The Optimal Release Time in Cost Model Using PCLS Model

Kwang Yoon Song, In Hong Chang†, Min Su Choi, and Da Hye Lee

Abstract

The basic goal of software development is to produce high quality software at low cost. Therefore, when to stop software testing and release the software product is a significant point in the software development. The software cost model is an effective tool used to help software developers control costs and determine the release time. In this paper, we discuss the cost model to apply all 6 models with consideration of time to remove errors, cost of removing each error and risk cost due to software failure. We show the impact of cost coefficients and parameter values on the expected total cost by changing the values and comparing the optimal release times.

Keywords: software reliability, cost model, NHPP, optimal release time

1. Introduction

In the past, software developers are considered a high performance hardware to the important factor. The capabilities of hardware in modern society is reduced, however, the software have been expanded. The software including many features has been more complex than in the past. And also, the most important thing is to develop a high quality software at low cost.

Many software reliability model have been developed for the estimation of software reliability. Among all these proposed models, the non-homogeneous Poisson process (NHPP) software reliability models (SRMs) have been widely and successfully applied in the field of software reliability engineering. Many software reliability models based on NHPP have been proposed[1-7]. NHPP SRMs assume that the cumulative number of faults detected up to testing time $t$ follows an NHPP with a mean value function $m(t)$. Therefore, an NHPP SRM is characterized by its mean value function. Most of the research in these areas has been limited to consideration of either the hardware system alone, or the software system alone. In these systems, the hardware and the software subsystem are not independent of each other, and consequently the system reliability is affected by interactions between them[8]. Chang et al.(2014) and Lee et al.(2016), recently, presented the reliability model with consideration of hardware and software (PCLS model)[9,10].

Moreover, the growth of reliability, and the trade-off between cost expenditure and optimal release both depend on the accuracy of software reliability model established. So, in software reliability engineering, building accurate software reliability model, implementing effectively balance control between software cost reliability and optimal release time are important guarantees for realizing projected objectives[11]. The software before released goes through a long-time testing process in order to achieve a safe and stable reliability level. However, the testing process consumes time and money. At other times, shorter time of the software testing is more conductive to reduce its expense, but at the same time the unreliability of the software is brought forward with the increasing costs of maintenance. On the other hand, longer time of the software testing raises its expense while the delay of the software release time cuts down the product's market share[12,13]. Pham(1996) proposed a model having a penalty cost that affect the optimal release of the policy with incomplete software debugging, random life cycle[14]. Pham and Zhang (1999) developed a cost model having time to remove each error detected in the system software and risk costs due to software failure[15]. Chatterjee and Singh(2014) proposed a software reliability growth model based on a NHPP that incorporates a logistic-exponential testing
coverage function with imperfect debugging and developed a software cost model incorporating testing coverage\textsuperscript{[16]}. Dwivedi and Kumar (2016) discussed a software release time problem using fuzzy optimization under constraints for the objective cost minimization in relation with the software reliability and its testing effort\textsuperscript{[17]}.

In this paper, we show the impact of cost coefficients and parameter values on the expected total cost by changing the values, and compare the optimal release times of the PCLS model and existing well-known NHPP software reliability models. In section 2, NHPP SRM and cost model is presented. In Section 3, numerical examples and results are provided. Finally, in section 4, conclusions are drawn.

### Notation

| Symbol | Description |
|--------|-------------|
| $m(T)$ | Expected number software failures by time $T$ |
| $\lambda(T)$ | Fault detection rate per unit time |
| $R(x|T)$ | Reliability function of software by time $T$ for a mission time $x$ |
| $T$ | Software release time |
| $T_0$ | Maximum time to remove a fault |
| $T^*$ | Optimal software release time |
| $C_1$ | Software test cost per unit time |
| $C_2$ | Cost of removing each fault per unit time during testing |
| $C_3$ | Loss due to software failure |
| $E(T)$ | Expected cost of a software system at time $T$ |
| $Y$ | Variable of time to remove a fault, which is $E(T)$ |
| $x$ | Mission time |

### 2. Software Reliability Models and Cost Model

#### 2.1. NHPP Software Reliability Model

The software fault detection process has been widely formulated by using a counting process. A counting process $\{N(t), \ t \geq 0\}$, is said to be a non-homogeneous Poisson process (NHPP) with intensity function $\lambda(t)$, if

$$P\{N(t) = n\} = \frac{\lambda(t)^n}{n!}e^{-\lambda(t)}$$

$n = 0, 1, 2, ...$

The mean value function $m(t)$, which is the expected cumulative number of faults detected at time $t$ can be expressed as

$$m(t) = \int_0^t \lambda(s)ds$$

where $\lambda(t)$ represents the fault detection rate function, which is the number of faults detected per unit time $t$.

