일정한 使用期間을 갖는 Software System의 最適放出政策
Optimal Release Time of Software System having a Fixed Operational Period

崔 明 鎬*
尹 德 均**

ABSTRACT

This paper deals with Software Release policy having a Fixed Operational Period after a Software System is released.

The underlying Software Reliability Growth is described by J-M Model.

Optimal Software Release Time minimizing Total Expected Cost is obtained.

1. 서 론

Computer Software System의 중요성이 인식되기 시작한 1970년대 이후 많은 Software System의 信賴性에 관한 研究論文이 발표되었다. 또한 Software System의 信賴性에 대한 요구가 점증되고 있다. Software System의 테스트에 필수적으로 제기되는 2가지 問題 즉,

① 테스트가 끝난 시점에서 잔존하는 誤謬의 수를 예측하는 問題
② 예측을 바탕으로 테스트의 완료 시점을 성하는 問題

이다.

①의 問題는 각각의 Software System에 알맞은 誤謬의 發見 模型을 만들고, Parameter를 推定하는 방법이다.

②의 問題는 각 System의 성질을 잘 나타내는 誤謬 發見 模型을 적용하여 언제 테스트를 중단하고 System을 User에게 넘겨줄 것인가를 결정하는 問題 즉, 最適放出政策問題(Optimal Release Policy Problem)이다.

본 研究에서는 기존의 放出問題를 바탕으로 새로운 放出政策을 제시하고자 한다. 기존의 放出政策模型 중에서 Koch & Kubat[5] Model은 放出後 使用期間을 무한대(∞)로 두고, System의 最適放出時間 T^*를 구하였다. Bai & Yun[1] Model에서는 放出後 使用期間을 무한대(∞)로 두고 System의 最適放出誤謬個數 n^*을 구하였다. Goel & Lkumoto[3] Model은 테스트 시점에서 폐기처분되는 시점까지를 Software의 總壽命週期(T_{LC} : Total Life Cycle)로 정의하고, 이때 最適放出時間(Optimal Release Time)T^*을 구하였다.

이 경우 테스트단계의 所要期間과 放出後의 使用期間 사이에는 상충관계(trade-off)가 존재한다. 테스트 단계의 期間이 길어지면 放出後의 使用期間이 단축되며, 그 반대인 경우로 테스트 단계의 期間이 짧아지면 放出後의 使用期間은 실제적으로 길어진다. 따라서 테스트 단계의 期間이 매우 길어지면 放出하는 즉시 System은 폐기처분(死藏)되어야 하는 問題點을 갖게된다. 이러한 問題點을 해결하기 위해서 본 論文에서 새로운 放出政策을 고안하여 제시하고자 한다.

*蔚山專門大學 工業經營科 教授
**漢陽大學校 產業工學科 教授

접수일 1990. 8. 21.

Software System을 開發하여 User에게 放出했을 때 모든 System들은 얼마동안의 使用期間을 경과하여 폐기처분(死藏)되는 것은 경험적으로나, 실제적으로 보다 현실적인 問題이다. 따라서 테스트 기간을 거쳐 User에게 放出된 System은 一定期間 동안 사용된 後에 폐기처분(死藏)되는 경우를 模型化하였다.

Software System의 성질을 나타내는 誤謬의 發見은 Hardware System과 상당히 다른 성질을 포함하고 있다. Hardware의 고장은 마모현상이나 우연에 의해서 유발되지만 Software의 고장은 Software 그 자체 내부에 현존하고 있는 誤謬가 특수한 入力資料에 의해서 현재화될 때 즉, 고장이 일어나게 된다. Software를 완벽하게 하기 위해서는 모든 가능한 入力에 대해서 Software를 테스트해야 하나 이는 불가능하므로 Software의 誤謬가 있음을 입증할 수는 있어도 誤謬가 없음을 입증하기에는 현재의 기술로도 불가능한 실정이다. 오직 가능한 것이 있다면 현존하는 誤謬의 個數를 그때까지 테스트한 결과를 기초로 예측하는 것이다. 즉, Software System의 發見된 誤謬를 기초로 誤謬의 個數를 예측하여왔다. Software System 내의 誤謬가 발견되는 Process에 대하여 많은 硏究가 있어 왔다. 그 중에서 J-M Model[4]은 지수분포를 갖는 Order Statistic Process[7]로 보는 것이고, G-O Model[3]은 NHPP[6](Nonhomogeneous Poisson Process)로 해석하는 것이다. 그러나 J-M Model이 보다 많이 사용되고 있다.

