www.revistageintec.net
ISSN: 2237-0722
revista GEINTEC
Gestão, Inovação e Tecnologias

# An Imperfect Debugging Software Reliability Growth Model: Optimal Release Problems through Warranty Period based on Software Maintenance Cost Model

Ahmad Raad Raheem[1]; Dr. Shaheda Akthar[2]; Dr. Shaik Mohammad Rafi[3]

[1]Research Scholar, Department of Computer Science and Engineering, Acharya Nagarjuna Engineering, A.P, India.

[1]ahmedraadr@yahoo.com

[2]Head of the Department of Computer Science, Government College for Women(A), Guntur, A.P, India.

[2]shahedaakthar76@gmail.com

[3]Associate Professor, Department of Computer Science & Engineering, Gudlavalleru Engg College, Andhrapradesh, India.

**Abstract**

*Software come to be an important element in recent times, from small residence hold gadgets to large machinery wishes fine software. software development is a technical oriented system where range of quantitative and qualitative duties have been completed parallel a good way to meets the needs of the consumer. Many people play a vital role within the improvement of software program product, consequently there is chance of committing errors by way of these humans and these errors becomes faults in later stages. Computing software program cost for the duration of software development can facilitate us predicting the time of release of the software. In this paper we have investigated release time of software program by way of considering the imperfect debugging software program reliability growth model and cost model.*

**Keywords:** Software Testing, Operating Period, Software Reliability, Software Cost, Warranty Time.

## 1. Introduction

The improvement and modernization of industries has increased the more volume of current gadgets, whether the hospitals, commercial enterprise agencies, telecommunication and aeronautics. These contemporary gadgets and computing devises want continually software program hence software program ought to have extra quality and reliability. software program industries struggling from beyond few many years to prepare and design a satisfactory and dependable software as the improvement system its self is fuzzy and complicated in nature. Now a days industries are adapting new strategies and metrics to acquire the meant best quality and reliability. Software reliability was defined as probability of software product could able to work for certain period of time before it could actually fail. Usually, software development activities are amalgamated with several tasks. Software testing is considered to be the most important phase among all the phases where heavy resources were consumed. Software testing phase usually done in systematic environment by considering all resource and time aspects. Past few years several authors have proposed different types of Software reliability growth models under Nonhomogeneous Poisson process models, where each model have its own pros and cons. [1] Software release time and optimal time to stop testing is considered as major issue and studies by many authors under different conditions. [Goel and Okumato 1979][2] on this paper they furnished the stochastic model which follows the Non-homogeneous Poisson method model. These Non-homogeneous reliability growth models widely categorized based on assumption that whether or not the debugging is perfect or it can also generate new mistakes into the software merchandise, which we name it as imperfect debugging. At some point of debugging process there's danger of producing new errors into the software program, hence number of mistakes at some point of trying out might also varies and delivered to the preliminary number of mistakes. [11][12] In this paper he proposed two imperfect debugging software program reliability growth models with the aid of fall introducing rates. [6] [9][10] Delayed s shaped and inflection models were designed based totally on time varying failure detection rate, right here they assumed overall failure stays consistent. software checking out generally divided into stages one testing phase and another operating phase. [15] Yamada 1994 has derived most effective time of software program shipping primarily based on warranty period and maintenance cost model. Essentially, very software product has guarantee duration for the duration of working section which become additionally considered even as computing the entire preservation cost. software warranty was implemented on any new software product which was decided by way of industry based on in its enterprise rules. So, at the same time as designing the software cost model it's far higher to considered the warranty length. [4][7] Many authors have proposed diverse aggregate of cost models with the aid of contemplating of product reliability and warranty duration.

[14] Yamada 1993 in this paper proposed cost model and computed optimal delivery time by considering the discount rate. [Zhang 1998] [13] he proposed new cost model based on warranty period and additionally added risk cost. [Kimura 1999][5] In this he proposed new warranty cost model and derived the optimal delivery time based on warranty period and reliability of the product. [8] In this paper author has computed the optimal delivery time based on imperfect debugging by considering perfect debugging factor.

In this paper we have investigated optimal delivery time of software product by using fault introducing imperfect debugging software reliability growth model under warranty cost model. Section II describes the Non-homogeneous Poisson process software reliability models. Section III derives the various policies related to warranty cost model. Section IV deliberate the results of various policies with imperfect debugging models. Section V designates conclusion.