#### Table 1. Software reliability models

| Model                      | $m(t)$                      |
|----------------------------|-----------------------------|
| Goel-Okumoto Model\textsuperscript{[1]} | $m(t) = a(1-e^{-bt})$      |
| Inflection S-shaped Model\textsuperscript{[2]} | $m(t) = a\left(1-e^{-\frac{bt}{b}}\right)$ |
| Yamada Imperfect Debugging Model\textsuperscript{[3]} | $m(t) = a\left[1-e^{-bt}\right] + \frac{aat}{1+bt}$ |
| PNZ Model\textsuperscript{[4]} | $m(t) = \frac{a[1-e^{-bt}]\left[1+\frac{aat}{bt}\right] + \frac{aat}{1+bt}}{1+be^{-bt}}$ |
| Pham-Zhang Model\textsuperscript{[5]} | $m(t) = \frac{(c+a)[1-e^{-bt}]\left[\frac{ab}{b-a}e^{-at} - e^{-bt}\right]}{1+be^{-bt}}$ |
| PCLS Model\textsuperscript{9-10} | $m(t) = \frac{ab(e^{(a-b)t}-1)}{ae^{(a-b)t}-b}$ |
The software reliability $R(x|t)$ is defined as the probability that a software failure does not occur in the time interval $[t, t+x]$ ($t \geq 0, x \geq 0$).

$$R(x|t) = e^{-m(t)+m(t+x)}$$

Table 1 show the mean value of a function $m(t)$ of existing well-known NHPP software reliability models and a reliability model with consideration of hardware and software (PCLS model).

2.2 Cost Model

Zhang and Pham (1998) have discussed a cost model considering error removal times and risk costs [18]. The assumptions of the model are given as follows:

1. The cost to carry out testing is proportional to the testing time.
2. The cost to remove faults during the testing phase is proportional to the total time of removing all faults found in the system by the end of testing phase.
3. There is a risk cost related to the reliability at each release time point.
4. It takes time to remove faults, and we assume that the time to remove each fault follows a truncated exponential distribution.

From assumption (4), the probability density functions of $Y$ is given by

$$f_Y(y) = \frac{\lambda e^{-\lambda y}}{1-e^{-\lambda y}}, \text{ for } 0 \leq y \leq T_0$$

The expected time to remove each fault is given by

$$E(Y) = \int_0^{T_0} f_Y(y) dy = \int_0^{T_0} \frac{\lambda e^{-\lambda y}}{1-e^{-\lambda y}} dy$$

$$= \int_0^{T_0} \frac{\lambda e^{-\lambda y}}{1-e^{-\lambda y}} dy = \frac{1-(\lambda T_0+1)e^{-\lambda T_0}}{\lambda (1-e^{-\lambda T_0})}$$

The expected software system each cost, $C(T) = \sum E_i(T)$

(i) the cost to do testing phase;

$$E_1(T) = C_1T$$

(ii) the cost incurred in removing faults during testing phase;

$$E_2(T) = C_2 \left( \sum_{i=1}^{n(T)} Y_i \right) = C_2 m(T) \mu_y$$

(iii) the risk cost due to software failures;

$$E_3(T) = C_3 (1-R(x|T))$$

Therefore, the expected total software cost $C(T)$ can be expressed as

$$C(T) = E_1(T) + E_2(T) + E_3(T) = C_1 T + C_2 m(T) \mu_y + C_3 (1-R(x|T))$$

We will expressed the expected cost model for the optimal software release time, $T^*$, which minimizes the expected total cost.