본 論文에서는 System의 誤謬 發見이 J-M Model에 의하여 해석될 때 放出後 一定期間 사용된다고 하고, 總費用을 最小化하는 最適放出時間을 구하는 것에 목적을 두고 있다.

또한 본 模型은 Koch & Kubat와 Goel & Okmoto의 放出政策模型의 一般化이다.

2. 放出後 一定期間동안 사용되는 Software System Model

2.1 記號의 定義

본 論文에서 사용하는 정의는 다음과 같다.

T	테스트시간
T^*	最適放出시간(최적 테스트 중단 시간)
M	테스트단계의 초기에 시스템에 존재하는 總 誤謬의 個數
$M(T)$	테스트시간 T시간 까지 발견된 誤謬의 평균수($\int_0^T \rho(x)dx$)
$\rho(T)$	T시간에서의 誤謬의 發見率($\frac{d}{dT}M(T)$)
τ	Software System의 放出後의 使用期間(τ : 一定한 상수)
$C[T]$	總費用函數
C_1	단위 테스트시간당 費用(檢査費用)
C_2	Software System 放出 前 단위 誤謬 당 修正費用
C_3	Software System 放出 後 단위 誤謬 당 修正費用(단, $C_3>C_2$)

2.2 放出後 一定期間 동안 사용되는 Software System Model 開發

Software System을 개발하여 만족한 信賴度 수준에 이르기 위해서 테스트단계를 거쳐야 한다. 이후 System은 User에게 넘겨져 一定期間 동안 사용되고 폐기처분 된다. 이 전 과정을 System의 總壽命週期(T_{LC})이라 하고, 放出後 사용되어 폐기될 때 까지의 期間은 使用期間이라 한다.

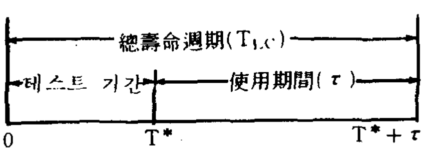

Figure 1. Life Cycle X is a Fixed Period

위 Figure 1과 같이 總壽命週期는 테스트期間과 使用期間으로 구분된다. 使用期間(τ)이 一定한 경우 最適放出期間 T^*를 구하는 模型을 開發하고자 한다. 模型의 기본가정은 다음과 같다.

① 誤謬는 발견과 동시에 제거된다.

② 故障 修正 과정 동안 새로운 고장은 삽입되지 않는다.

③ 故障率은 남아있는 誤謬 個數에 비례한다.

④ 使用期間 τ는 一定하다.

誤謬를 발견하기 위해 Software를 테스트하는데는 Hardware의 費用을 포함해서 이 작업에 대한 人件費 등 費用을 수반하게 된다. Software를 테스트 없이 방출할 경우, 誤謬에 의한 費用이 발생하게 될 것이다. 즉 Software의 檢査費用과 放出 前 修正費用 그리고 放出後 修正費用이 발생하게 되며, 이러한 費用의 적절한 分配問題가 제기된다.

본 論文에서는 費用에 관한 3가지 기본가정을 하고 있다.

(1) 테스트費用은 테스트 시간에 비례해서 지출된다.

(2) 放出前 修正費用은 테스트시간(T)까지 발견된 誤謬의 수에 비례하여 지출된다.

(3) 放出後 修正費用은 放出後 使用期間 내에 System에 잔류하는 誤謬의 수에 비례해서 지출된다.