## 2. Non-homogeneous Poisson Process Models

These Nonhomogeneous Poisson process software reliability growth models usually have the stochastic in nature [15]. These models have failure intensity function which was represented by $h_m(t)$. N(t) represents the number of failures in the software at a time t and m(t) represents the mean or expectation. This mean value is computed through by the following equation. The relation between mean and intensity of failure data was given by E[N(t)]=m(t) and the relation between mean and failure intensity is given by [3]

$$m(t) = \int_0^t h_m(t)\, dt \qquad (1)$$

N(t) have the probability mass function which was given by

$$Pr\{N(t) = n\} = \frac{(m(t))^n \times e^{-m(t)}}{n!} \; n = 0,1,2..\infty \qquad (2)$$

Based on Yamada ‘s et.al 1992 [11] proposed model where imperfect debugging software reliability growth model with faults introducing.

$$\frac{dm(t)}{dt} = b_2(a(t) - pm(t)) \; (p > 0 \text{ and } b_2 > 0) \qquad (3)$$

In the equation (3) $b_2$ representing the failure detection rate which is constant, a(t) is initial faults present in the software before testing here in this expression which is a time varying function and p represents the rate of fault removal efficiency. Now let us assume that a(t) is given by

$$a(t) = \alpha_2(1 + \gamma t) \; (\alpha_2 > 0 \text{ and } \gamma > 0) \qquad (4)$$

Solving the equation (3) and (4) at m(0)=0, we get the ultimate form of m(t) as

$$m(t)e^{\int_o^t pb_2 dt} = \int_0^t b_2\, \alpha_2(1 + \gamma t)e^{pb_2 t}dt \qquad (5)$$

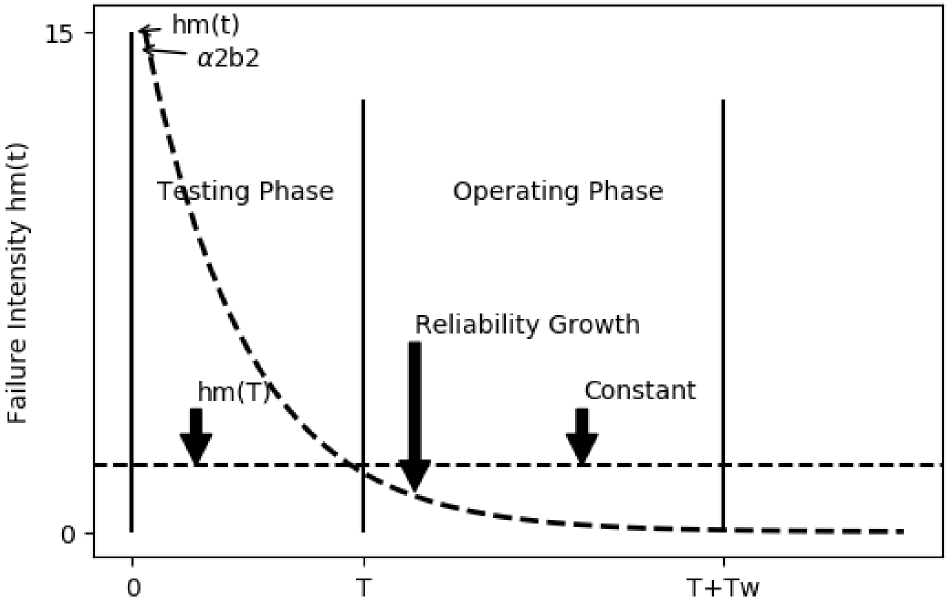
$$m(t)e^{pb_2 t} = b_2\alpha_2 \left\{\int_0^t e^{pb_2 t}dt + \gamma \int_0^t te^{pb_2 t}dt\right\} \qquad (6)$$

$$m(t) = \frac{\alpha_2}{p}\left\{\left(1 - \frac{\gamma}{pb_2}\right)(1 - e^{-pb_2 t}) + \gamma t\right\} \qquad (7)$$

Now the failure intensity is derived by

$$h_m(t) = \frac{dm(t)}{dt} = \frac{\alpha_2}{p}\{(pb_2 - \gamma)(e^{-pb_2 t}) + \gamma\} \qquad (8)$$

Fig. 1 - Software Reliability Growth Model During the Warranty Period




From the Fig.1 the curve shows the failure intensity which was denoted by $h_m(t)$, and declines during software testing progresses. T denotes the total testing time after which software is transition from testing phase to operating phase. $T_w$ denoted the warranty period which was associated with software product after transition from testing phase to operating phase. $h_m(T)$ denotes the failure intensity of software product at particular constant time and where the reliability remains constant.

