

予定된 引渡時期를 가진 Software 시스템의 最適放出政策 (Optimal Release Policies of Software System with Scheduled Delivery Time)

鄭 永 培 *
申 鉉 宰 **

Abstract

A decision procedure to determine when computer software should be released after testing is described.

This paper extends optimum release policies minimizing the total expected software cost with a scheduled software delivery time under reliability requirement constraint. Such cost considerations enable us to make a release decision as to when transfer a software system from testing phase to operational phase.

The underlying model is software reliability growth model described by a nonhomogeneous poisson process. It is assumed that the penalty cost function due to delay for a scheduled software delivery time is linearly proportional to time. Numerical examples are shown to illustrate the results.

1. 序 論

現在 企業이나 研究機關 등 電算化가 必要한 곳에서 Computer Software의 需要가 급격히 增加하는 추세에 있다. 하지만 需要者가 요구하는 信賴性을 가진 Software를 가장 經濟的인 費用으로 어느 時點에서 放出해야 하는가에 대한 效果的인 政策을 거의 가지고 있지 않은 現實에서, 本 研究는 引渡時期가 미리 정해진 Computer Software의 테스트 단계에서 費用과 信賴度를 함께 고려하여 最適의 放出時期를 決定하는 政策을 다룬다.

從來의 研究에서 Okumoto & Goel은 NHPP (Nonhomogeneous Poisson Process)에 바탕을 둔 Software Reliability Growth Model을 이용하여 Software 放出 問題를 研究했다. 그들은 最適放出時期를 決定하는 데에 2개의 評價基準을 導入했다.

① 總平均 Software testing 費用

② Software Reliability

지금까지 여러 論文들에서 이 두개의 基準들을 각

각 혹은 같이 使用하여 Software의 最適放出政策을 研究하였다.

本 研究에서는 NHPP에 의한 Software Reliability Growth Model을 利用하여 위의 두가지 基準을 同時에 適用하는 것은 물론 이제까지 함께 고려되지 않았던 豫定된 引渡時期를 넘길 경우의 Penalty 費用을 고려하여 보다 現實的인 Model을 構成하여 最適 Software 放出時期를 정하는 政策을 보인다.

2. 假定 및 記號說明

2.1 假定

① 發見된 error는 즉시 除去되며, 除去過程중 또 다른 error가 삽입되지 않는다.

② Counting Process $N(t)$ 는 NHPP에 따른다.

③ Penalty Cost C_p 는 $(T^* - D)$ 에 線型으로 比例한다.

④ Software 總使用期間 T_{LC} 는 $T_{LC} > \max\{T_0, T_1\}$ 이다.

* 漢陽大學校 産業工學科 講師

** 仁川大學 助教授

2 · 2 記號說明

- N : Software 內에 내장된 初期 error 數
- τ : error 1 개당 發見率
- $N(t)$: time t 까지 發見된 error의 累積數
- $m(t)$: $E\{N(t)\}$
- C_a : 테스트期間 동안 發見된 error를 고치는 平均費用 ($C_a > 0$)
- C_b : 放出後 發見된 error를 고치는 平均費用 ($C_b > C_a$)
- C_c : 테스트期間 동안의 平均所要費用
- C_p : 引渡時期를 넘길 경우 單位 Penalty 費用
- $C(T)$: 總平均 Software 費用
- $R(x/t)$: Software 信賴度 즉, time t 에서 가장 最近에 error가 發見되었을 때 ($t, t+x$) 사이에서 어떤 失敗도 일어나지 않을 確率
- T : Software 放出時間 즉, 總테스트時間
- T_0 : $dC(T)/dT = 0$ 를 만족시키는 T 값
- T_1 : $R(x/T) = R_0$ 를 만족시키는 T 값
- T^* : 最適 Software 放出時間
- R_0 : 使用者가 요구한 Software 信賴度
- T_{LC} : Software 總使用期間
- $P(t)$: 引渡時期를 넘길 경우의 Penalty 함수

3. 모델의 設定

假定 ②에 의해 時間 t 까지의 Counting Process $N(t)$ 는 다음과 같은 Poisson 分布를 한다.

$$Pr\{N(t) = n\} = \text{poim}(n; m(t)), \quad n = 0, 1, 2, \dots \quad (3-1)$$

$$m(t) = N(1 - e^{-\tau t}), \quad N > 0, \tau > 0, t \geq 0 \quad (3-2)$$

假定 ③에 의해 豫定된 引渡時期를 넘길 경우, Penalty 費用은 그 期間 ($t-D$)만큼 線型으로 增加한다.

$$P(t) = \begin{cases} 0, & t < D \\ C_p(t-D), & t \geq D \end{cases} \quad \dots (3-3)$$

그러므로 Software 테스트와 運用的 總平均費用은

$$C(t) = \begin{cases} C_a m(t) + C_b \{m(T_{LC}) - m(t)\} + C_c t, & t < D \\ C_a m(t) + C_b \{m(T_{LC}) - m(t)\} + C_c t + C_p(t-D), & t > D \end{cases} \quad \dots (3-4)$$

로 나타낼 수 있다.