가정 (1)에 의해서 테스트시간을 T라 할 때 테스트費用은 다음 식과 같다.

$$C_a(T)=C_1 \cdot T \quad \cdots\cdots (1)$$

가정 (2)에 의해서 放出前 修正費用은 테스트시간을 T라 할 때 그 시간까지 발견된 誤謬의 수에 비례하므로 다음 식과 같다.

$$C_b(T)=C_2 \cdot M(T) \quad \cdots\cdots (2)$$

가정 (3)에 의해서 放出後 修正費用은 테스트시간을 T라 할 때 그 시간 이후 使用期間내에 System에 잔류하는 誤謬의 수에 비례하므로 다음 식과 같다.

$$C_f(T)=C_3\{M(T+\tau)-M(T)\} \quad \cdots\cdots (3)$$

식(1), 식(2), 식(3)으로부터 總費用 函數 C[T]을 구하면 다음 식과 같이 된다.

$$C[T]=C_1 \cdot T+C_2 \cdot M(T)+C_3\{M(T+\tau)-M(T)\} \quad \cdots\cdots (4)$$

식(4)를 最小化하는 T*를 구하기 위하여 식(4)를 T에 대하여 미분하면 다음 식과 같이 된다.,

$$\frac{d}{dT}C[T]=C_1+C_2 \cdot \rho(T)+C_3\{\rho(T+\tau)-\rho(T)\} \quad \cdots\cdots (5)$$

最適時間(Optimal Time) T*는 식(5)를 0으로 만족해야 한다.

다음으로 最適解(Optimal Solution) T*가 유일(Unique)함을 보자.

$\frac{d}{dT}C[T]=0$을 정리하여 쓰면 다음 식과 같다.

$$-C_1=C_2 \cdot \rho(T)+C_3\{\rho(T+\tau)-\rho(T)\} \quad \cdots\cdots (6)$$

식(6)에서 등호 오른쪽을 [R. H. S]라 하면 다음 식과 같다.

$$[R.H.S]=C_2 \cdot \rho(T)+C_3\{\rho(T+\tau)-\rho(T)\} \quad \cdots\cdots (7)$$

따라서 T*가 유일(Unique)하기 위한 필요조건은 식(7)이 항상 單調增加 또는 單調減少하여야 한다. 따라서 [R. H. S]를 미분하면 다음과 같다.

$$\begin{aligned}\frac{d}{dT}[R.H.S]&=C_2 \cdot \rho'(T)+C_3\{\rho'(T+\tau)-\rho'(T)\}\\&=\left[C_2+C_3\left(\frac{\rho'(T+\tau)}{\rho'(T)}-1\right)\right] \cdot \rho'(T) \quad \cdots\cdots (8)\end{aligned}$$

식(8)에서 주어진 τ에 대하여 배타으로 다음의 2가지의 경우가 발생된다.

① $\left[C_2+C_3\left(\frac{\rho'(T+\tau)}{\rho'(T)}-1\right)\right]\cdot\rho'(T)\leq 0$인 경우에서는

[R. H. S]는 減少函數(Decreasing Function)이다.

② $\left[C_2+C_3\left(\frac{\rho'(T+\tau)}{\rho'(T)}-1\right)\right]\cdot\rho'(T)\geq 0$인 경우에서는

[R. H. S]는 增加函數(Increasing Function)가 된다.

왜냐하면 J−M Model에서 $\frac{\rho'(T+\tau)}{\rho'(T)}$은 τ만의 函數이고 $\tau=0$일 때 1이고 τ가 增加함에 따라 單調減少하여 $\tau=\infty$일때 0이기 때문이다.

①의 경우에 있어서 最適解(Optimal Solution)의 존재를 알아 보자.

$$[R.H.S]_{T=0}=C_2\cdot\rho(0)+C_3\{\rho(\tau)-\rho(0)\}$$
$$=\left[C_2+C_3\left(\frac{\rho(\tau)}{\rho(0)}-1\right)\right]\cdot\rho(0)>0 \quad \cdots\cdots (9)$$

$$[R.H.S]_{T=\infty}=0 \quad \cdots\cdots (10)$$

식(9)가 성립하는 이유는 아래와 같다. $\rho'(T)$는 0 보다 항상 적으므로 경우 ①이 성립하는 경우 $\left[C_2+C_3\left(\frac{\rho'(T+\tau)}{7'(T)}-1\right)\right]$는 0 보다 항상 커야 한다.

