* \\ 0 & ; P > P^* \end{cases} \qquad \mu_2(T) = \begin{cases} 1 & ; Q \geq Q_0 \\ \dfrac{Q - Q^*}{Q_0 - Q^*} & ; Q^* \leq Q < Q_0 \\ 0 & ; Q < Q^* \end{cases}$$

$$\mu_3(T) = \begin{cases} 1 & ; B \leq B_0 \\ \dfrac{B^* - B}{B^* - B_0} & ; B_0 < B \leq B^* \\ 0 & ; B > B^* \end{cases}$$ respectively, where

P0, Q0 and B0 are the restriction and aspiration levels set by management, respectively, and P*, Q* and B* are the tolerance levels.

5. By applying the Extension principle to solve the fuzzy decision problem, the crisp mathematical programming problem is given by:

Maximize α
Subject to $\mu_i(T) \geq \alpha$ i=1,2,3 ; $\alpha \geq 0$, $T \geq 0$ ……..(P3)

The above crisp problem can be solved by the standard mathematical programming algorithms.

Problem Solution

Parameters of SRGM have as of now been evaluated from the testing information set by gathered by Brooks and Motley (1980). The assessed estimations of parameters are a= 1334.26, b= 0.119, c=18.902, α= 1.972 and β= 0.024. Further it is accepted that estimations of Co1, Co2, Co3, and TW are known. The release time issue in view of the accompanying information could be dissected. Here we have taken Co1= 6, Co2= 15, Co3= 30, TW= 5. The values of C0 , R0 and B are taken as 20000, 0.99 and 20350 respectively with tolerance levels on cost, reliability and budget constraint as C*= 22000, R*= 0.80, and CB*= 21000. (We have accepted these qualities for representation, however by and by these qualities are situated by the management in light of past experience). Utilizing above estimations of different parameters and constants, arrangement of the issue is acquired with the fuzzy optimization problem examined previously in section 3 above.

Parameters of SRGM have as of now been evaluated from the real time testing information gathered by Brooks and Motley (1980). Their values are given as:

Parameters	a	b	c	α	β
Values	1334.26	0.119	18.902	1.972	0.024

Further it is assumed that values of C1, C2, C3, and TW are known and given by management as follows:

Parameters	Co_1	Co_2	Co_3	Ko_0	Ko*	T_W	R_0	R*
Values	10	25	50	8500	9900	10	0.95	0.7

Further, we restate the problem by using defuzzification function F2(A),we can write:

Min $\tilde{}$ ..... $_2(\tilde{}$ ...

Maximize $F_2(\tilde{}$ ...

Subject to

$CB(T) \square B$

$T \geq 0$ .....(4.1)

where

$$Co(T) = Co_1 T + Co_2 m_c(T) + Co_3 (m_c(T + T_w) - m_c(T))$$

$$R(T_w \mid T) = e^{-(m_c(T+Tw)-m_c(T))}$$

$$C_B(T) = Co_2 * m_c(T) + Co_1 T$$

Substituting these values in the equation we get:

Minimize $Co(T) = 6T + 15*m_c(T) + 30*(m_c(T+5) - m_c(T))$

Maximize $R(T_w \mid T) = e^{-(m_c(T+5)-m_c(T))}$

Subject to $C_B(T) = 15*m_c(T) + 6*T \leq B$

$T \geq 0$

where

$m_c(t) = (1334.26/(1+(18.902*EXP((-0.024*t)/0.119))))*(1-((1+((((1+0.119*t)^((0.024/(0.119*0.119))-(1.972/0.119)+1))-1)/((0.024/(0.119*0.119))-(1.972/0.119)+1))*(0.024/(0.119*0.119)))*EXP((-0.024*t)/0.119)))$

and $m_c(t+t_w) = (1334.26/(1+(18.902*EXP((-0.024*(t+5))/0.119))))*(1-((1+((((1+0.119*(t+5))^((0.024/(0.119*0.119))-(1.972/0.119)+1))-1)/((0.024/(0.119*0.119))-(1.972/0.119)+1))*(0.024/(0.119*0.119)))*EXP((-0.024*(t+5))/0.119)))$

……(4.2)

Restating the problem with fuzzifier min and max objective function and including them as a restriction level constraint. The problem 4.2 can be restated as:

