Life-time Warranty Cost Model For Software Reliability With Discount Rate

[1]D.R Prince Williams, [2]S. Venkateswaran, and [3]P. Vivekanandan
[1,2,3] Anna University, Chennai 600025, India

Abstract

This paper describes a modified approach to calculate the delivery cost of a software product when lifetime warranty is to be provided. Unlike existing cost models, here the strategy has been to consider maintenance support as long as the software product is operated by the customer. Hence, the optimal release time can be calculated for various reliability levels by minimizing the cost. The vendor can accordingly choose the appropriate combination of cost and reliability levels to arrive at the optimal release time that will be of benefit to him as well as to the customer.

1. Introduction

Professional software development companies are expected to release error free computer software to their customers based on a schedule. But it has become increasingly difficult for the developers to produce highly reliable software products within a specific time interval and cost. Thus, a necessity to control software in terms of reliability, cost and release time becomes critical.

The life cycle (LC) of software development involves a series of specific stages, viz., analysis, design, coding, testing and delivery. The execution of these stages involves a considerable amount of time. Any delay in the release of software may cost a company a certain amount of financial loss. At the same time, the software should reach the customer with a desired level of reliability and with a minimum cost. In reality, the software reaches the customer before all the faults in the software are completely removed. This means the maintenance cost during the operational phase increases. Hence it becomes important that we have a trade-off between delivery schedule and reliability. In other words, it is important to determine the optimal length of software testing subject to reliability and cost. Such an optimization need is called an optimal software release problem.

Many researchers have studied **[1, 2, 3, 8, 10, 15,17]**. this optimization/decision problem and presented different policies. Okumoto et al **[9]** studied the optimal release problem with simulation cost and reliability requirement. Yamada et al **[19]** implemented the optimal release problem with warranty cost and reliability requirement. Yang et al **[20]** studied operational and testing reliability in software assessment models. . Tal et al **[14]** provided an optimal statistical testing policy for software reliability. Jain et al **[6]** investigated hybrid warranty policy, i.e., the cost prediction of the total warranty reserve by including the effects of the time value of money. Trivedi.K.S. et al **[11]** studied Architecture-based approach to reliability assessment of software systems. Yamada et al **[18]** implemented the optimal software release problems with life-cycle distribution and discount rate. In this paper, we discuss optimal release policies that will help a software vendor calculate the software release cost by taking into account the following:

- The software development cost with discount rate
- The software maintenance cost during the operational phase with discount rate
- Expected reliability level

The optimal release time is determined by minimizing the total expected software cost and by satisfying the reliability requirement.

NOTATIONS

a Expected number of initial fault content in a software.

b Error detection rate per one fault

C_0 Initial testing cost which is the barest minimum requirement.

C_t Testing cost per unit time.

C_w Maintenance cost per one fault during the warranty period.

T Software release time

T^* Optimal software release time.

$m(t)$ Expected number of observed failures in the time interval $[0,t)$

$\lambda(t)$ Instantaneous failure intensity at time t

R_0 Desired level of reliability

α Discount Rate

2. The Model

During the software testing stage various test cases are applied to the software product successively. Each test case result is validated to ensure that it meets the user specifications. If the result is unacceptable, the software has at least one software fault, which must be removed. In such a black-box testing process the software failure-occurrence and fault-detection phenomena can be treated as a stochastic process. This phenomenon can be described by defining a suitable mathematical model and are called software reliability models [8] [10] [12].

In our model, NHPP is used to describe the time-dependent nature of the cumulative number of faults detected up to a specific testing time. If $N(t)$ and $m(t)$ be the cumulative number of faults detected up to testing time t and the expectation of $N(t)$, respectively, then the software fault detection phenomenon can be represented as

$$pr\{N(t)=n\} = \frac{\{m(t)\}^n}{n!}\exp[-m(t)]$$

$$(n=0,1,2,\ldots), \tag{1}$$

$$\lambda(t) = \int_0^t m(x)dx \tag{2}$$

In Equations (1) and (2), $m(t)$ is called the mean value function, which represents the expected cumulative number of faults detected in the testing time interval (0,t). $\lambda(t)$ is the failure intensity function which denotes the fault detection rate per unit time.

