The Korean Communications
in Statistics Vol. 12 No. 3, 2005
pp. 673-682

Optimal Software Release Policy for Random Cost Model

Hee Soo Kim[1], Mi Young Shin[2], and Dong Ho Park[3]

Abstract

In this paper, we generalize the software reliability growth model by assuming that the testing cost and maintenance cost are random and adopt the Bayesian approach to determine the optimal software release time. Numerical examples are provided to illustrate the Bayesian method for certain parametric models.

Keywords : Software reliability growth model, Optimal release policy, Prior distribution

1. Introduction

As the computer system becomes more complex and multiple-function oriented, the developers of software systems are required to produce more reliable software products to prevent the computer systems from stopping its operations during the mission period. Since most of the computer system failures are often caused by the software system failures, rather than by the hardware systems, it is of great importance to apply the best possible software development policy to produce the most reliable software system.

To achieve such a goal, it is a general practice for the software developers to carry out the preliminary testing during a certain length of time before the software system is released to the user for the operational use. During such a testing phase, an effort is given to detect and debug the faults latent in the software system so that the possibility of software failures can be mitigated during the operational phase. Although the number of faults detected and debugged during the testing phase is proportionally increased as the length of software testing period becomes longer, other factors including the testing cost, delivery schedule and the competition with other developers, must be taken into account to determine the optimal length of testing phase as well.

Thus, it is of great importance to determine the best possible software release policy with

1) Research Professor, School of Aerospace and Naval Architecture, Chosun University, Gwangju, 501-759, Korea.
E-mail : krhskim@chosun.ac.kr

2) Corresponding Author, Associate Professor, Department of Mathematics, Catholic University of Korea, Puchon, 420-743, Korea. E-mail : shin@catholic.ac.kr

3) Professor, Department of Information and Statistics, Hallym University, Chuncheon, 200-702, Korea.

respect to certain criteria regarding the software operation. The total software cost and software reliability are commonly used as the criteria for optimal software release time from the viewpoint of software developers. These optimal software release problems have been studied by many researchers (Kimura, Toyota and Yamada(1999), Leung(1992), Okumoto and Goel(1980), Pham(1996), Yamada and Osaki(1985)). For example, Pham(1996) develops a cost model with an imperfect debugging and random life cycle as well as a penalty cost that is used to determine the optimal release policies for a software system and Kimura, Toyota and Yamada(1999) discuss several optimal release problems by using the maintenance cost model and consider the concept of present value into the cost factors and warranty period in the operational phase.

In this paper, we extend the Kimura, Toyota and Yamada's(1999) software reliability growth model by assuming that the testing cost and maintenance cost are random and adopt the Bayesian approach to propose an optimal software release policy with respect to the total software cost incurred until the end of warranty period. Although most of the optimal software release problems have been discussed within the non-Bayesian framework, there exist a few Bayesian discussions in the literature. Mazzuchi and Soyer(1988) discuss the Bayes and empirical-Bayes software reliability models and fit actual software failure data to compare their predictive performance. Recently, McDaid and Wilson(2001) and Morail and Soyer(2003) propose the single stage and multistage testing models to discuss the theoretical approach to derive the optimal software release time.

In Section 2, we present the software reliability growth model proposed by Kimura, Toyota and Yamada(1999), which is based on non-homogeneous Poisson Process(NHPP). Section 3 discusses the Bayesian method by assigning the prior distributions for the testing cost and maintenance cost characterizing the total software cost model and obtain the Bayesian optimal software release policy by minimizing the expected total software cost. The mathematical formula to calculate the expected total software cost is derived as well. Section 4 provides numerical examples to illustrate the proposed optimal software release policy and discusses the effects of several parameters on the release time and its corresponding expected total software cost. Concluding remarks is given in Section 5.

2. Software Reliability Growth Model

It is quite desirable that whenever the debugging activity is performed to correct the software faults detected, the faults are perfectly debugged and are eliminated from the software system without introducing any new faults into the system and thereby, the total number of faults latent in the software system is reduced and consequently the software reliability improves. In this section, we present a software reliability model based on an NHPP, which is proposed by Kimura, Toyota and Yamada(1999), for the purpose of applying the Bayesian approach in the next section.