3. Numerical Example

3.1. Data Information

The Data set #1 and #2, given in Table 2, was reported by Pham (2006) [19]. The On-line Communication System (OCS) project at ABC Software Company was completed in 2000. The project consisted of one unit-manager, one user interface software engineer, and ten software engineers/testers. The data was collected over a period of 12 weeks during which time the testing started and stopped many times. Table 2 shows cumu-

| Weeks Index | Cumulative Failures |
|-------------|---------------------|
|             | Data set #1 | Data set #2 |
| 1           | 10        | 17        |
| 2           | 12        | 24        |
| 3           | 16        | 28        |
| 4           | 22        | 38        |
| 5           | 28        | 48        |
| 6           | 36        | 71        |
| 7           | 40        | 81        |
| 8           | 43        | 89        |
| 9           | 44        | 91        |
| 10          | 50        | 104       |
| 11          | 51        | 105       |
| 12          | 55        | 110       |

Table 2. OCS data - Data Set #1, #2
The Optimal Release Time in Cost Model Using PCLS Model

Table 3. Model Parameter Estimation from OCS data - Data set #1, #2

| Model | Data set #1 | Estimated parameters | Data set #2 | Estimated parameters |
|-------|-------------|----------------------|-------------|---------------------|
| GO    | \( \hat{a} = 94.344, \hat{b} = 0.07330 \) | \( \hat{a} = 275.182, \hat{b} = 0.045 \) |
| IS    | \( \hat{a} = 65.781, \hat{b} = 0.206, \hat{p} = 1.293 \) | \( \hat{a} = 121.796, \hat{b} = 0.325, \hat{p} = 4.000 \) |
| YID   | \( \hat{a} = 94.344, \hat{b} = 0.07333, \hat{\alpha} = 0 \) | \( \hat{a} = 269.020, \hat{b} = 0.046, \hat{\alpha} = 0.001 \) |
| PNZ   | \( \hat{a} = 64.922, \hat{b} = 0.208, \hat{\alpha} = 0.001, \hat{\beta} = 1.286 \) | \( \hat{a} = 121.909, \hat{b} = 0.324, \hat{\alpha} = 0, \hat{\beta} = 3.977 \) |
| PZ    | \( \hat{a} = 0.001, \hat{b} = 0.206, \hat{\alpha} = 0.001, \hat{\beta} = 1.293, \hat{c} = 65.781 \) | \( \hat{a} = 0.001, \hat{b} = 0.325, \hat{\alpha} = 0.001, \hat{\beta} = 4.000, \hat{c} = 121.795 \) |
| PCLS  | \( \hat{a} = 310.427, \hat{b} = 113.333, \hat{k} = 0.0002 \) | \( \hat{a} = 895.360, \hat{b} = 348.630, \hat{k} = 0.0004 \) |

Table 4. Optimal Release times and Minimum Expected Total Costs of all 6 models from Case #1 (Data set #1, #2)

| Model | Data set #1 | Data set #2 |
|-------|-------------|-------------|
|       | \( C(T) \) | \( T^* \) | \( C(T) \) | \( T^* \) |
| GO    | 1385.7127   | 31.5       | 2840.9516 | 48.9 |
| IS    | 910.0345    | 24.1       | 1121.1575 | 22.5 |
| YID   | 1385.7127   | 31.5       | 2931.0501 | 45.4 |
| PNZ   | 938.9179    | 23.7       | 1122.6932 | 22.5 |
| PZ    | 910.0351    | 24.1       | 1121.1526 | 22.5 |
| PCLS  | 1562.3936   | 28.8       | 3216.1839 | 36.2 |

Table 5. Optimal Release times and Minimum Expected Total Costs of all 6 models from Case #2 (Data set #1)

|       | \( C(T) \) | \( T^* \) | \( C(T) \) | \( T^* \) | \( C(T) \) | \( T^* \) |
|-------|-------------|-------------|-------------|-------------|-------------|-------------|
| GO    | 1174.3946   | 36.0        | 1385.7127   | 31.5        | 1575.6676   | 27.9        |
| IS    | 753.7428    | 25.6        | 910.0345    | 24.1        | 1059.5314   | 22.9        |
| YID   | 1174.3946   | 36.0        | 1385.7127   | 31.5        | 1575.6676   | 27.9        |
| PNZ   | 784.0512    | 25.2        | 938.9179    | 23.7        | 1087.0475   | 22.6        |
| PZ    | 753.7434    | 25.6        | 910.0351    | 24.1        | 1059.5320   | 22.9        |
| PCLS  | 1358.9036   | 34.8        | 1562.3936   | 28.8        | 1736.9087   | 24.3        |
Fig. 1. Optimal Release Times and Minimum Expected Total Costs of Case #2 (Data set #1).