If we consider the exponential software reliability growth model [4], the mean value function and instantaneous failure rate of the same is given by

$$m(t) = a(1-e^{-bt})\ (a>0, b>0), \tag{3}$$

$$\lambda(t) = abe^{-bt}, \tag{4}$$

3. Warranty Cost Model

When the cost of software development is estimated, software vendors will also have to consider the cost for after-sales support. This is known as the Warranty cost. The computation of this warranty cost is dependant on the release time of the software.

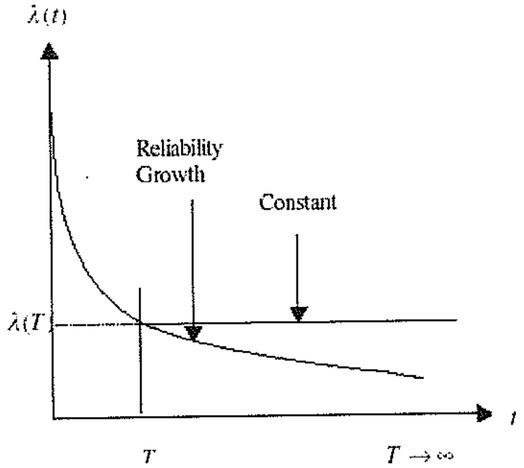

Figure 1. Software Reliability Growth for infinite warranty period.

In this paper, we focus on the optimal release time problem for Life Long Warranty (LLW) with discount rate. The total expected software product cost including maintenance cost can be formulated as

$$LLWC(T) = C_0 + C_t \int_0^T e^{-\alpha t} dt + C_w(T) \quad (5)$$

where $C_w(T)$ is the maintenance cost during the warranty period.

There are two cases in terms of the behavior of $C_w(T)$ (see fig 1).

Case (1) : Let us assume that during the period the software reliability growth does not occur after the testing phase. Then $C_w(T)$ is defined as

$$C_w(T) = C_w \int_T^\infty \lambda(T) e^{-\alpha t} dt \quad (6)$$

Case (2): Let us assume that during period the software reliability growth occurs even after the testing phase. Then $C_w(T)$ is defined as

$$C_w(T) = C_w \int_T^\infty \lambda(t) e^{-\alpha t} dt \quad (7)$$

4. Warranty Cost Model with Reliability Constraint

The software reliability of the NHPP model **[7,11,12,13,16]** is defined as the probability that a software failure does not occur during the testing time and T is mathematically stated as

$$R(x/T) = \exp[-\{m(T+x) - m(T)\}] \quad (8)$$

Substituting Equation (4) in Equation (8), we get

$$R(x/T) = \exp[-e^{-bt} m(x)] \quad (9)$$

Let R_0 be the desired level of reliability. Then the optimal release problem is formulated as

$$\left.\begin{array}{l} \text{Minimize } LLWC(T) \\ \quad \text{Subject to } R(x/T) \geq R_0 \end{array}\right\} \quad (10)$$

4.1 Optimal Software Release Policies – Cost Minimization

Having defined the problem, we now derive the optimal release time policies by minimizing the total expected software cost $LLWC(T)$

Optimal Release Policy 1

Based on $Case(1)$ the total expected software product cost is given by

$$LLWC(T) = C_0 + C_t \int_0^T e^{-\alpha t} dt + C_w \int_T^\infty \lambda(T) e^{-\alpha t} dt \quad (11)$$

Replacing $\lambda(T)$ *with* abe^{-bT} in Equation (11), we get

$$LLWC(T) = C_0 + C_t\left(\frac{1-e^{-\alpha t}}{\alpha}\right) + \frac{C_w ab\, e^{-(\alpha+b)T}}{\alpha} \tag{12}$$