Find T

s.t. $6T + 15*m_c(T) + 30*(m_c(T+5) - m_c(T)) \leq 20000$

$e^{-(m_c(T+10)-m_c(T))} \geq 0.99$

$15*m_c(T) + 6*T \leq 20350$

The membership functions μi(t) ,i =1,2,3 for the fuzzy cost, reliability and budget constraint are given as:

$$\mu_1(T) = \begin{cases} 1 & ; Co(T) \leq 20000 \\ \dfrac{22000 - (6T + 15m_c(T) + 30(m_c(T+5) - m_c(T))}{22000 - 20000} & ; 20000 < Co(T) \leq 22000 \\ 0 & ; Co(T) > 22000 \end{cases}$$

$$\mu_2(T) = \begin{cases} 1 & ; R(Tw \mid T) \geq 0.99 \\ \dfrac{e^{-(m_c(T+5)-m_c(T))} - 0.80}{0.99 - 0.80} & ; 0.99 \leq R(T_w \mid T) < 0.80 \\ 0 & ; R(T_w \mid T) < 0.80 \end{cases}$$

$$\mu_3(T) = \begin{cases} 1 & ; C_B(T) \leq 203500 \\ \dfrac{22000 - 15*m_c(T) + 6*T}{22000 - 15000} & ; 20350 < C_B(T) \leq 21000 \\ 0 & ; C_B(T) > 21000 \end{cases}$$

By applying methodology describe in section 3. The three membership functions are plotted on the scale of time to find the optimal release time. The cost, reliability and budget

constraint membership functions plotted on time scales respectively are shown in figures 1, 2 and 3.