Let $N(t)$ denote the cumulative number of faults detected up to time t and let $h(t)$ denote the intensity function of an NHPP which describes the software fault detection phenomenon as follows.

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

$$,m(t) = E[N(t)] = \int_0^t h(\tau)d\tau$$

where $m(t)$ represents the expected number of faults detected during (0,t] which is called a mean value function.

As for the intensity function of an NHPP which represents the fault detection rate per unit time, we apply an exponential software reliability growth model discussed in Goel and Okumoto(1979), which is based on the following intensity function, and discuss the optimal software release time.

$$h(t) = \alpha\beta e^{-\beta t},\quad \alpha>0,\quad \beta>0 \tag{2.1}$$

where the parameters α and β are the expected number of initial faults latent in the software and the fault detection rate per fault, respectively.

To illustrate the Kimura, Toyota and Yamada's(1999) software reliability growth model incorporating the testing cost during the testing phase and the maintenance cost during the operational phase by the user, we first introduce the following notations, which are used throughout this paper.

Notations.

C_0	initial setup cost required to initiate the test.
C_t	test cost per unit time
C_w	maintenance cost per fault during the warranty period
T	software release time
T^*	optimal software release time
T_w	length of warranty period
r	discount rate of cost

Based on the intensity function given in (2.1), Kimura, Toyota and Yamada(1999) discuss various types of software reliability growth models with consideration of the software cost incurred during the testing and operational phase. Among them, we consider the following two models to determine the optimal software release time by applying the Bayesian approach on the assumption that the testing cost and the maintenance cost are random.

[Case 1] When the length of warranty period is constant and the software reliability growth does not occur after the testing phase, the total software cost is represented as

$$WC_1(T) = C_0 + C_t\int_0^T e^{-rt}dt + C_w\int_T^{T+T_w} h(T)e^{-rt}dt. \tag{2.2}$$

[Case 2] When the length of warranty period is constant and the software reliability growth is assumed to occur exponentially after the testing phase, the total software cost can be expressed as

$$WC_2(T) = C_0 + C_t\int_0^T e^{-rt}dt + C_w\int_T^{T+T_w} h(t)e^{-rt}dt. \tag{2.3}$$

For the non-Bayesian solution for the optimal software release time, the testing and maintenance cost are assumed to be fixed. However, it is quite likely that these costs are unknown in most of practical situations. Actually, it is more realistic to assume that more restrictive informations such as the upper and lower bound of these costs or the tendency of spending of these costs is given. In the next section, we discuss the optimal software release policy in the context of Bayesian concepts by considering the random cost model.

3. Bayesian Method For Optimal Software Release Policy

To apply the Bayesian method to propose the optimal software release policy, we assume that the testing cost and maintenance cost, C_t and C_w, are random and the initial setup cost, C_0, is a known constant. By considering two costs C_t and C_w as random variables with certain prior distributions, the expected total software cost can be obtained by taking the expectations on the total software cost with respect to C_t and C_w.

3.1. When The Reliability Growth Does Not Occur After Testing Phase

In this case, the expected total software cost is obtained by taking the expectation on $WC_1(T)$ given in (2.2), with respect to C_t and C_w, as follows.

$$C_1(T) = E_{C_t,C_w}WC_1(T) = C_0 + E(C_t)\int_0^T e^{-rt}dt + E(C_w)\int_T^{T+T_w} h(T)e^{-rt}dt. \tag{3.1}$$

Differentiating the equation (3.1) with respect to T and set it equal to 0, we have the equation

$$h(T) = \frac{E(C_t)}{E(C_w)((\beta+r)/r)(1-e^{-rT_w})} \tag{3.2}$$

where $h(t) = \alpha\beta e^{-\beta t}$ is given as in (2.1). Solving the equation (3.1) for T, the solution T_1 can be expressed as

$$T_1 = \frac{1}{\beta}\ln\left[\frac{\alpha\beta E(C_w)((\beta+r)/r)(1-e^{-rT_w})}{E(C_t)}\right]. \tag{3.3}$$

