REVUE FRANÇAISE D'AUTOMATIQUE, D'INFORMATIQUE ET DE RECHERCHE OPÉRATIONNELLE. RECHERCHE OPÉRATIONNELLE

P. K. KAPUR

R. B. GARG

Optimal sofware release policies for software reliability growth models under imperfect debugging

Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle, tome 24, n° 3 (1990), p. 295-305.

<http://www.numdam.org/item?id=RO_1990__24_3_295_0>

Recherche opérationnelle/Operations Research
(vol. 24, n° 3, 1990, p. 295 à 305)

OPTIMAL SOFTWARE RELEASE POLICIES FOR SOFTWARE RELIABILITY GROWTH MODELS UNDER IMPERFECT DEBUGGING (*)

by P. K. KAPUR (1) and R. B. GARG (2)

Abstract. – *In this paper we discuss two software reliability growth models (SRGMs) under imperfect debugging based on non-homogeneous Poisson process (NHPP). Parameters of the models are estimated and optimal release policies are discussed. Numerical results are presented at the end.*

Keywords : Release policies; software; Imperfect debugging; Life cycle.

Résumé. – *Nous examinons dans cet article deux modèles de croissance de fiabilité de logiciels sous l'hypothèse de correction imparfaite des erreurs, en se basant sur un processus de Poisson non homogène. Les paramètres du modèle sont estimés et les politiques optimales de mise à disposition publique sont examinées. Nous terminons avec des résultats numériques numériques.*

INTRODUCTION

Software error occurance phenomenon has been studied extensively in the literature with the objective of improving software performance. After the software has been developed its life cycle may be considered to consist of two phases e. g., testing phase and operation phase. During both these phases software is run and may fail at times due to errors remaining in it. On a failure an attempt is made to correct the cause of the failure. However, it is not always possible to find the cause of the failure and remove it. This may be attributed to lack of sufficient knowledge about the software, poor documentation of the software, etc.

It is also of utmost importance to find the appropriate release time of the software. If the release of the software is unduly delayed, the manufacturer

(*) Received in October 1989, revised in January 1990.
(1) Department of Operational Research, University of Delhi, Delhi-110 007, India.
(2) Department of Computer Science, University of Delhi, Delhi-110 007, India.

Recherche opérationnelle/Operations Research, 0399-0559/90/03 295 11/$ 3.10

may suffer in terms of penalties and revenue loss, while a premature release may cost heavily in terms of fixes to be done after release and may even harm manufecturer's reputation. Therefore, manufecturer must have some idea about the possible attributes of the software like its initial error content, failure rate, reliability at time t and its potential release time. Several models have been developed in the past to estimate the attributes of a software system. Review articles by Shanthikumar [7], Yamada and Osaki [8], Goel [2] summarise most of these models. All of them assume that the error removal phenomenon is perfect. However, in reality this is not always true.

In this paper we develop two models based on NHPP incorporating the concept of imperfect debugging. The two models considered are exponential and modified exponential reliability growth models based on NHPP. Parameters of the models are estimated and optimal release policies based on cost and reliability criterian are discussed. Total cost incurred on the software until it is supported also includes the cost incurred on those failures which could not be removed. Our release policies also tend to minimise such a wasteful expenditure. Optimal release policies for similar models under perfect debugging can be found in Okumoto and Goel [5], Yamada *et al.* [12, 13]. Finally numerical examples are also presented at the end.

MODEL I

Assumptions

1. Software system is subject to failures during execution caused by errors remaining in the software.

2. Failure rate of the software is equally affected by errors remaining in the software.

3. At any time the failure rate of the software is proportional to the errors remaining in the software at that time.

4. On a failure instantaneous repair effort starts and the following may occur:

(*a*) fault contents are reduced by one with probability p_0;

(*b*) fault contents are unchanged with probability $1-p_0$.

It is assumed that $p_0 \geqslant 1-p_0$.

5. Software life cycle length is assumed to be more than optimum release time.

6. The error removal phenomenon in the software is modelled by NHPP.

Notations

a: initial error content.

b: proportionality constant (failure rate per error).

p_0: pr { repair effort removes a fault }.