Figure 1.
Cost Membership Vs. Time

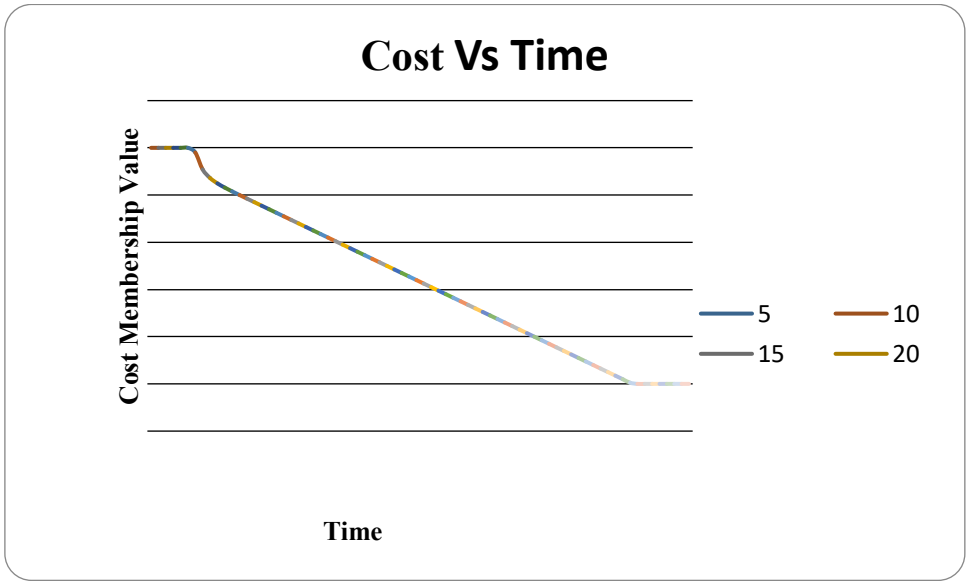


Figure 2.
Reliability Membership Vs. Time

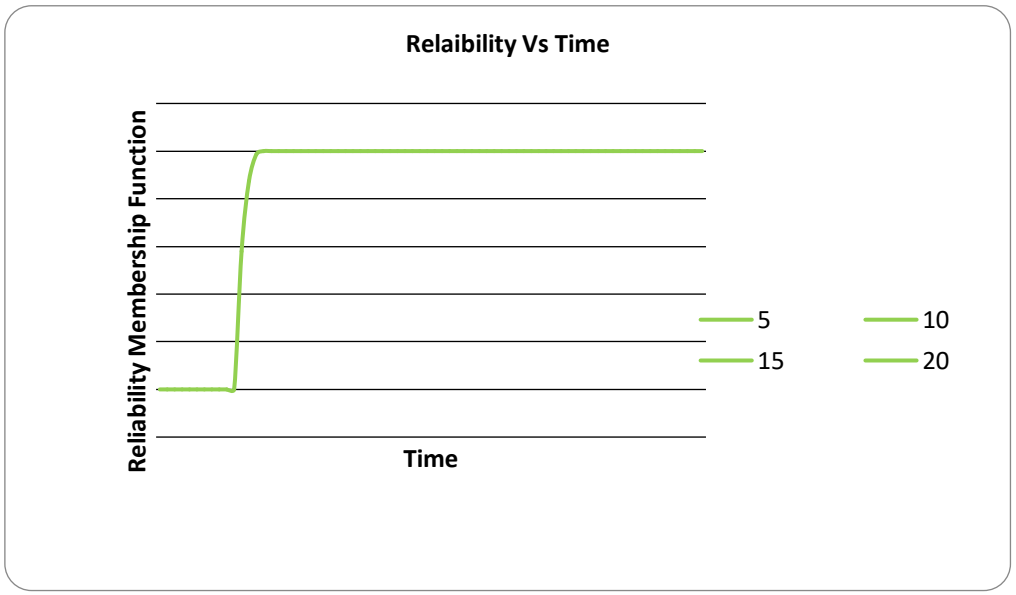


Figure 3.
Budget Membership Vs. Time

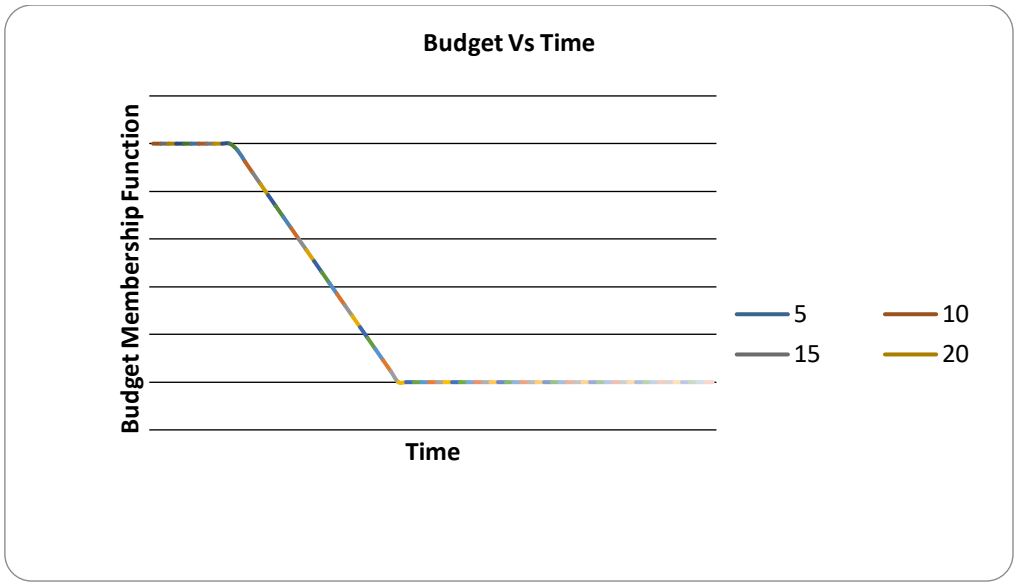
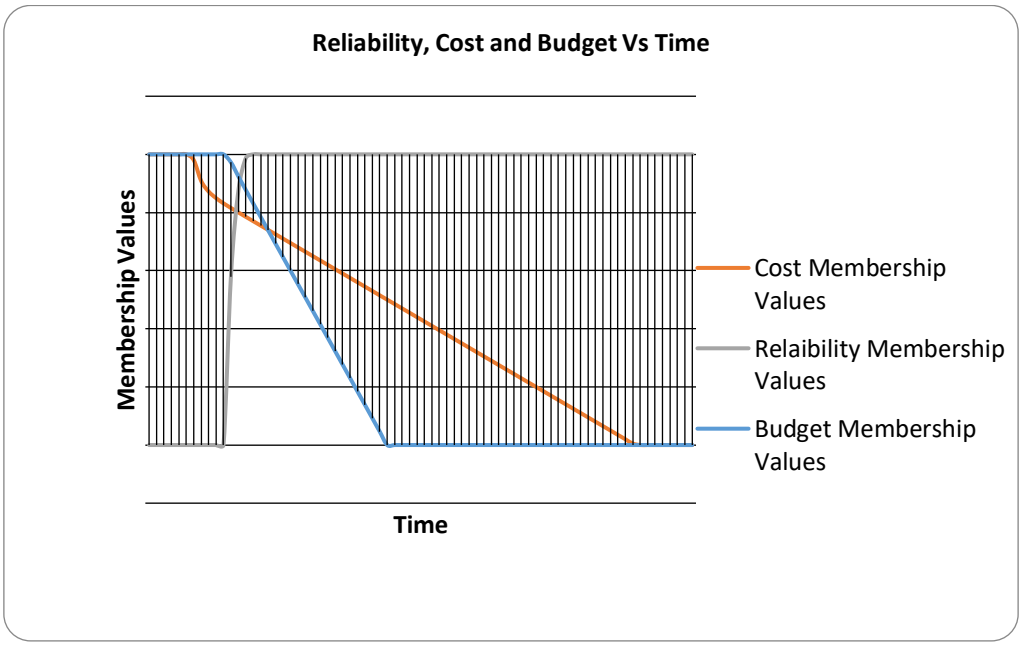


Figure 4.
Reliability, Cost & Budget membership Vs. Time



Taking care of the problem (4.6) from Figure. 4, we acquire ideal release time T= 67 and α= 0.924326.