Since $h(T)$ is a decreasing function of T for $T \geqq 0$, it can easily be checked that there exists a finite unique solution T_1 given in (3.3) if $h(0) > E(C_t)/\{E(C_w)((\beta+r)/r)(1-e^{-rT_w})\}$.
In addition, straightforward calculations show that $d^2 C_1(T)/dT^2 \mid_{T=T_1} > 0$ and thus $C_1(T)$ achieves its minimum at $T^* = T_1$.

If $h(0) \leqq E(C_t)/\{E(C_w)((\beta+r)/r)(1-e^{-rT_w})\}$ then $C_1(T)$ is a monotonically increasing function of T. Thus, it is clear that $C_1(T)$ is minimized at $T=0$ and thus $T^* = 0$ in this case.

Consequently, when the length of warranty period is constant and the software reliability growth is not assumed to occur after the testing phase, the optimal software release policy can be stated as follows :

If $h(0) > E(C_t)/\{E(C_w)((\beta+r)/r)(1-e^{-rT_w})\}$, the optimal release time is $T^* = T_1$.
Otherwise, $T^* = 0$.

3.2. When The Reliability Growth Occurs After Testing Phase

When the software reliability grows exponentially during the warranty period, the software reliability improves whenever the fault is detected and debugged while in operation by the user during the warranty period and the maintenance cost incurs to the software developer. In this case, the expected total software cost is obtained by taking the expectation on $WC_2(T)$ given in (2.3), with respect to C_t and C_w as follows.

$$C_2(T) = E_{C_t, C_w} WC_2(T) = C_0 + E(C_t)\int_0^T e^{-rt}dt + E(C_w)\int_T^{T+T_w} h(t)e^{-rt}dt. \tag{3.4}$$

Differentiating the equation (3.4) with respect to T and set it equal to 0, we obtain the following relation.

$$h(T) = \frac{E(C_t)}{E(C_w)(1-e^{-(\beta+r)T_w})}. \tag{3.5}$$

Solving the equation (3.5) for T, the solution T_2 can be written as

$$T_2 = \frac{1}{\beta}\ln\left[\frac{\alpha\beta E(C_w)(1-e^{-(\beta+r)T_w})}{E(C_t)}\right]. \tag{3.6}$$

By using the similar arguments as in the previous case discussed in Section 3.1, $C_2(T)$

achieves its minimum at $T^* = T_2$ if $h(0) > E(C_t)/E(C_w)(1-e^{-(\beta+r)T_w})$. Otherwise, $C_2(T)$ is a monotonically increasing function of T and thus, its minimum is achieved at $T=0$ and thus T^* is taken to be equal to 0.

Therefore, when the length of warranty period is constant and the software reliability growth is assumed to occur after the testing phase, the optimal software release policy is stated as follows :

If $h(0) > E(C_t)/\{E(C_w)(1-e^{-(\beta+r)T_w})\}$ the optimal release time is $T^* = T_2$.
Otherwise, $T^* = 0$.

4. Numerical Examples

To exemplify the Bayesian method discussed in Section 3 numerically, we first assign the prior distributions for both costs C_t and C_w. As for the prior distribution for the testing cost C_t, we take a truncated normal distribution, $TN(\mu_c, \sigma_c^2, c, d)$, which has the following probability density function and the expectation :

$$f(c_t) = \frac{\frac{1}{\sqrt{2\pi}\sigma_c} e^{-(c_t-\mu_c)^2/2\sigma_c^2}}{\Phi\left(\frac{d-\mu_c}{\sigma_c}\right) - \Phi\left(\frac{c-\mu_c}{\sigma_c}\right)} I_{[c,d]}(c_t) \ ,$$

$$E(C_t) = \mu_c + \left[\frac{\phi\left(\frac{c-\mu_c}{\sigma_c}\right) - \phi\left(\frac{d-\mu_c}{\sigma_c}\right)}{\Phi\left(\frac{d-\mu_c}{\sigma_c}\right) - \Phi\left(\frac{c-\mu_c}{\sigma_c}\right)}\right]\sigma_c \ ,$$

where, $\mu_c > 0$, $\sigma_c > 0$, $c \leqq c_t \leqq d$, ϕ and Φ are the probability density function and cumulative distribution function of standard normal , respectively.