## 3. Warranty Cost Model

Total cost of the software product is considered to be the cost associated with testing phase and operating phase. Following equation (9) shows the relation between total cost, and cost incurs during testing and operating phase. By optimizing the equation (9) with respect to time T then we get the optimal time to release and stop testing [15].

$$C(T) = C_0 + C_t T + C_w(T) \quad (9)$$

$C_0$= miscellaneous cost or startup cost

$C_t$= test cost per unit time incurs during testing phase

$C_w$= cost incurs for while detecting and correcting errors during warranty phase.

$W(t)$= Length of the warranty period determined from the end of the test distribution function for (t >0)

Optimal policy I: In this cost incurs at warranty period during operating phase follows the expression. Here in this Case $T_w$ warranty period is constant, $W(t)=U(t-T_w)$ and the end of the testing reliability growth stops, where $U(t) = 1$ $(t \geq 0)$ and $U(t) = 0$ $(t < 0)$ at this time

$$C_w(T) = C_w h_m(t) T_w \tag{10}$$

Now cost function

$$C(T) = C_0 + C_t T + C_w h_m(t) T_w \tag{11}$$

We can get the optimal value of release time $T^*$ by differentiating equation (11) and equating it to zero.

When the time T=0 then from the equation (11) we get

$$C(0) = C_0 + C_w \alpha_2 b_2 T_w \tag{12}$$

In the same way when $T=\infty$ then equation (11) becomes

$$C(\infty) = \infty \tag{13}$$

$$\frac{C(T)}{dT} = C_t + C_w T_W \frac{dh_m(T)}{dT} = 0 \tag{14}$$

Solving above equation (14) on the condition that if $h_m(0) > \left(\gamma + \frac{C_t}{C_w T_w b_2 p}\right)$ we get optimal time for stop testing.

$$T^* = \frac{1}{pb_2} ln\left\{\frac{\alpha_2 C_w T_w (pb_2 - \gamma) b_2}{C_t}\right\} \tag{15}$$

If $h_m(0) \leq \left(\gamma + \frac{C_t}{C_w T_w b_2 p}\right)$ then optimal release time $T^*=0$.

Optimal policy II: In this we assumes the cost incurs in the operating phase is given by equation (16). In this software reliability growth continuous after the time period T.

$$C_w(T) = C_w[m(T+T_w) - m(T)] \tag{16}$$

Now

$$\frac{C(T)}{dT} = C_t + C_w \frac{d[m(T+T_w) - m(T)]}{dT} = 0 \tag{17}$$

Solving above equation (17) and applying condition $h_m(0) > \left(\frac{C_t}{C_w\left(1-e^{-pb_2 T_w}\right)} + \gamma\right)$ then optimal release time

$$T^* = \frac{1}{pb_2} ln\left\{\frac{C_w \alpha_2 b_2 \left(1-\frac{\gamma}{pb_2}\right)\left(1-e^{-pb_2 T_w}\right)}{C_t}\right\} \tag{18}$$

If the condition $h_m(0) \leq \left(\frac{C_t}{C_w\left(1-e^{-pb_2 T_w}\right)} + \gamma\right)$ then optimal release time or stop time $T^*=0$.

Optimal policy III: In this we assumes the cost incurs in the operating phase is given by equation (19). In this warranty period follows the some general distribution W(t), reliability growth continuous after the testing period is over.

$$C_w(T) = C_w \int_0^{\infty}[m(T+T_w) - m(T)]dW(T_w) \qquad (19)$$

In this we used exponential distribution function of the warranty period.

$$W(t) = (1 - e^{-\mu t})\ ;\ (\mu > 0 \text{ and } t > 0) \qquad (20)$$

Now

$$\frac{C(T)}{dT} = C_t + C_w \frac{d(\int_0^{\infty}[m(T+T_w)-m(T)]dW(T_w))}{dT} = 0 \qquad (21)$$

Solving the above equation (21) and satisfying the condition $h_m(0) > \left(\frac{C_t(pb_2+\mu)}{b_2 C_w} + \gamma\right)$ we get the optimal solution as

$$T^* = \frac{1}{pb_2} ln\left\{\frac{\alpha_2 C_w b_2(pb_2-\gamma)}{pC_t(pb_2+\mu)}\right\} \qquad (22)$$

If $h_m(0) \leq \left(\frac{C_t(pb_2+\mu)}{b_2 C_w} + \gamma\right)$ then the optimal release time and stop time is $T^*=0$.

