

시험시간과 오류수정개수를 고려한 소프트웨어 출시 시점결정

유영관(한라대학교 경영학과)*

국 문 요 약

본 논문에서는 시험시간과 오류의 수정개수를 동시에 고려하는 소프트웨어 출시 모형을 제시한다. 소프트웨어의 시험기간은 정해진 시험 시간이나 오류수정개수 중 먼저 도래하는 시간까지 지속된다. 고려되는 비용은 시험기간 중의 오류수정비용과 출시 지연비용, 그리고 시험 기간 경과 후의 운용 중 발생하는 오류수정비용으로 구성된다. 생명주기는 특별한 제한이 없고, 오류의 검출과정은 비제차 포아송과정을 따른다고 가정한다. 총비용함수가 도출되며 이전의 출시방안들은 제시된 모형의 특별한 경우임을 보인다.

핵심주제어: 소프트웨어 출시, 소프트웨어 오류, 소프트웨어 고장, 생명주기 비용

1. 서론

최근 들어 소프트웨어의 규모가 점점 커지고 복잡해짐에 따라 시스템 개발 비용이 크게 증가하고 있다. 따라서 언제 소프트웨어 시험을 끝내고 사용자에게 출시(release)하는 것이 소프트웨어의 신뢰도를 충족시키면서 비용 면에서 경제적인가를 결정하는 것은 매우 중요한 일이 되고 있다.

일반적으로 소프트웨어의 품질은 시험 시간의 길이에 달려 있다고 할 수 있다. 즉, 시험을 길게 하면 결함 검출 확률을 높여 신뢰도가 증가한다. 그러나 시험 비용은 시간에 비례하여 증가하게 되므로 무한정 늘릴 수는 없다. 반대로 시험을 짧게 하면 비용은 적게 들지만 신뢰성이 떨어지는 소프트웨어를 고객이 구입하게 되므로 운용 중 고장이 발생할 확률이 높아진다. 이 경우 운용 단계에서의 고장은 시험단계에서의 고장에 비해 훨씬 큰 비용이 소요되므로 개발자의 비용 지출이 증가하게 된다.

소프트웨어 시험이 소프트웨어 내의 결함을 제거할 수 있는 가장 효율적인 방법이긴 하지만, 모든 결함을 발견해 낼 수 있는 모든 실행 경로에 대해 시험한다는 것은 현실적으로 거의 불가능하며 비실용적이다. 또한 시험이 어느 정도 진행되고 난 후에는 더 이상의 결함을 찾아내기 위해 필요한 시간과 비용이 지수적으로 증가하게 된다. 따라서 적절한 시기에 시험을 멈추고 고객에게 제품을 출시하는 것이 시험 비용, 제품 보증 비용, 그리고 고객의 위험 비용 면에서 바람직하다고 할 수 있다.

소프트웨어 생명주기(life cycle)에 걸친 총 비용이 최소가 되는 소프트웨어 시험 시간의 결정은 상충되는 비용 요소들의 절충(trade-off)이라고 할 수 있다. 소프트웨어 생명주기의 총

비용은 크게 고객에게 출시되기 전의 비용과 출시된 후의 비용으로 나눌 수 있다. 소프트웨어 시험 시간이 증가하면 시험에 소요되는 비용 등 출시 전의 비용이 증가하게 되고, 시험 시간이 짧으면 고객의 소프트웨어 사용 중 고장으로 인한 수리비용과 호감 상실 비용 등 출시 후의 비용이 증가하게 된다. 소프트웨어 생명주기에 걸친 총 비용은 이 둘을 합한 것이며 총 비용이 최소가 되는 지점에서 최적 시험시간, 즉 출시 시점을 구할 수 있다.

이와 같이 소프트웨어의 시험 시간을 얼마로 정하는 것이 소프트웨어의 신뢰성이나 수명주기 총 비용 면에서 최적인가의 문제를 최적 소프트웨어 출시문제(optimal software release problem)라고 한다. 소프트웨어 출시 모형에 관한 연구들은 소프트웨어 신뢰도 모형을 설정하고 소프트웨어 생명주기 상의 관련 비용들을 합하여 총 비용을 구한 후, 이 총 비용이 최소가 되는 최적 시점을 구하는 것으로 요약된다.