As for the prior distribution for C_w, which is the maintenance cost per fault during the warranty period, we assign a discrete type of distribution based on a discrete beta density on the interval (c_L, c_U). It is widely known (Juang and Anderson, 2004) that the discrete beta distribution allows for great flexibility in representing the prior uncertainty regarding some parameters. The beta density defined on the interval (c_L, c_U) is given as

$$g(u) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \frac{(u-c_L)^{a-1}(c_U-u)^{b-1}}{(c_U-c_L)^{a+b-1}} \ , \ 0 \leqq c_L \leqq u \leqq c_U,$$

where $c_L, c_U, a, b > 0$. Applying the beta density $g(u)$, the prior distribution for C_w , denoted by $dBeta(a,b)$ is defined as

$$P_j = \Pr(C_w = c_j) = \int_{c_j - \delta/2}^{c_j + \delta/2} g(u)du,$$

where $c_j = c_L + \delta(2j-1)/2$ and $\delta = (c_U - c_L)/m$ for j=1,2,$\cdots$,m.

To obtain the optimal software release policies based on the Bayesian approach by assuming the truncated normal prior for C_t and the discrete beta prior for C_w, we apply the Bayesian method discussed in Section 3. For numerical calculations of the software release times, we fix $\alpha = 1000$, $\beta = 0.05$, $c_0 = 1000$, $r = 0.001$, and consider several choice of the length of warranty period, $T_w = 5, 10, 20, 50, 100$ with $m = 20$. For the parameters of the prior distributions, we take $c = 250, d = 500$ and several choice of μ_c and σ_c^2 for $TN(\mu_c, \sigma_c^2, c, d)$ and $c_L = 50, c_U = 100$ and several choice of a and b for $dBeta(a, b)$

The numerical results are summarized in Tables 1 and 2, where the first entry shows the optimal release time T^* and the second entry shows the corresponding expected total software cost. Figure 1 illustrates these results when the software reliability growth occurs after the testing phase and when $T_w = 5$, $C_t \sim TN(300, 100^2, 250, 500)$ and $C_w \sim dBeta(a, b)$. For instance, if we take $a = b = 2$, then we obtain the optimal release time T^*=20.6592.

Table 1 shows the results when the software reliability does not occur after the testing phase, while Table 2 lists the case when the software reliability does occur after the software is released to the user. From Tables 1 and 2, we observe the following facts regarding the effects of warranty time, pattern of reliability growth and prior distribution on the optimal release time and its corresponding expected total software cost.

1) Effects of warranty time T_w.

The optimal release time and its corresponding expected total software cost increase as the warranty period gets longer in both cases. Such trend implies that the software developer should take a longer testing period in order to offer a longer warranty period for the user.

2) Effects of whether the software reliability growth occurs during the warranty period.

It is quite clear from Tables 1 and 2 that the optimal release time is always shorter when the software reliability growth occurs than when the software reliability keeps unchanged even after the release of the software system. Thus, the testing period must be longer if the software is subject not to improve after the testing phase, which is as expected.

3) Effects of prior distribution for C_t and C_w

It is interesting to note that the optimal release time gets shorter as the mean value of the testing cost per unit time C_t during the testing phase increases or as the mean value of the maintenance cost per fault C_w during the warranty period gets smaller (that is, when $a < b$). Figure 1 also shows the same trend for the discrete beta prior for the maintenance cost.

Table 1. Optimal release time (the first entry) and the expected total software cost (the second entry) with $C_t \sim TN(\mu_t, \sigma_t^2, 250, 500)$, $C_w \sim dBeta(a,b)$, $50 \leq c_j \leq 100$ when the software reliability growth does not occur.