Table 6. Optimal Release times and Minimum Expected Total Costs of all 6 models from Case #2 (Data set #2)

| Model | $C_1 = 15, C_2 = 45$ | $C_1 = 20, C_2 = 50$ | $C_1 = 25, C_2 = 55$ |
|-------|---------------------|---------------------|---------------------|
|       | $C(T)$              | $T^*$               | $C(T)$              | $T^*$               | $C(T)$              | $T^*$               |
| GO    | 2451.3491           | 57.3                | 2840.9516           | 48.9                | 3186.0836           | 41.5                |
| IS    | 945.7833            | 23.4                | 1121.1575           | 22.5                | 1292.4474           | 21.8                |
| YID   | 2559.9318           | 53.4                | 2931.0501           | 45.4                | 3257.4984           | 38.1                |
| PNZ   | 947.0657            | 23.5                | 1122.6932           | 22.5                | 1294.2208           | 21.8                |
| PZ    | 945.7790            | 23.4                | 1121.1526           | 22.5                | 1292.4419           | 21.8                |
| PCLS  | 2880.3416           | 48.9                | 3216.1839           | 36.2                | 3474.0665           | 24.7                |

Fig. 2. Optimal Release Times and Minimum Expected Total Costs of Case #2 (Data set #2).
The Optimal Release Time in Cost Model Using PCLS Model

The release time $T^*$ of PCLS model is 28.8 weeks in Data set #1, and 36.2 weeks in Data set #2.

Cost coefficients $C_1$ and $C_2$ can be interpreted as the weight put on testing cost and error removal cost in the model. In Case #2, we were changing, $C_1=15$, $C_2=45$ and $C_1=25$, $C_2=55$ and compare the optimal release time $T^*$ with that of Case #1. As can be seen from Table 5 and 6, we found out that $T^*$ is increasing when to decrease the value of cost coefficients $C_1$ and $C_2$. On the other hand, $T^*$ is decreasing when to increase the value of cost coefficients $C_1$ and $C_2$.

In Case #3, we were changing, $\mu_y=0.07$ and $\mu_y=0.13$ and compare the optimal release time $T^*$ with that of Case #1. As can be seen from Table 7 and 8, we found out that $T^*$ is increasing when to decrease the value of expected time to remove a fault $\mu_y$. On the other hand,

### Table 7. Optimal Release times and Minimum Expected Total Costs of all 6 models from Case #3 (Data set #1)

| Model | $C(T)$ | $T^*$ | $C(T)$ | $T^*$ | $C(T)$ | $T^*$ |
|-------|--------|-------|--------|-------|--------|-------|
| GO    | 1257.8836 | 32.2  | 1385.7127 | 31.5  | 1512.7528 | 30.7  |
| IS    | 812.9283 | 24.1  | 910.0345 | 24.1  | 1007.1106 | 24.0  |
| YID   | 1257.8836 | 32.2  | 1385.7127 | 31.5  | 1512.7528 | 30.7  |
| PNZ   | 841.2643 | 23.8  | 938.9179 | 23.7  | 1036.5257 | 23.6  |
| PZ    | 812.9289 | 24.1  | 910.0351 | 24.1  | 1007.1112 | 24.0  |
| PCLS  | 1430.2668 | 30.4  | 1562.3936 | 28.8  | 1691.6945 | 27.2  |