$m_2(t)$: mean number of faults removed in the software till time t.

$m(t)$: mean number of failures in $(0, t]$.

$C_1(C_2)$: cost incurred on a perfect (imperfect) debugging effort before release of the software system.

$C_3(C_4)$: cost incurred on a perfect (imperfect) debugging effort after release of the software system ($C_3 > C_1$, $C_4 > C_2$).

C_5: testing cost per unit time.

T^*: optimal release time.

T_c: software life cycle length.

Analysis of model I

The following differential equation may easily be written

$$dm_2(t)/dt = bp_0(a - m_2(t)) \tag{1}$$

and solving it we get

$$m_2(t) = a(1 - \exp(-bp_0 t)). \tag{2}$$

Mean number of failures in $(0, t]$ is given by

$$m(t) = a(1 - \exp(-bp_0 t))/p_0. \tag{3}$$

This model has also been discussed in [11].

If $p_0 = 1$, $m(t) = m_2(t)$ and coincides with Goel and Okumoto [1]. The NHPP intensity function is given by

$$\lambda(t) = ab \exp(-bp_0 t) \tag{4}$$

It may be noted that $\lambda(t)$ is decreasing in t with $\lambda(0) = ab$ and $\lambda(\infty) = 0$.

Estimation of parameters

Suppose the data on n failure occurance times $s(s_1, s_2, \ldots, s_n)$ where $(0 < s_1 \leqq s_2 \ldots \leqq s_n)$ are observed during testing. Then the likelihood function for the unknown parameters a and b (assuming p_0 known) in the NHPP

model with $m(t)$ given s is

$$L = \exp(-m(s_n)) \prod_{i=1}^{n} \lambda(s_i) \tag{5}$$

(refer [1]).

Taking log of the likelihood function maximum likelihood extimates $\hat{a}$ and $\hat{b}$ can be obtained.

Alternatively, suppose the data on cumulative number of failures $y_k (0 < y_1 \leqq y_2 \ldots \leqq y_n)$ in a given time interval $(0, t_k]$ are observed $[(k = 1, 2, \ldots n), (0 < t_1 < t_2 \ldots < t_n)]$. Then the likelihood function for the unknown parameters a and b is

$$L = \prod_{k=1}^{n} \frac{[m(t_k) - m(t_{k-1})]^{y_k - y_{k-1}}}{(y_k - y_{k-1})!} \exp[-(m(t_k) - m(t_{k-1}))] \tag{6}$$

(refer [10]).

Taking log of the likelihood function, maximum likelihood estimates $\hat{a}$ and $\hat{b}$ (assuming p_0 known) can be obtained.

Now, to estimate a and b (p_0 is known) in (3) we make use of the data given in [1]. The data is on time intervals between software failures and consists of 26 observations during testing. We have assumed that the data pertains to failures under imperfect debugging. Using the likelihood function in (5) the maximum likelihood estimates of a and b (assuming $p_0 = 0.8$) are $\hat{a} = 27.195$, $\hat{b} = 0.007238$.

Optimal release policy

We determine a software release time T such that the total expected software cost is minimised or total expected gain is maximised subject to software reliability being not less than a specified reliability objective.

Mathematically, we may state, minimise

$$C(T) = (C_1 p_0 + C_2 (1 - p_0)) m(T) + (C_3 p_0 + C_4 (1 - p_0)) . (m(T_c) - m(T)) + C_5 T. \tag{7}$$

or maximise

$$g(T) = ((C_3 - C_1) p_0 + (C_4 - C_2)(1 - p_0)) m(T) - C_5 T \tag{8}$$

subject to

$$R(x \mid T) = \exp[-(m(T + x) - m(T))] \geqq R_0 \tag{9}$$

where $0 < R_0 < 1$ and $x > 0$.

It may be observed that maximising gain is same as minimising cost.