식(9)에서 $\rho(0)$는 0 보다 항상 크므로 $\left[C_2+C_3\left(\frac{\rho(\tau)}{\rho(0)}-1\right)\right]$도 0 보다 커야 한다.

왜냐하면 J−M Model에서는 모든 T에 대하여

$$\frac{\rho'(T+\tau)}{\rho'(T)}=\frac{\rho'(\tau)}{\rho'(0)}=\frac{\rho(\tau)}{\rho(0)}=\frac{\rho(T+\tau)}{\rho(T)}$$

이기 때문이다.

따라서 식(9)와 식(10)에 의하여 [R. H. S]는 T=0에서 양의 값을 갖고 T에 따라 單調減少하다가 T=∞에서 거의 0에 접근하고 이를 그림으로 나타낸 것이 Figure 2이다.

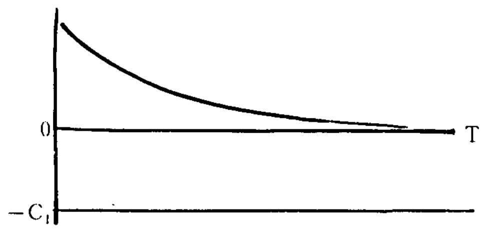

Figure 2. The region of [R. H. S] in the case ①

따라서 $\frac{d}{dT}C[T]=0$을 만족하는 T가 존재하지 않는다.

그러나 $\frac{d}{dT}C[T]>0$이므로 C[T]가 增加函數이고 $T^*=0$이다.

②의 경우에 있어서 最適解(Optimal Solution)의 존재를 알아보자.

$$[R.H.S]_{T=0}=\left[C_2+C_3\left(\frac{\rho(\tau)}{\rho(0)}-1\right)\right]\cdot\rho(0)<0 \quad \cdots\cdots (11)$$

$$[R.H.S]_{T=\infty}=0 \quad \cdots\cdots (12)$$

식(11)이 0보다 적어야 하는 이유는 경우 ①의 경우와 마찬가지로 보일 수 있다. 따라서 식(11), 식(12)에서

$[R.H.S]$는 $T=0$에서 음의 값을 갖고 T에 따라 增加하다가 $T=\infty$에서 0에 접근한다. 따라서 이를 그림으로 표현하면 Figure 3과 같다.

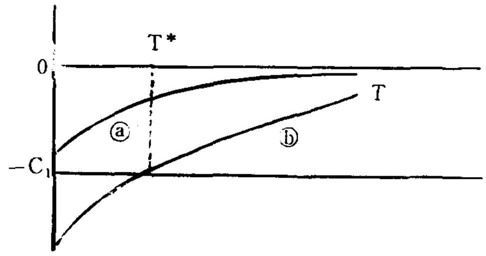

Figure 3. The Region of [R. H. S] in the case ②

ⓐ의 경우는 $[R.H.S]_{T=0}=[C_2\cdot\rho(0)+C_3(\rho(\tau)-\rho(0))]\geq -C_1$이고 앞에서와 같이 T^*0이다. ⓑ의 경우는 $[R.H.S]_{T=0}=[C_2\cdot\rho(0)+C_3\cdot\rho(\tau)-\rho(0))]<-C_1 T^*\fallingdotseq 0$이다. 따라서 위의 사실에 의하여 最適時間(Optimal Time) T^*는 유일(Unique)하다.