Differentiating Equation (12) with respect to T and equating to zero,

$$T = T_1 = \frac{1}{b}\ln\left(\frac{ab(\alpha+b)c_w}{\alpha c_t}\right) \tag{13}$$

The second derivative of T is greater than zero, that is

$$\frac{d^2\left(LLWC(T)\right)}{dT_1^2} > 0$$

Hence $LLWC(T)$ has a minimum value. The **Optimum Release Policy 1** can now be stated as

P1.1 $T^* = T_1$ when $\lambda(0) > \lambda(T_1)$ (14)

P1.2 $T^* = 0$ when $\lambda(0) \le \lambda(T_1)$ (15)

Optimal Release Policy 2

Based on *Case*(2) the total expected software maintenance cost is given by

$$LLWC(T) = C_0 + C_t\int_0^T e^{-\alpha t}dt + C_w\int_T^\infty \lambda(t)e^{-\alpha t}dt \tag{16}$$

where $\lambda(t) = abe^{-bt}$

$$LLWC(T) = C_0 + C_t\left(\frac{1-e^{-\alpha t}}{\alpha}\right) + \frac{C_w ab\, e^{-(\alpha+b)T}}{(\alpha+b)} \tag{17}$$

Differentiating Equation (17) with respect to T and equating to zero, we get

$$T = T_2 = \frac{1}{b}\ln\left(\frac{abc_w}{c_t}\right) \tag{18}$$

Again the second derivative of T is greater than zero, that is $\frac{d^2\left(LLWC(T)\right)}{dT_1^2} > 0$.

Hence $LLWC(T)$ has a minimum value. The **Optimum Release Policy 2** in this case, can be stated as:

P2.1: $T^* = T_2$ when $\lambda(0) > \lambda(T_2)$
(19)

P2.2 $T^* = 0$ when $\lambda(0) \le \lambda(T_2)$
(20)

4.2 Optimal Software Release Policies – Cost and Reliability Requirement

These policies are derived by considering both the minimization of software cost as well as the required level of reliability.

Let T_R be the optimum release time with respect to T satisfying the relation $R(x/T) = R_0$. By applying the relation $R(x/T) = R_0$ in Equation (8), we obtain T_R as

$$T_R = \frac{1}{b}\left\{\ln m(x) - \ln\ln\left(\frac{1}{R_0}\right)\right\} \tag{21}$$

Having obtained T_R, the optimal release policies including minimization of the total expected software cost can be derived for the two cases identified earlier

Case 1 : Optimum Release Policy 3

P3.1 If $\lambda(0) > \lambda(T_1)$ and $R(X/0) < R_0$,

then $T^* = \max\{T_1, T_R\}$ (22)

P3.2 If $\lambda(0) > \lambda(T_1)$ and $R(X/0) \geq R_0$,

then $T^* = T_1$ (23)

P3.3 If $\lambda(0) \leq \lambda(T_1)$ and ,

then $T^* = T_R$ (24)

P3.4 If $\lambda(0) \leq \lambda(T_1)$ and $R(X/0) \geq R_0$,

then $T^* = 0$ (25)

For Case 2 : Optimum Release Policy 4

P4.1 If $\lambda(0) > \lambda(T_2)$ and $R(X/0) < R_0$,

then $T^* = \max\{T_2, T_R\}$ (26)

P4.2 If $\lambda(0) > \lambda(T_2)$ and $R(X/0) \geq R_0$, then

$T^* = T_2$ (27)

P4.3 If $\lambda(0) \leq \lambda(T_2)$ and $R(X/0) < R_0$,

then $T^* = T_R$ (28)

P4.4 If $\lambda(0) \leq \lambda(T_2)$ and $R(X/0) \geq R_0$,

then $T^* = 0$ (29)