From the cost function, it is observed that

$$dC(T)/dT = 0 \quad \text{if} \quad \lambda(T) = C_5/(D_2 - D_1) \tag{10}$$

where

$$D_1 = C_1 p_0 + C_2(1 - p_0) \quad \text{and} \quad D_2 = C_3 p_0 + C_4(1 - p_0).$$

Thus if $ab > C_5/(D_2 - D_1)$, finite and unique $T = T_0 (> 0)$ satisfying (10) exists. While if $ab \leqq C_5/(D_2 - D_1)$, $dC(T)/dT > 0$ for $T > 0$.

Moreover, for a specific operational time requirement $x > 0$ and reliability objective R_0, it is evident that if $R(x \mid 0) < R_0 < 1$, finite and unique $T = T_1 (> 0)$ exists satisfying

$$R(x \mid T) = R_0 \tag{11}$$

while if $0 < R_0 \leqq R(x \mid 0)$, $R(x \mid T) > R_0$ for $T > 0$.

Combining the cost and reliability requirements we may state the following theorem for optimal release policy.

THEOREM 1: *Assuming* $C_3 > C_1 > 0$, $C_4 > C_2 > 0$, $C_5 > 0$, $x > 0$ *and* $0 < R_0 < 1$:

(1) *if* $ab > C_5/(D_2 - D_1)$ *and* $R(x \mid 0) < R_0 < 1$, $T^* = \max(T_0, T_1)$;

(2) *if* $ab > C_5/(D_2 - D_1)$ *and* $0 < R_0 \leqq R(x \mid 0)$, $T^* = T_0$;

(3) *if* $ab \leqq C_5/(D_2 - D_1)$ *and* $R(x \mid 0) < R_0 < 1$, $T^* = T_1$;

(4) *if* $ab \leqq C_5/(D_2 - D_1)$ *and* $0 < R_0 \leqq R(x \mid 0)$, $T^* = 0$.

Numerical example: Using $\hat{a} = 27.195$, $\hat{b} = 0.007238$ and $P_0 = 0.8$, we discuss optimal policy for the software system described in [1]. We assume $C_1 = 1.1$, $C_2 = 0.6$, $C_3 = 10.0$, $C_4 = 10.0$, $C_5 = 1.0$, $T_c = 500.0$, $x = 2.0$ and $R_0 = 0.8$. Using these parameters, $T_0 = 99.0$, $T_1 = 97.0$ and $R(2.0 \mid 0) < 0.8$. From Theorem 1 we get $T^* = \max(T_0, T_1) = 99.0$, *i.e.* software should be released after testing for 99.0 units of time (*figs.* 1 and 2).