### Table 8. Optimal Release times and Minimum Expected Total Costs of all 6 models from Case #3 (Data set #2)

| Model | $C(T)$ | $T^*$ | $C(T)$ | $T^*$ | $C(T)$ | $T^*$ |
|-------|--------|-------|--------|-------|--------|-------|
| GO    | 2471.2882 | 51.5  | 2840.9516 | 48.9  | 3204.7690 | 45.8  |
| IS    | 939.0713 | 22.5  | 1121.1575 | 22.5  | 1303.2437 | 22.5  |
| YID   | 2562.7887 | 48.5  | 2931.0501 | 45.4  | 3290.1815 | 41.6  |
| PNZ   | 940.4397 | 22.6  | 1122.6932 | 22.5  | 1304.9373 | 22.5  |
| PZ    | 939.0678 | 22.5  | 1121.1526 | 22.5  | 1303.2373 | 22.5  |
| PCLS  | 2852.5643 | 43.5  | 3216.1839 | 36.2  | 3538.5595 | 25.7  |

Fig. 3. Optimal Release Times and Minimum Expected Total Costs of Case #3 (Data set #1).
Fig. 4. Optimal Release Times and Minimum Expected Total Costs of Case #3 (Data set #2).

Table 9. Optimal Release times and Minimum Expected Total Costs of all 6 models from Case #4 (Data set #1)

|   | \( x = 0.1 \) | \( x = 0.2 \) | \( x = 0.5 \) |
|---|---|---|---|
| \( C(T) \) | \( T^* \) | \( C(T) \) | \( T^* \) | \( C(T) \) | \( T^* \) |
| GO  | 1385.7127 | 31.5 | 1596.1771 | 42.1 | 1855.7169 | 55.1 |
| IS  | 910.0345  | 24.1 | 979.7618  | 27.6 | 1067.7417 | 32.0 |
| YID | 1385.7127 | 31.5 | 1596.1771 | 42.1 | 1855.7169 | 55.1 |
| PNZ | 938.9179  | 23.7 | 1040.7738 | 27.2 | 1222.7994 | 31.5 |
| PZ  | 910.0351  | 24.1 | 979.7629  | 27.6 | 1067.7442 | 32.0 |
| PCLS| 1562.3936 | 28.8 | 1916.9761 | 43.9 | 2379.5920 | 65.2 |

Fig. 5. Optimal Release Times and Minimum Expected Total Costs of Case #3 (Data set #1).
The Optimal Release Time in Cost Model Using PCLS Model 213

$T^*$ is decreasing when to increase the value of expected time to remove a fault $\mu_y$.

In Case #4, we were changing, $x$=0.2 and $x$=0.5 and compare the optimal release time $T^*$ with that of Case #1. As can be seen from Table 9 and 10, we found out that $T^*$ is increasing when to increase the value of mission time $x$.

4. Conclusions

In this paper, we discuss the cost model to apply all 6 models with consideration of time to remove errors, cost of removing each error during the testing environment, and risk cost due to software failure. As can be seen from Tables, we are shown the impact of cost coefficients and parameter values on the expected total cost and the optimal release time. we found out that optimal release time $T^*$ is increasing when the value of cost coefficients $C_1$, $C_2$ and expected time to remove a fault $\mu_y$ are decrease. On the other hand, $T^*$ is decreasing when the value of cost coefficients $C_1$, $C_2$ and expected time to remove a fault $\mu_y$ are increase. And also, we found out that $T^*$ is increasing when the value of mission time $x$ is increase. In conclusion, the results of the PCLS model and the other NHPP software reliability models can be seen that has a similar shape. Future work in broader validation of this conclusion is needed based on recent data sets.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2013R1A1A2009277) and (2015R1D1A1A01060050).

| Model | $x = 0.1$ | $x = 0.2$ | $x = 0.5$ |
|-------|-----------|-----------|-----------|
| C($T$) | $T^*$      | C($T$)    | $T^*$      | C($T$)    | $T^*$      |
| GO    | 2840.9516 | 48.9      | 3217.2072 | 68.3      | 3657.6676 | 90.6      |
| IS    | 1121.1575 | 22.5      | 1163.9776 | 24.6      | 1218.0993 | 27.4      |
| YID   | 2931.0501 | 45.4      | 3439.2373 | 63.8      | 4226.1939 | 83.6      |
| PNZ   | 1122.6932 | 22.5      | 1165.6428 | 24.7      | 1219.9430 | 27.4      |
| PZ    | 1121.1526 | 22.5      | 1163.9732 | 24.6      | 1218.0964 | 27.4      |
| PCLS  | 3216.1839 | 36.2      | 3953.9300 | 66.3      | 4854.5873 | 100.0     |

Fig. 6. Optimal Release Times and Minimum Expected Total Costs of Case #4 (Data set #2).
References

[1] A. L. Goel and K. Okumoto, “Time-dependent error-detection rate model for software reliability and other performance measures”, IEEE T. Reliab., Vol. R-28, pp. 206-211, 1979.