1.1 소프트웨어 신뢰도 모형

지난 25년간 소프트웨어 신뢰도공학 분야에 대한 연구가 진행되어 오면서, 50여 개 이상의 소프트웨어 신뢰도 모형이 개발되어 소프트웨어의 신뢰도 예측에 기여하고 있다. 소프트웨어 신뢰도 모형의 목적을 한마디로 요약하면 소프트웨어에 남아 있는 오류의 수를 추정하는 것이라고 할 수 있다. 알려진 바에 의하면 단위 시험 후 1,000개의 명령문에는 대략 8개의 오류가 있다고 한다(Pham 2000). 이 숫자는 물론 평균적인 것이고 시험은 고려하지 않은 것이다.

소프트웨어 신뢰도 모형에는 크게 두 가지, 확정적 모형과 확률적 모형이 있다. 확정적 모형은 명령어나 연산자의 수 등의 프로그램의 구조를 분석하여 오류의 수를 추정하는 데

* 주저자, 한라대학교 경영학과 교수, ykyoo@halla.ac.kr.

투고일: 2012-11-16 · 수정일: 2012-12-18 · 게재확정일: 2012-12-20

사용되며, 소프트웨어 메트릭(metric)과 McCabe(1976)의 cyclomatic complexity metric이 대표적이다. 반면에 확률적 모형에서는 고장의 발생과 결함의 제거를 확률적 사건으로 취급한다. 확률적 신뢰도 모형은 근본적인 가정과 분석 기법에 따라 오류과중(error seeding)모형, 고장률(failure rate)모형, 곡선적합(curve fitting)모형, 신뢰도 성장(reliability growth)모형, 비제차 포아송 과정(non-homogeneous Poisson process, NHPP) 모형, 마아코브(Markov)모형 등으로 나뉜다.

1.2 수명주기 비용

소프트웨어 수명주기 전체를 통해 발생하는 비용들 중 소프트웨어의 시험과 출시에 관련된 주요 비용들은 다음과 같다 (Pham 2000).

(1) 단위 시험시간 당 비용

소프트웨어를 시험하는데 소요되는 비용으로서 시간은 일반적으로 CPU time을 의미한다.

(2) 시험 중 오류 수정비용/개

소프트웨어의 시험 중 발생하는 오류를 하나 수정하는데 소요되는 비용이다.

(3) 시험 후 오류 수정비용/개

소프트웨어가 출시된 후 발생하는 오류를 하나 수정하는데 소요되는 비용이다. 고객이 소프트웨어를 운용하는 중에 발생한 고장의 수리비용이므로 시험과정의 오류 수정비용에 비해 일반적으로 크다고 할 수 있다.

(4) 호감상실 비용

소프트웨어 운용 중 발생 고장에 대해 고객이 갖게 되는 호감 상실 비용으로서 (3)에 포함시킬 수도 있다.

(5) 벌과 비용(penalty cost)

계획된 인도일에 맞추어 출시하지 못했을 경우 발생하는 벌과비용이다.

소프트웨어 출시모형에 관한 고전적 연구는 비제차 포아송 과정(이하 NHPP) 소프트웨어 신뢰도 성장모형(Goel & Okumoto 1979)을 기반으로 한 Okumoto & Goel(1980)의 연구이다. Yamada & Osaki(1985)는 소프트웨어 생명주기 상의 총 비용을 최소화 하면서 출시 후의 요구 신뢰도를 동시에 충족할 수 있는 모형으로 발전시켰다. 이외에도 여러 요소들을 고려한 다양한 출시 모형이 개발되었다.

이전의 대부분의 연구들은 소프트웨어 출시시점의 결정 변수로써 시험시간을 주로 사용하고 있다(박일광·공명복 2007, 이진승 등 2004, Boland & Chuiv 2007, Chatterjee et al. 1997, Hou et al. 1996, Huang 2005, Koch & Kubat 1983, Kumar et al. 1995, Pham & Zhang 2003, Xie & Hong 1999).

그러나 소프트웨어의 출시 시점을 반드시 시험시간으로만 정할

필요는 없으며, 충분한 수의 오류가 검출되었다면 시험을 멈추고 출시준비를 하는 것도 가능할 것이다. 출시시점의 결정변수로써 검출된 오류개수를 사용한 연구로는 Bai & Yun(1988)과 Shinohara et al.(1997), 그리고 유영관(2011)을 들 수 있다.