MODEL II

1. Failure rate of the software is affected by type I (minor) and type II (major) errors remaining in the software.

2. On a type I (type II) failure, instantaneous repair effort starts and the following may occur:

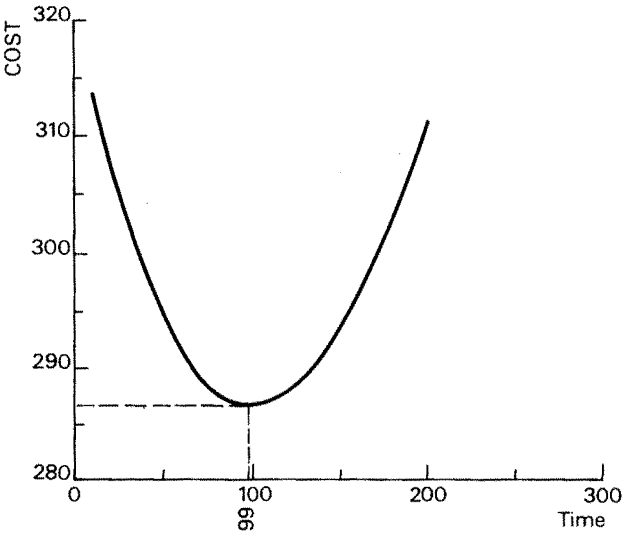

Figure 1

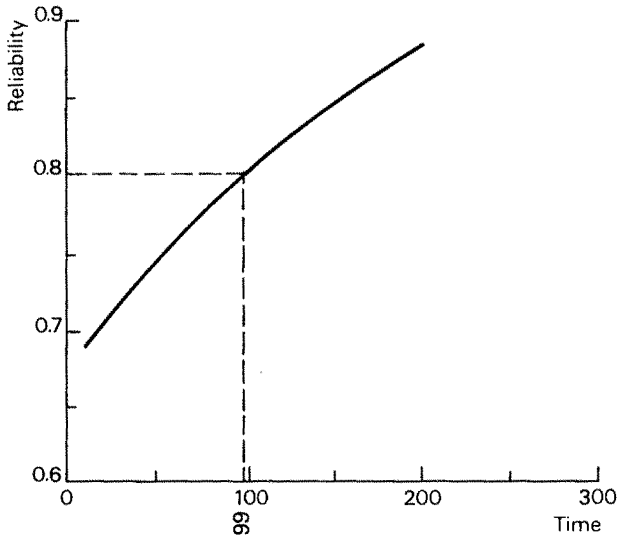

Figure 2

(*a*) fault contents are reduced by one with probability p_1 (p_2);

(*b*) fault contents remain unchanged with probability $(1-p_1)((1-p_2))$;

(*c*) it is assumed that $p_1 \gg (1-p_1)$ and $p_2 \gg (1-p_2)$.

3. Failure rate of the software due to type $i (i=1,2)$ errors is proportional to the remaining type $i (i=1,2)$ errors in the software.

Other assumptions are same as in Model I.

Notations

a: initial error content.

$r_1 (r_2)$: proportion of type I (II) errors in the software, $r_1 + r_2 = 1$.

b_i: proportionality constant for type i $(i=1, 2)$ errors (failure rate per type i error).

p_i: pr {repair effort for a type i $(i=1,2)$ failure removes the fault}.

$m_{2i}(t)$: mean number of type i faults removed in the software till time $t (i=1,2)$.

$m_i(t)$: mean number of failures due to type i errors in $(0, t]$ $(i=1,2)$.

$C_{1i}(C_{2i})$: cost incurred on a perfect (imperfect) debugging effort on a failure due to type i error before release of the software system $(i=1,2)$.

$C_{3i}(C_{4i})$: cost incurred on a perfect (imperfect) debugging effort on a failure due to type i error after release of the software system $(C_{3i} > C_{1i}$, $C_{4i} > C_{2i})$ $(i=1,2)$.

C_5: testing cost per unit time.

T^*: optimal release time.

T_c: software life cycle length.

Analysis of Model II

Proceeding as in the case of Model I, mean number of type i errors removed in $(0, t]$ is given by

$$m_{2i}(t) = r_i a (1 - e^{-b_i p_i t}), \qquad i=1,2 \tag{12}$$

mean number of failures due to type i errors is, therefore, given by

$$m_i(t) = \frac{r_i a}{p_i} (1 - e^{-b_i p_i t}), \qquad i=1,2 \tag{13}$$

and

$$m(t) = \sum_{i=1}^{2} m_i(t).$$

The NHPP intensity function is given by

$$\lambda(t) = a \sum_{i=1}^{2} r_i b_i e^{-b_i p_i t}. \tag{15}$$

Estimation of parameters

Proceeding as in the case of Model I we make use of the data given in [3] to estimate a, b_1 and b_2 (p_1, p_2 known). The data is on the number of major (major + critical) and minor errors detected on a weekly basis and consists of 38 weeks of test data. Using the likelihood function in (6) maximum likelihood estimates of a, b_1 and b_2 are obtained as $\hat{a} = 424.36$, $\hat{b_1} = 0.3983 \times 10^{-3}$, $\hat{b_2} = 0.1370 \times 10^{-3}$ (assuming $p_1 = 0.8$ and $p_2 = 0.6$) where $r_1 = 0.64$ and $r_2 = 0.36$.

Optimal release policy

Proceeding as in Model I, optimal release policy can be obtained, which is summarised in Theorem 2.