## 4. Numeric Example

To compute the optimal release time of the software we consider the equation (7) model for software reliability growth model and assumed the values to the parameters present in the model, $\alpha_2$=1000, $b_2$=0.05, p=2.0, and $\gamma$=0.03. Also, during warranty period fault processing cost $C_w$ =1.0 and minimum startup cost $C_0$=1000.

The values which was shown in the Table-1 represents the optimal release time of software by considering the Optimal release policy 1, where by varying the parameters values of test cost per unit time $C_t$ and warranty period $T_w$.

Table I - Optimal Release Time of Software by Warranty Period (Optimal Release Policy 1)

| Tw \ Ct | 0.5 | 1 | 5 | 10 | 20 | 50 |
|---|---|---|---|---|---|---|
| **1** | 19.46 | 12.53 | 0 | 0 | 0 | 0 |
| **2** | 26.39 | 19.46 | 3.36 | 0 | 0 | 0 |
| **5** | 35.55 | 28.62 | 12.53 | 5.6 | 0 | 0 |
| **10** | 42.48 | 35.55 | 19.46 | 12.53 | 5.6 | 0 |
| **20** | 49.42 | 42.48 | 26.39 | 19.46 | 12.53 | 3.36 |
| **50** | 58.58 | 51.65 | 35.55 | 28.62 | 21.69 | 12.53 |
| **100** | 65.51 | 58.58 | 42.48 | 35.55 | 28.62 | 19.46 |

From the Table-I it is observed that by varying $T_w$ and keeping the $C_t$ constant, the release time of the software increase.

Table -2 represents the software release time for second policy.

The second policy is more near to practical than the first policy. In the second policy it is observed that software reliability is improved after the test is completed.

Table 2 - Optimal Release Time of Software by Warranty Period (Optimal Release Policy 2)

| **Tw\Ct** | **0.5** | **1** | **5** | **10** | **20** | **50** |
|---|---|---|---|---|---|---|
| **1** | 22.52 | 15.59 | 0 | 0 | 0 | 0 |
| **2** | 28.97 | 22.04 | 5.94 | 0 | 0 | 0 |
| **5** | 36.72 | 29.79 | 13.69 | 6.76 | 0 | 0 |
| **10** | 41.46 | 34.53 | 18.43 | 11.5 | 4.57 | 0 |
| **20** | 44.59 | 37.66 | 21.56 | 14.63 | 7.7 | 0 |
| **50** | 45.98 | 39.05 | 22.95 | 16.02 | 9.09 | 0 |
| **100** | 46.04 | 39.11 | 23.02 | 16.09 | 9.15 | 0 |

Table-3 describes the optimal release time for the policy 3. In this policy optimal release time is computed by varying the values of a parameter $\mu$ and $C_t$.

Table 3 - Optimal Release Time of Software by Warranty Period (Optimal Release Policy 3)

| **μ\Ct** | **0.5** | **1** | **5** | **10** | **20** | **50** |
|---|---|---|---|---|---|---|
| **0.1** | **51.81** | **37.94** | **5.75** | **0** | **0** | **0** |
| **0.05** | **59.91** | **46.05** | **13.86** | **0** | **0** | **0** |
| **0.01** | **70.13** | **56.27** | **24.08** | **10.22** | **0** | **0** |
| **0.005** | **71.87** | **58.01** | **25.82** | **11.96** | **0** | **0** |
| **0.001** | **73.38** | **59.52** | **27.33** | **13.47** | **0** | **0** |

## 5. Conclusion

In this paper we have discussed about one major problem in software development industries, release time determination. For the purpose of study, we have used imperfect debugging software reliability growth model with faults introduction. We had discussed about cost incurs during testing and operating phase of software development. Further we have derived the relation between warranty period $T_w$ and $C_t$ unit cost incurs during the software testing with the help of three optimal release policies. Policy 1 describes that software reliability remains constant during testing period, thus there is no further improvement in the software reliability and ultimately remain constant. Policy 2 is more realistic than policy 1, it assumes that after reliability growth occurs even after the testing period completes and during warranty period. Policy 3 proposed that length of warranty period $T_w$ which follows the exponential curve. Concerned results were describe through Table 1,2 and 3.