5. Numerical Illustrations

We assume the parameter values for the model as

$C_0 = 1000, a = 1000, b = 0.5$ and $\alpha = 0.2$

The optimal release times are then calculated for different values of C_t and C_w for the respective two policies defined earlier. These have been tabulated and shown in Table 1 and Table 2.

w / C_t	5	10	20	40	50	100
10	28.1	29.5	30.9	32.2	32.7	34.1
20	26.7	28.1	29.5	30.9	31.3	32.7
30	25.9	27.3	28.7	30.0	30.5	31.9
40	25.3	26.7	28.1	29.5	30.5	31.3
50	24.9	26.2	27.6	29.0	29.5	31.3
100	23.5	24.9	26.2	27.6	28.1	29.5

Table 1

Optimal Release Policy 2:

cw / C_t	5	10	20	40	50	100
10	64.4	78.2	92.1	106.0	110.4	124.3
20	50.5	64.4	78.2	92.1	96.6	110.4
30	42.4	56.3	70.1	84.0	88.5	102.3
40	36.7	50.5	64.4	78.2	82.7	96.6
50	32.2	46.1	59.1	73.8	78.2	92.1
100	18.3	32.2	46.1	59.9	64.4	78.2

Table 2

From the above tables it can be seen that when the unit testing cost and warranty cost increases, the optimal release time decreases.

Comparing Case 1 (refer table 1) and Case 2 (refer table2) the optimal release time of Case 2 is always greater than Case 1. This is because the reliability level is also factored into calculating the optimal release time.

Next, we also calculate the R_0 and x, assuming $a = 1000$ and b = 0.5, and tabulate the results as follows:

R_0 \ X	1	2	5	10	50
0.9	122.7	143.7	153.0	164.5	181.4
0.8	107.7	128.7	137.9	149.5	166.4
0.7	98.4	119.3	129.0	140.1	157.1
0.6	91.2	112.2	121.4	132.9	149.1
0.5	85.0	106.1	115.3	126.8	143.8
0.4	79.5	100.5	109.7	121.2	138.2
0.3	74.0	95.0	104.3	115.8	132.7
0.2	68.2	89.2	98.5	109.9	126.9
0.1	61.1	82.1	91.3	102.8	119.8

Table 3

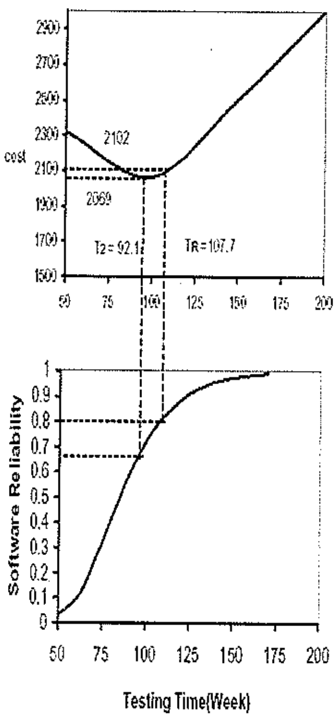

Figure 2. Optimal release time problem

Table 3 shows the optimal release time satisfying Equation (21) for various values of software reliability requirement R_0 and the operational period x. The figures in the table indicate that the optimal release time increases for longer operational period. Figure 2 illustrates the optimal release time of Case 2 with reliability requirement R_0 for Optimal Release Policy 3.

In the case of $C_w = 20$, $C_t = 10$, we derived $T_2 = 92.1$ (see Table 2). We have also obtained $T_R = 107.7$ (see Table 3) for the values of $R_0 = 0.8$ and $x = 1.0$. Therefore, using Optimal Release Policy 4, the Figure 2 shows the optimal release time as follows:

$$T^* = \max\{T_2, T_R\} = \max\{92.1, 107.7\} = 107.7$$

6. Conclusion

In this paper, we have developed a suitable cost model that would help a software vendor to calculate the total software product cost for providing life-time warranty. Also, this model accommodates the following two situations:

- Warranty is provided to only retain the reliability level promised at the time of software release
- Warranty is provided to increase the reliability level from that stated at the time of software release

The numerical example for optimal policy 1 indicates that the optimal release time remains a constant whenever the total Unit Cost is equal to the total warranty Cost. The numerical example for optimal policy 2 indicates that the warranty cost decreases with increase in the operational period. Also, a comparison between the results shown in **Tables 1 and 2** indicate that the rate of decrease in release time as the warranty period increases for optimal release policy 1 is less than that for optimal release policy 2.