T시간 까지 誤謬의 發見率 $\rho(T)$가 J−M Model에서 제시한 減少函數 $\rho(T)m\lambda e^{-\lambda T}$로 주어질 때 ⓑ의 경우 最適時間(Optimal Time) $T^*(\fallingdotseq 0)$는 다음과 같다.

$$\frac{d}{dT}C[T]=C_1+C_2\cdot\rho(T^*)+C_3\{\rho(T^*+\tau)-\rho(T^*)\}=0 \quad \cdots\cdots (13)$$

$$C_1+C_2\cdot M\lambda e^{-\lambda T}+C_3\{M\lambda e^{-\lambda(T+\tau)}-M\lambda e^{-\lambda T}\}=0 \quad \cdots\cdots (14)$$

식(14)를 전개하여 풀면 T^*는 다음 식과 같다.

$$T^*=\frac{\ln\left\{\dfrac{-C_1}{M\lambda\,[C_2+C_3(e^{-\lambda\tau}-1)]}\right\}}{-\lambda} \quad \cdots\cdots (15)$$

본 論文에서 제시한 放出後 일정한 使用期間을 갖는 경우는 Koch & Kubat[5] Model과 Goel & Okumoto [3] Model을 일반화시킨 模型이 된다.

본 模型에서 使用期間 τ를 무한대(∞)로 두면 Koch & Kubat 放出模型이 되고 使用期間 τ를 $\tau=T_{LC}-T$로 두면 G−0 放出模型이 된다.

〈NUMERICAL EXAMPLE〉

Koch & Kubat[5] Model의 例에서 $C_1=21500$(원/월), $C_2=2000$(원/개), $C_3=3500$(원/개), $M=30$(개), $\lambda=1.2$(개/월)로 주어져 있다. 使用期間 τ가 주어지면 最適放出時間(Optimal Release Time) T^*을 구할 수 있다. $\tau=2$(년)일 때

$[R.H.S]_{T=0}=2000\cdot 30\cdot 1.2+3500(30\cdot 1.2e^{-1.2\cdot 2}-30\cdot 1.2)<-21500$이므로 식(15)에서 $T^*=0.569$(월)이고 이때 $M(T)=17.8$(개)이다. Koch & Kubat[5] Model에서는 $T^*=0.81$(월)이다.

따라서 본 論文에서 제시한 模型은 最適放出時間(Optimal Release Time) T^*가 기존 論文에 비하여 짧은 데 이는 System의 使用期間이 $\tau=2$로 주어졌기 때문이다.

3. 결 론

본 論文에서 제시한 模型은 放出後 使用期間이 테스트期間에 영향이 없이 一定하게 주어지는 경우 最適放出政策에 관한 研究이다. 이것은 실제로 Software System이 무한성의 使用期間을 가질 수 없기 때문에 이를 반영한 매우 현실적인 경우이다.

본 模型은 放出後 使用期間이 무한대(∞)인 경우의 模型이나 放出後 사용되는 期間이 放出시점에 영향을 받는 模型을 一般化시킨 것이다.

⟨Reference⟩

1. Bai, D. S. & Yun, W. Y. (1988), "Optimum Number of Errors Corrected before Releasing a Software System," *IEEE transaction on Reliability, 37(1)*, pp. 41~44, April.
2. Donelson, Ⅲ, J. (1977), "Cost Model for Testing Program BAsed on Nonhomogeneous Poisson Failure Model," *IEEE transaction on Reliability, 26(3)*, pp. 189~194, August.
3. Goel, A. L. & Okumoto, K. (1979), "Optimum Release Time for Software System," *IEEE transaction on Reliability, R-28(3)*, pp. 206~211.
4. Jelinski, Z. & Moranda, P. B. (1983), *Software Relibility Research*, in Statistical Computer Performance Evaluation(W. Freiberger, ed.), Academic Press, pp. 465~484.
5. Koch, H. S. & Kubat, P. (1983), "Optimal Release Time of Computer Software," *IEEE transaction on Software Engineering, SE-9(3)*, pp. 323~327.
6. Thompson, Jr, W. A. (1981), "On the Foundations of Reliability," *Technometrics, 23(1)*, pp. 1~13, February.
7. Ascher, H. & Feingold, H. (1984), *Repairable System Reliability*, Marcel Dekker, INC., New York & Basel.