式 (3-4)의 費用모델을 圖表로 表示하면 그림

1과 같다.

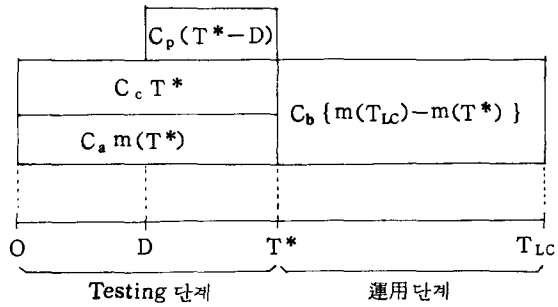


그림 1. A configuration of software cost model

最適의 Software 放出時期는 式 (3-4)의 總平均 Software 費用을 最少化시키는 時點이 된다. 그리고 또 하나의 評價基準인 Software 信賴度는 NH-PP 모델로부터 구할 수 있다.

Software 信賴度 $R(x/t)$ 는

$$R(x/t) = \exp[-m(x)e^{-\tau t}], \quad x \geq 0, t \geq 0 \quad \dots (3-5)$$

이다.

x 는 放出된 Software의 일정한 運用時間이고, 規定된 Software 信賴度を 滿足시키는 testing 時間 t 는 또 다른 最適 Software 放出時間이다.

여기서 式 (3-4)의 母數들에 制約을 부여하면 總平均 Software 費用을 最少化시키는 유일한 解를 얻을 수 있다.

式 (3-4)의 解를 얻기 위해 $dC(T)/dT = 0$ 로 하면 $N\tau(C_b - C_a) > C_c C_p$ 일 경우에

$$\tau T_0 = \ln\{N\tau(C_b - C_a)/C_c C_p\}, \quad (C_p = 1, t < D) \quad \dots (3-6)$$

이 된다. 그리고 境界條件은 $T_0 = 0$ 이다.

마찬가지로 Software의 Reliability를 요구된 R_0 에 近似시키는 유일한 時點을 式 (3-5)로부터 얻을 수 있다.

$$R(x/0) < R_0 \text{ 면 } R(x/T) = R_0 \text{의 解는 } \tau T_1 = \ln\{m(x)\} - \ln\{\ln(1/R_0)\} \quad \dots (3-7)$$

에서 구한다. 境界條件은 $T_1 = 0$ 이다.

4. Software의 最適放出政策

事전에 요구된 Software의 信賴도와 정해진 引渡時期를 넘길 경우 Penalty 變數가 값을 갖는 制約下에서 總平均費用 $C(T)$ 를 最少化하는 最適放出問題를 고려한다. 따라서 이 Software의 最適放出問題는 다

음과 같이 構成된다.

$$\text{Minimize } C(T) = C_a m(T) + C_b \{ m(T_{LC}) - m(T) \} + C_c T + C_p (T - D)$$

$$\text{Subject to } R(x/T) \geq R_0 \\ C_p = k, T > D \\ C_p = 0, T \leq D \dots\dots (4-1)$$

式(4-1)에서 먼저 첫번째 制約式을 滿足시키는 하나의 最適解 T_1 을 式(3-7)에서 구하고, 이 解에 따라 目的式에서 다른 하나의 最適解 T_0 를 구해 Software 시스템의 最適解를 찾는다.

$$T_1 > D \text{ 이면} \\ \tau T_0 = \ln \{ N\tau (C_b - C_a) / C_c C_p \} \dots\dots (4-2-1)$$

$$T_1 \leq D \text{ 이면} \\ \tau T_0 = \ln \{ N\tau (C_b - C_a) / C_c \} \dots\dots (4-2-2)$$

에서 T_0 를 얻는다.

그러므로 다음과 같이 最適放出政策을 세울 수 있다.

$T_1 > D$ 이면

① $N\tau (C_b - C_a) > C_c \cdot C_p$ 이고 $R(x/0) < R_0$ 이면 $T^* = \max \{ T_0, T_1 \}$

② $N\tau (C_b - C_a) > C_c \cdot C_p$ 이고 $R(x/0) \geq R_0$ 이면 $T^* = T_0$

③ $N\tau (C_b - C_a) \leq C_c \cdot C_p$ 이고 $R(x/0) < R_0$ 이면 $T^* = T_1$

④ $N\tau (C_b - C_a) \leq C_c \cdot C_p$ 이고 $R(x/0) \geq R_0$ 이면 $T^* = 0$

$T_1 \leq D$ 이면

① $N\tau (C_b - C_a) > C_c$ 이고 $R(x/0) < R_0$ 이면 $T^* = \max \{ T_0, T_1 \}$

② $N\tau (C_b - C_a) > C_c$ 이고 $R(x/0) \geq R_0$ 이면 $T^* = T_0$

③ $N\tau (C_b - C_a) \leq C_c$ 이고 $R(x/0) \geq R_0$ 이면 $T^* = T_1$

④ $N\tau (C_b - C_a) \leq C_c$ 이고 $R(x/0) \geq R_0$ 이면 $T^* = 0$

5. 数值例

Okumoto & Goel [2]의 데이터를 引用한다. $N = 1348$, $\tau = 0.124$, $x = 0.1$, $R_0 = 0.8$, $D = 30$, $T_{LC} = 100$, $C_a = 1$, $C_b = 15$, $C_c = 50$, $C_p = 2$.