T_w	$C_t \sim TN(\mu_t,\sigma_t^2,250,500)$					
	$(\mu_t,\sigma_t^2)=(300,50^2)$		$(\mu_t,\sigma_t^2)=(300,100^2)$		$(\mu_t,\sigma_t^2)=(400,50^2)$	
	$(a=2,b=2)$					
5	23.1156	13615.16	23.1051	13618.78	17.3837	15599.49
10	36.9286	17561.46	36.9181	17567.20	31.1967	20885.74
20	50.6918	21439.69	50.6813	21447.51	44.9599	26080.79
50	68.7194	26439.46	68.7088	26449.97	62.9874	32778.21
100	82.0886	30089.50	82.0780	30101.98	76.3566	37667.60
	$(a=2,b=3)$					
5	21.7358	13217.94	21.7252	13221.35	16.0038	15067.40
10	35.5488	17169.69	35.5382	17175.22	29.8168	20360.95
20	49.3120	21053.27	49.3014	21060.89	43.5800	25563.17
50	67.3395	26059.95	67.3290	26070.26	61.6076	32269.84
100	80.7087	29715.03	80.6982	29727.30	74.9768	37165.98
	$(a=3,b=2)$					
5	24.4064	13986.24	24.3958	13990.05	18.6744	16096.57
10	38.2194	17927.45	38.2088	17933.38	32.4874	21375.99
20	51.9826	21800.67	51.9720	21808.69	46.2506	26564.35
50	70.0101	26793.99	69.9996	26804.70	64.2782	23253.12
100	83.3793	30439.33	83.3688	30451.99	77.6474	38136.21

Table 2. Optimal release time (the first entry) and the expected total software cost (the second entry) with $C_t \sim TN(\mu_t, \sigma_t^2, 250, 500)$, $C_w \sim dBeta(a,b)$, $50 \leq c_j \leq 100$ when the software reliability growth occurs.

T_w	$C_t \sim TN(\mu_t,\sigma_t^2,250,500)$					
	$(\mu_t,\sigma_t^2)=(300,50^2)$		$(\mu_t,\sigma_t^2)=(300,100^2)$		$(\mu_t,\sigma_t^2)=(400,50^2)$	
	$(a=2,b=2)$					
5	20.6697	12910.69	20.6592	12913.93	14.9378	14655.83
10	32.1448	16200.92	32.1343	16205.93	26.4129	19063.23
20	41.5511	18869.95	41.5405	18876.39	35.8191	22638.51
50	48.8694	20929.24	48.8589	20936.79	43.1375	25397.02
100	50.3731	21350.49	50.3626	21358.26	44.6412	25961.30
	$(a=2,b=3)$					
5	19.2899	12512.50	19.2794	12515.53	13.5579	14122.43
10	30.7650	15807.27	30.7544	15812.07	25.0330	18535.93
20	40.1712	18479.99	40.1607	18486.22	34.4393	22116.14
50	47.4896	20542.12	47.4791	20549.46	41.7577	24878.46
100	48.9933	20963.95	48.9827	20971.52	43.2613	25443.52
	$(a=3,b=2)$					
5	21.9605	13282.68	21.9500	13286.11	16.2286	15154.12
10	33.4356	16568.66	33.4250	16573.87	27.7036	19555.84
20	42.8418	19234.25	42.8313	19240.88	37.1099	23126.51
50	50.1602	21290.88	50.1497	21298.62	44.4283	25881.46
100	51.6639	21711.59	51.6533	21719.56	45.9319	26445.01