References

[1]. Chaar JK. Halliday MJ. Bhandari TS. Chillarege R..In-process evaluation for software inspection and test. IEEE Trans. Softw. Engg 1993;19(11):1055-1070.

[2]. Cho BC, Park KS. An optimal time for software testing under the user's requirement of failure-free demonstration before release. IEICE Trans. Fundamentals 1994;E77-A(3):563-570.

[3]. Foreman EH. Singpurwalla ND. Optimal time intervals for testing hypotheses on computer software errors. IEEE Trans. Reliability 1979;R-28(3):250-253.

[4]. Kimura M, Toyota T and Yamada S.: Economic analysis of software release problems with warranty cost and reliability requirement. Reliability Engineering and System Safety, Vol 66, 1999 pp., 49-55.

[5]. Kimura M. Yamada S. Optimal software policies with random life-cycle and delivery delay. In: Proe. 2nd ISSAT Int. Conf. Reliability and Quality in Design, 195, pp. 215-219.

[6]. Jain M. and Handa B.R.: Cost analysis for repairable units under hybrid warranty. Recent developments in Operational Research, (ed. Manju L:ata Agarwal & Kanwar Sen), Narosa Publishing House, New Delhi, 2001, pp., 149-165ouse House.

[7]. Malaiya YK. Srimani PK. Editors. Software reliability models: theoretical developments. Evaluation and applications Los Alamitos: IEEE computer society press. 1990.

[8]. Musa JD. Tannino A. Okumoto K. Software reliability: measurement, prediction, and application. New York: McGraw-Hill. 1987.

[9]. Okumoto K. Goel AL. Optimum release time for software system based on reliability and cost criteria. J. System and software 1980;j(14):315-318.

[10]. Pham H. editor. Software reliability and testing Los Alamitos: IEEE Computer Society Press. 1990.

[11]. Popstojanova K.G. and Trivedi K.S.: Architecture-based approach to reliability assessment of software systems, Performance Evaluation Vol 45, 2001, pp., 179-204.

[12]. Rallis N.E. and Lansdowne Z.F.: Reliability estimation for a software system with sequential independent reviews. IEEE Transactions on Software Engineering, Vol 27, No 12, December 2001, pp., 1057-1061.

[13]. Shooman ML. Software engineering: design, reliability, and management. New York: McGraw-Hill, 1983.

[14]. Tal. U. Mc Collin C. and Ben dell A.: An optimal statistical testing policy for software reliability. Demonstration of safety critical systems, Vol 137, No.3, March 16, 2002,p.p., 544-557.

[15]. Xie M. Software reliability modeling. Singapore: World scientific, 1991.

[16]. Yamada S. Software quality/reliability measurement and assessment: software reliability growth models and data analysis. J. Info. Process. 1991;14(3):254-266.

[17]. Yamada S. Optimal release problems with warranty period based on a software maintenance cost model. Trans. IPS Japan 1994;35(10):2197-2202.

[18]. Yamada S. Kimura M. Terane E. Osaki S. Optimal software release problems with life-cycle distribution and discount rate. Trans. IPS Japan 1993;34(5):1188-1197 in Japanese.

[19]. Yamada S. Osaki S. Optimal software release policies with simultaneous cost and reliability requirements. Eur. J. Oper. Res. 1987;31(1):46-51.

[20]. Yang B. and Xie M.: A study of operational and testing reliability in software reliability analysis. Reliability Engineering and System Safety Vol 70, 2000, pp., 323-329.