[2] M. Ohba, “Inflection S-shaped software reliability growth model”, in stochastic models in reliability theory, Ed. S. Osaki, Y. Hatoyama, Berlin: Springer-Verlag, pp. 144-162, 1984.

[3] S. Yamada, K. Tokuno, and S. Osaki, “Imperfect debugging models with fault introduction rate for software reliability assessment”, Int. J. Syst. Sci., Vol. 23, pp. 2241-2252, 1992.

[4] H. Pham and X. Zhang, “An NHPP software reliability models and its comparison”, International Journal of Reliability, Quality and Safety Engineering, Vol. 4, pp. 269-282, 1997.

[5] H. Pham, L. Nordmann, and X. Zhang, “A general imperfect software debugging model with S-shaped fault detection rate”, IEEE T. Reliab., Vol. 48, pp. 169-175, 1999.

[6] J. Gao, X. Bai, W. T. Tsai, and T. Uehara, “Mobile application testing: A tutorial”, Computer, Vol. 47, pp. 46-55. 2014.

[7] F. Khomh, T. Dhaliwal, Z. Ying, and B. Adams, “Do faster releases improve software quality? An empirical case study of Mozilla Firefox”, (paper presented at the Mining Software Repositories(MSR), 2012 9th IEEE Working Conference on, June 2-3, pp. 179-188, 2012.

[8] P. K. Kapur, R. B. Garg, and S. Kumar, “Contributions to hardware and software reliability”, Singapore: World Scientific Publishing Co. Ltd, 1999.

[9] I. H. Chang, H. Pham, S. W. Lee, and K. Y. Song, “A reliability model with hardware and software failures”, Proceedings of 6th Asia-Pacific International Symposium on Advanced Reliability and Maintenance Modeling, pp. 41-48, 2014.

[10] S. W. Lee, I. H. Chang, H. Pham, and K. Y. Song, “A new NHPP reliability model with considerations of hardware and software failures”, Journal of the Korean Data Analysis Society, Vol. 18, pp. 1179-1189, 2016.

[11] C. Zhang, G. Cui, F. Meng, H. Liu, and S. Wu. “A study of optimal release policy for SGRM with imperfect debugging”, Journal of Engineering Science and Technology Review, Vol. 6, pp. 111-118, 2013.

[12] F. Gao, C. Lin, B. Meng, and S. Chen. “A software cost model considering the difference between testing environment and operating environment”, 7th International Joint Conference on Computational Sciences and Optimization, July 4-6, pp. 45-50, 2014.

[13] W. Ehrlich, B. Prasanna, J. Stampfel, and J. Wu. “Determining the cost of a stop testing decision”, IEEE Software, Vol. 10, pp. 33-42, 1993.

[14] H. Pham, “A software cost model with imperfect debugging, random life cycle and penalty cost”, Int. J. Syst. Sci., Vol. 27, pp. 455-463, 1996.

[15] H. Pham and X. Zhang, “Software release policies with gain in reliability justifying the costs”, Annals of Software Engineering, Vol. 8, pp. 147-166, 1999.

[16] S. Chatterjee and J. B. Singh, “A NHPP based software reliability model and optimal release policy with logistic-exponential test coverage under imperfect debugging”, International Journal of System Assurance Engineering and Management, Vol. 5, pp. 399-406, 2014.

[17] A. Dwivedi and D. Kumar, “Optimal release policy of software with imperfect debugging and testing effort under fuzzy environment”, International Journal of Engineering Applied Sciences and Technology, Vol. 1, pp. 103-107, 2016.

[18] X. Zhang and H. Pham, “A software cost model with error removal times and risk costs”, Int. J. Syst. Sci., Vol. 29, pp. 435-442, 1998.

[19] H. Pham, “System software reliability”, London: Springer-Verlag, 2006.