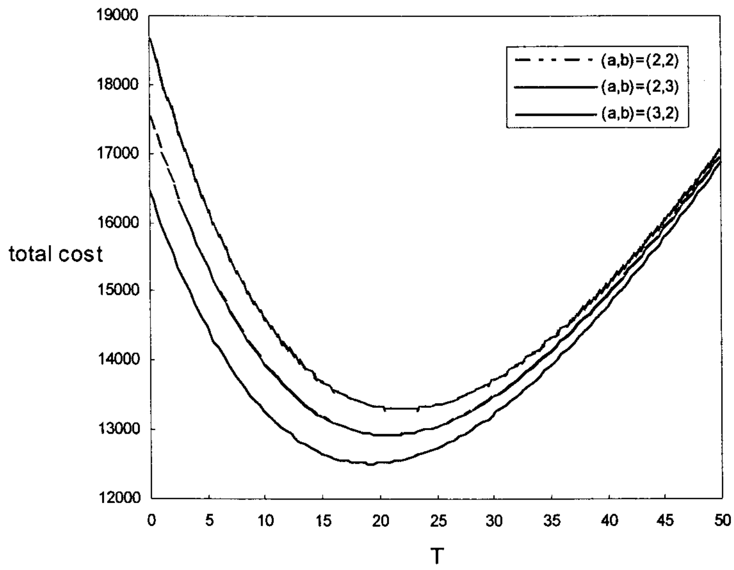

Figure 1. Expected total software cost $C_2(T)$ when $T_w = 5$ and $C_t \sim TN(300, 100^2, 250, 500)$

5. Concluding Remarks

Most of the software reliability growth models have been studied in the context of non-Bayesian approach by many researchers. In such approach, the parameters characterizing the model are either known constants or fixed, but unknown. However, it is more realistic that somewhat restrictive informations such as the upper and lower bound or range regarding certain parameters is available. In such cases, it is widely recognized that the Bayesian method could be quite effective relative to the non-Bayesian method.

This paper generalizes the Kimura, Toyota and Yamada's(1999) software reliability growth models by assuming that the testing cost and the maintenance cost are random variables. As the prior distributions for such costs, we consider the truncated normal distribution and the discrete beta distribution, which are known to have great flexibility in representing the prior uncertainty regarding some parameters. Based on such prior distributions, the mathematical expressions to formulate the expected total software cost are derived and the Bayesian optimal software release policy is proposed by minimizing the expected total software cost. The sensitivity analysis with respect to length of warranty time, pattern of reliability growth and prior distribution is conducted numerically and the effects of each parameter on the optimal release time is studied in details. Consequently, as the testing cost gets lower and the maintenance cost becomes more expensive, the length of testing phase needs to be greater in order to minimize the expected total software cost.

References

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

[2] Juang, M. G. and Anderson, G.(2004), A Bayesian method on adaptive preventive maintenance problem, *European Journal of Operational Research*, vol. 155, pp.455-473.

[3] Kimura, M., Toyota, T. and Yamada, S.(1999), Economic analysis of software release problems with warranty cost and reliability requirement, *Reliability Engineering & System Safety*, vol. 66, pp.49-55.

[4] Leung, Y. W.(1992), Optimum software release time with a given cost budget, *The Journal of Systems and Software*, vol. 17, pp.233-242.

[5] Mazzuchi, T. A. and Soyer, R.(1988), A Bayes empirical-bayes model for software reliability, *IEEE Transactions on Reliability*, vol.37, pp.248-254.

[6] McDaid, K. and Wilson, S. P.(2001), Deciding how long to test software, *Journal of The Royal Statistical Society, Series D*, vol. 50, pp.117-134.

[7] Morail, N. and Soyer, R.(2003), Optimal stopping in software testing, *Naval Research Logistics*, vol.50, pp .88-104.

[8] Okumoto, K. and Goel, A. L.(1980), Optimum release time for software systems based on reliability and cost criteria, *The Journal of Systems and Software*, vol.1, pp.315-318.

[9] Pham, H.(1996), A Software cost model with imperfect debugging, random life cycle and penalty cost, *International Journal of Systems Science*, vol. 27, pp.455-463.

[10] Yamada, S. and Osaki, S.(1985), Cost reliability optimal release policies for software systems, *IEEE Transactions on Reliability*, vol.34, pp.422-424.

[Received July 2005, Accepted October 2005]