THEOREM 2: *Assuming* $C_{3i} > C_{1i} > 0$, $C_{4i} > C_{2i} > 0$, $C_5 > 0$, $x > 0$, $0 < R_0 < 1$ *and let*

$$D_{1i} = C_{1i} p_i + C_{2i}(1 - p_i),$$

$$D_{2i} = C_{3i} p_i + C_{4i}(1 - p_i) \qquad (i = 1, 2)$$

(1) *if* $a \sum_{i=1}^{2} (D_{2i} - D_{1i}) r_i b_i > C_5$ *and* $R(x \mid 0) < R_0 < 1$, $T^* = \max(T_0, T_1)$;

(2) *if* $a \sum_{i=1}^{2} (D_{2i} - D_{1i}) r_i b_i > C_5$ *and* $0 < R_o \leqq R(x \mid 0)$, $T^* = T_0$;

(3) *if* $a \sum_{i=1}^{2} (D_{2i} - D_{1i}) r_i b_i \leqq C_5$ *and* $R(x \mid 0) < R_0 < 1$, $T^* = T_1$.

(4) *if* $a \sum_{i=1}^{2} (D_{2i} - D_{1i}) r_i b_i \leqq C_5$ *and* $0 < R_0 \leqq R(x \mid 0)$, $T^* = 0$.

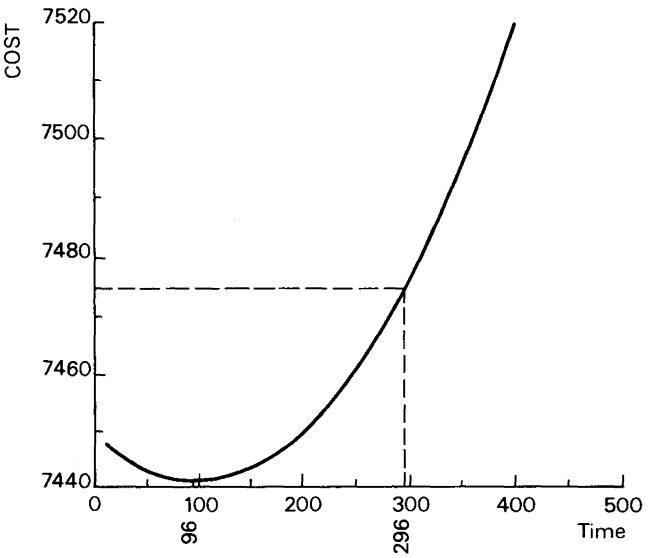

Figure 3

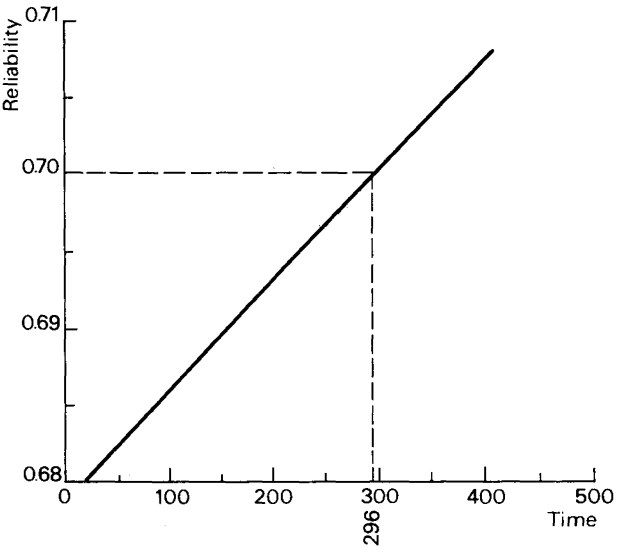

Figure 4

Numerical example: Using $\hat{a}=424.36$, $\hat{b}_1=0.3983\times10^{-3}$, $\hat{b}_2=0.1370\times10^{-3}$ and $p_1=0.8$, $p_2=0.6$ ($r_1=0.64$, $r_2=0.36$) we discuss optimal release policy for the software system described in [3]. We assume $C_{11}=2.2$, $C_{21}=1.2$, $C_{12}=12.0$, $C_{22}=7.0$, $C_{31}=C_{41}=40.0$, $C_{32}=400.0$, $C_{42}=150.0$, $C_5=10.0$,

$T_c = 750.0$, $x = 3.0$, $R_0 = 0.7$. Using these parameters, $T_0 = 96.0$, $T_1 = 296.0$ and $R(3.0 \mid 0) < 0.7$. From Theorem 2 we get $T^* = \max(T_0, T_1) = 296.0$ (*figs.* 3 and 4).