이 값을 利用하여 最適放出時點을 구하면 $T_1^* = 34.8$ 이고 $T_1^* > D$ 이므로 式(4-2)에서 $T_0^* =$

25.4를 얻는다.

그러나 費用의 最適解인 $T_0^* = 25.4$ 에서는 Software 信賴度 $R(0.1/T_0^*) = 0.37$ 이므로 最適放出政策에 의해 最適放出時點 $T^* = 34.8$ 이다. 이 Software 最適放出時點은 주어진 條件 $R_0 = 0.8$ 을 만족하고 總費用 $C(T^*) = 2732$ 이다.

위의 結果를 圖表로 나타내면 그림 2와 같다.

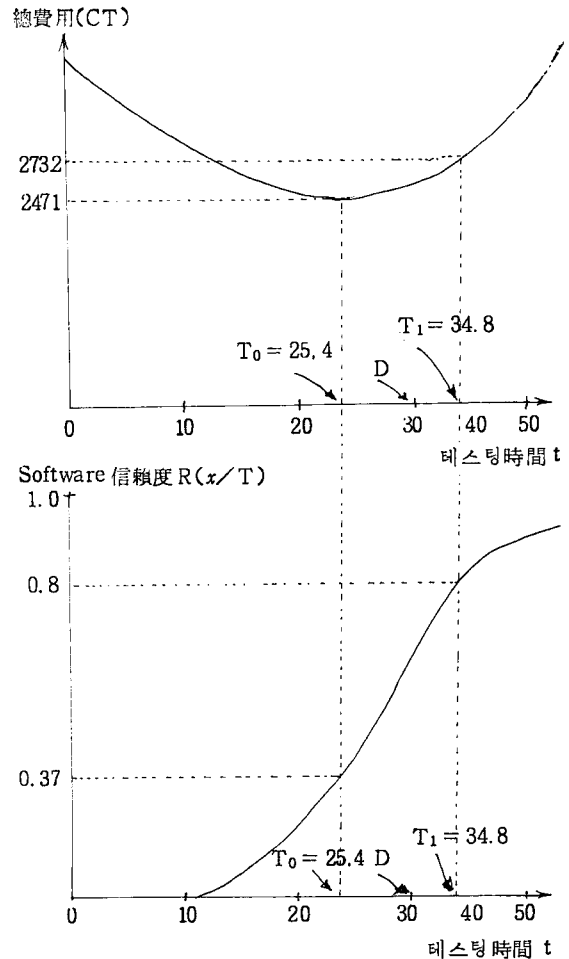


그림 2. An illustration of the cost-reliability optimal release problem

6. 結 論

從來의 最適 Software 放出政策에 관한 研究들은 tester의 立場에서나 user의 편에서도 반드시 必要한 豫定引渡時期를 고려하지 않았다.

本 研究에서는 Software의 最適放出時期的 決定에 있어 豫定引渡時期를 고려하여 研究의 實用性を 擴張시킨 점에 그 意義가 있다 하겠다.

本 研究에서는 여러 母數들을 過去의 經驗 혹은 資料에서 얻은 既知의 數라고 假定했으나, 初期 error 數 N , error의 發見率 r 등은 test를 해 나가는 過程에서 적절한 推定值를 구해 使用하는 것이 더욱 實用的일 것이다. 또한 放出後 發見된 error의 高치는 費用 C_b 의 正確한 推定을 할 수 있다면 더욱 더 合理的인 最適放出政策이 될 것이다.

參考文獻

1. J.D. Musa, "A theory of software reliability and its application", *IEEE Trans. Software Eng.*, vol. SE-1, pp. 312-317, May 1975.
2. K. Okumoto and A.L. Goel, "Optimum release time for software systems", *Proc. COM-PSAC*, pp. 500-503, Nov. 1979.
3. H. S. Koch and P. Kubat, "Optimal release time for computer software", *IEEE Trans. Software Eng.*, vol. SE-9, pp. 323-327, May 1983.
4. 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, pp. 206-211, Aug. 1979.
5. C. V. Ramamoorthy and F. B. Bastani, "Software reliability - status and perspectives", *IEEE Trans. Software Eng.*, vol. SE-8, pp. 354-371, Aug. 1982.
6. S. Yamada, H. Narihisa, and S. Osaki, "Optimum release policies for a software system with a scheduled software delivery time", *Int. J. Syst. Sci.*, vol. 15, pp. 905-914, Aug. 1984.
7. S. Yamada and S. Osaki, "Cost - reliability optimal release policies for software systems", *IEEE Trans. Rel.*, vol. R-34, No. 5, Dec. 1985.