Conclusion

As we know that software testing is an important phases of software development. In general, software development method incorporates a piece of vulnerabilities and ambiguities. In this paper, we have defined a fuzzy release time issue by minimizing the total cost and maximising the reliability subject to budget constraint set by management to fix vulnerabilities and ambiguities. The issue is examined and solved by fuzzy optimization method on a data set. The numerical illustration is indicated for the given issue of software release time problem for software. If there should arise in occurrence of an infeasible issue, it can be reformulated as Goal programming problem to acquire a negotiable arrangement. Given problem can be reformulated with imperfect debugging and error generation of software. This is an exciting issue of further study in fuzzy optimization problem of software development process and its release.

Conflict of interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

Bellman R.E. and Zadeh L.A. (1973). Decision making in a fuzzy environment, *Management Science*, 17,141-164,1973.

Brooks WD, Motley RW. (1980). Analysis of discrete software reliability models - Technical Report (RADC-TR-80-84), *New York: Rome Air Development Center*, 1980.

Deepak, K., Sharma, S. G., Saini, S., & Mrinal, N. (2012). Development of software reliability S-shaped models. *Review of Business and Technology Research*, 6(1), 101-110.

Goel A.L., Okumoto K. (1979). Time dependent error detection rate model for software reliability and other performance measures, *IEEE Transactions on Reliability*, 28(3): 206-211.

Huang, C. Y., & Lyu, M. R. (2005). Optimal release time for software systems considering cost, testing-effort, and test efficiency. *IEEE transactions on Reliability*, *54*(4), 583-591.

Jha, P. C., Deepak, K., & Kapur, P. K. (2006). Fuzzy release time problem. In *Proceedings of 3rd International Conference in Quality, Reliability and Infocomm Technology (ICQRIT'2006) pp* (pp. 304-310).

Kapur P. K., Jha P. C.**,** Kumar Deepak (2009). *A General Software Reliability Growth Model with different types of learning functions for fault detection and correction processes*, Communications in Dependability and Quality Management An International journal , 12(1), 11– 23.

Kapur P.K. and Garg R.B. (1990). Optimal release policies for software systems with testing effort, *Int. Journal System Science*, 22(9), 1563-1571.

Kapur P.K., Aggarwal S. and Garg R.B. (1994). Bicriterion release policy for exponential software reliability growth model, *RechercheOperationnelle– Operations Research,* 28, 165-180.

Kapur P.K., Garg R.B. and Kumar S.(1999). Contributions to hardware and software reliability, *Singapore: World Scientific Publishing Co. Ltd*., 1999.

Kapur P.K., Kumar Deepak, Gupta A., Jha P.C. (2006). On how to model software reliability growth in the presence of imperfect debugging and fault generation, Proceedings of 2nd International Conference on Reliability & Safety Engineering, 515-523.

Khatri, S. K., Kumar, D., Dwivedi, A., & Mrinal, N. (2012, September). Software Reliability Growth Model with testing effort using learning function. In *2012 CSI Sixth International Conference on Software Engineering (CONSEG)* (pp. 1-5). IEEE.

Kumar, D., & Gupta, P. (2019). Fuzzy software release problem with learning functions for fault detection and correction processes. In *Software Engineering: Proceedings of CSI* 2015 (pp. 655-661). Springer Singapore.

Ohba M. (1984). Software reliability analysis models, *IBM Journal of Research and Development*, 28, 428-443.

Okumoto K. and Goel A.L. (1983). Optimal release time for computer software, *IEEE Transactions On Software Engineering,* SE-9 (3), 323-327.

Peerzada, B., & Kumar, D. (2021, September). Analyzing software vulnerabilities using machine learning. In *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)* (pp. 1-4). IEEE.

Pham H. (2006). *System software Reliability*, Springer, Reliability Engineering Series.

Steuer R.E. (1986). Multiple criteria optimization: *Theory, computation and application*, Wiley, New York, 1986.

Thirez, H. (2000). *OR software LINGO*", *European Journal of Operational Research*, 124, 655-656, 2000.

Xie M. (2003). A Study of the Effect of Imperfect debugging on Software Development Cost, *IEEE Transactions on Software Engineering*, 29 (5).

Yamada, S. and Osaki, S. (1987). Optimal software release policies with simultaneous cost and reliability requirements, *European Journal of Operational Research*, 31, 46-51.

Zimmermann, H.J. (1991). Fuzzy set theory and its applications, *Academic Publisher*.

Bibliographic information of this paper for citing:

Kumar, Deepak; Gupta, Pankaj & Kapur P. K. (2023). Multi- Objective Fuzzy Software Release Problem with learning capacities for fault detection and correction processes. *Journal of Information Technology Management,* 15 (3), 17-30. https://doi.org/ 10.22059/jitm.2023.93621