Particular cases

If we assume perfect debugging $(p_1 = p_2 = 1)\, m(t)$ in (14) agrees with [10]. If we further assume $b_1 = b_2$, $m(t)$ agrees with [1,5].

CONCLUSION

We have considered two models based on NHPP with imperfect debugging and discussed optimal release policies based on cost-reliability criterian. Cost also includes the cost incurred on those failures which could not be fixed during the development and operational phases. Similar extensions are possible for *S*-shaped [13] and Inflection *S*-shaped [4] SRGMs. However, we feel in these cases, it is better to replace the reliability constraint with failure intensity function, which must decrease ultimately for any software error detection phenomenon. However, these results will be brought out in a future paper.

ACKNOWLEDGMENTS

The authors acknowledge with gratitude the suggestions of the referees which helped in revising the paper.

REFERENCES

1. A. L. Goel and K. Okumoto, Time Dependent Error Detection Rate Model for Software Reliability and Other Performance Measures, *I.E.E.E. Trans. Reliab.*, August 1979, R-*28*, No. 3, pp. 206-211.
2. A. L. Goel, Software Reliability Models: Assumptions, Limitations and Applicability, *I.E.E.E. Trans. Software Eng.*, December 1985, SE-*11*, No. 12, pp. 1411-1423.
3. P. N. Misra, Software Reliability Analysis, *I.B.M. Systems J.*, 1983, *22*, No. 3, pp. 262-270.
4. M. Ohba, Software Reliability Analysis Models, *I.B.M. J. Res. Dev.*, 1984, *28*, (4) pp. 428-443.
5. K. Okumoto and A. L. Goel, Optimum Release Time for Software Systems Based on Reliability and Cost Criteria, *J. of Systems and Software*, 1980, *1*, pp. 315-318.

6. S. OSAKI and Y. HATOYAMA Eds., Stochastic Models in Reliability Theory, Proc. Symposium, Nagoya, Japan, *Springer-Verlag*, 1984.
7. J. G. SHANTHIKUMAR, Software Reliability Models: a Review', *Micro-electron Reliab.*, 1983, *23*, (5), pp. 903-943.
8. S. YAMADA and S. OSAKI, Reliability Growth Models for Hardware and Software Systems Based on Non-Homogeneous Poisson Process: a Survey', *Micro-electron Reliab.*, 1983, *23*, pp. 91-112.
9. S. YAMADA and S. OSAKI, Software Reliability Growth Modelling: Models and Applications, *I.E.E.E. Trans. Software Eng.*, 1985, SE-*11*, (12), pp. 1431-1437.
10. S. YAMADA, S. OSAKI and H. NARIHISA, A Software Reliability Growth Model with Two Types of Errors, *R.A.I.R.O. Rech. Oper.*, 1985, *19*, (1), pp. 87-104.
11. S. YAMADA, T. YAMANE and S. OSAKI, Software Reliability Growth Models with Error Debugging Rate (in Japanese), *Trans. Information Processing Soc. Japan*, 1986, *27*, (1), pp. 64-71.
12. S. YAMADA, H. NARIHISA and H. OHTERA, Non-Homogeneous Software Error Detection Rate Model: Data Analysis and Applications, *R.A.I.R.O., Rech. Oper.*, 1986, *20*, (1), pp. 51-60.
13. S. YAMADA and S. OSAKI, Optimal Software Release Policies with Simultaneous Cost and Reliability Requirements, *Eur. J. Oper. Res.*, 1987, *31*, pp. 46-51.