## References


S.R. Dalal and C. L. Mallows, “When Should One Stop Testing Software?,” *Journal of the American Statistical Association*, vol. 83, no. 403, pp. 872–879, Sep. 1988. 10.1080/01621459.1988.10478676.

A. L. Goel and K. Okumoto, “Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures,” *IEEE Trans. Rel.*, vol. R-28, no. 3, pp. 206–211, Aug. 1979. 10.1109/TR.1979.5220566.

S. S. Gokhale and K. S. Trivedi, “Log-logistic software reliability growth model,” in *Proceedings Third IEEE International High-Assurance Systems Engineering Symposium (Cat. No.98EX231)*, Washington, DC, USA, 1998, pp. 34–41. doi: 10.1109/HASE.1998.731593.

P. K. Kapur, S. Yamada, A. G. Aggarwal, and A. K. Shrivastava, “Optimal Price and Release Time of a Software Under Warranty,” *Int. J. Rel. Qual. Saf. Eng.*, vol. 20, no. 03, p. 1340004, Jun. 2013, doi: 10.1142/S0218539313400044.

M. Kimura, T. Toyota, and S. Yamada, “Economic analysis of software release problems with warranty cost and reliability requirement,” *Reliability Engineering & System Safety*, vol. 66, no. 1, pp. 49–55, Oct. 1999, doi: 10.1016/S0951-8320(99)00020-4.

M. Ohba, “Inflection S-Shaped Software Reliability Growth Model,” in *Stochastic Models in Reliability Theory*, vol. 235, S. Osaki and Y. Hatoyama, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1984, pp. 144–162. doi: 10.1007/978-3-642-45587-2_10.

K. Rinsaka and T. Dohi, “Determining the optimal software warranty period under various operational circumstances,” *Int J Qual & Reliability Mgmt*, vol. 22, no. 7, pp. 715–730, Sep. 2005, doi: 10.1108/02656710510610857.

D. R. P. Williams, “Study of the Warranty Cost Model for Software Reliability with an Imperfect Debugging Phenomenon,” p. 13, 2007.

S. Yamada and S. Osaki, “Software Reliability Growth Modeling: Models and Applications,” *IIEEE Trans. Software Eng.*, vol. SE-11, no. 12, pp. 1431–1437, Dec. 1985, doi: 10.1109/TSE.1985.232179.

S. Yamada, M. Ohba, and S. Osaki, “s-Shaped Software Reliability Growth Models and Their Applications,” *IEEE Trans. Rel.*, vol. R-33, no. 4, pp. 289–292, Oct. 1984, doi: 10.1109/TR.1984.5221826.

S. Yamada, K. Tokuno, and S. Osaki, “Imperfect debugging models with fault introduction rate for software reliability assessment,” *International Journal of Systems Science*, vol. 23, no. 12, pp. 2241–2252, Dec. 1992, doi: 10.1080/00207729208949452.

S. Yamada, K. Tokuno, and S. Osaki, “Software reliability measurement in imperfect debugging environment and its application,” *Reliability Engineering & System Safety*, vol. 40, no. 2, pp. 139–147, Jan. 1993, doi: 10.1016/0951-8320(93)90104-7.

X. Zhang and H. Pham, “A software cost model with warranty cost, error removal times and risk costs,” *IIE Transactions*, vol. 30, no. 12, pp. 1135–1142, Dec. 1998, doi: 10.1080/07408179808966570.

S. Yamada, M. Kimura, E. Terane, S. Osaki, “Optimal software release problems with life-cycle distribution and discount rate”, Trans. *IPS Japan*, Vol. 34(5), pp. 1188-1197, 1993 (in Japanese).

S. Yamada, “Optimal release problems with warranty period based on a software maintenance cost model”, *Trans. IPS Japan*, Vol. 35 (9), pp. 2197-2202, 1994.