출시시점을 결정할 때 시험시간을 기준으로 사용하는 모형(이하 T-모형)과 오류개수를 기준으로 하는 모형(이하 N-모형)은 각각의 장단점이 있다. T-모형은 소프트웨어의 출시기한이 명확히 규정되어 있을 때 이를 지킬 수 있으나 충분한 수의 오류를 검출해 내었는지는 보장할 수 없다. N-모형은 미리 규정한 충분한 수의 오류개수를 검출해 낼 수 있으나 출시시점을 명확히 알 수 없으며 만약 규정된 출시기한이 넘어가게 된다면 곤란해진다.

본 연구에서는 소프트웨어의 출시시점을 결정하기 위해 T-모형과 N-모형을 혼합한 새로운 (N,T) 모형을 제시한다. 이 모형 하에서 소프트웨어 시험은 N개의 오류가 검출되어 수정되거나 일정한 시험시간 T의 경과 중 먼저 도래하는 시점까지 계속된다. 즉 충분한 수의 오류가 검출되면 시험을 중단하고, 만약 그렇지 않을 경우에도 정해진 시험기한이 다가오면 시험을 중단함으로써 현실적으로 두 경우에 모두 대응할 수 있도록 한 것이다.

다음 절에서는 소프트웨어 성장모형과 T-모형 및 N-모형에 대해 정리한다. 제3절에서는 제안된 (N,T) 모형 하에서의 수명주기 상의 총비용식이 도출되며, (N,T) 모형은 T-모형과 N-모형을 포함하고 있음을 보인다. 소프트웨어 신뢰도 성장모형은 Goel & Okumoto(1979)의 NHPP 모형을 따른다고 가정한다.

II. 소프트웨어 성장모형과 출시모형

2.1 소프트웨어 성장모형

본 연구에서 사용하는 Goel-Okumoto 소프트웨어 신뢰도 성장모형은 기본적으로 NHPP를 가정한다. 즉, 소프트웨어 시험이 진행됨에 따라 오류가 검출되는데, 오류의 도래(arrival)는 NHPP를 따르며 검출 즉시 수정된다. 따라서 시험이 진행됨에 따라 소프트웨어의 신뢰도는 증가(성장)하게 된다. 이제 $N(t)$ 를 시간 t 까지 검출된(수정된) 오류의 개수라고 하자.

가정에 의해 $\{N(t), t \geq 0\}$ 는 NHPP이므로

$$P\{N(t) = k\} = \frac{[m(t)]^k e^{-m(t)}}{k!}, k = 0, 1, 2, \dots \quad (1)$$

임을 알 수 있다. 또한

$$m(t) = E\{N(t)\} = a(1 - e^{-bt}) \quad (2a)$$

$$\lambda(t) \equiv m'(t) = abe^{-bt} \quad (2b)$$

는 각각 NHPP의 mean value function과 intensity function이며, 여기에서 $a = m(\infty)$ 는 이 소프트웨어에 포함되어 있는 오류

의 총 개수를, b 는 오류 당 고장률(hazard rate of an error)을 의미한다.

2.2 T-모형: 시험시간 모형

T-모형 하에서 소프트웨어 시험은 정해진 시험시간 T 까지 지속되며 이후 소프트웨어는 출시된다. 이제 다음과 같은 비용을 정의하자.

- c_1 시험기간 중 오류의 개당 수정비용
- c_2 출시 후 운용기간 중 오류의 개당 수정비용
- c_3 시험에 소요되는 단위시간 당 비용

소프트웨어 생명주기 상의 총 비용은 다음과 같이 구해진다. 시험기간 중 검출되는 오류의 개수에 대한 기댓값은 $m(T)$ 이므로 오류수정에 소요되는 총 비용은 $c_1m(T)$ 이고, 출시 후 예상되는 오류수정의 총 비용은 $c_2(m(\infty) - m(T)) = c_2(a - m(T))$ 이다. 따라서 수명주기 동안의 총비용은

$$C(T) = c_1m(T) + c_2(a - m(T)) + c_3T \quad (3)$$

이다. 식(3)을 T 에 관해 미분하면 T 의 최적해는 다음의 식을 만족하는 해임을 보일 수 있다(Okumoto & Goel 1980).

$$abe^{-bT} = \frac{c_3}{c_2 - c_1}$$

2.3 N-모형: 오류수정개수 모형

N-모형 하에서 소프트웨어 시험은 일정한 개수(N)의 오류가 검출되어 수정될 때까지 지속되며 이후 소프트웨어는 출시된다. 이제 T_N 을 N 번째 오류가 발생될 때까지의 시간이라고 하자. 그러면 다음의 관계식으로부터

$$\{T_N > t\} \Leftrightarrow \{N_1(t) < N\} \quad (4)$$

T_N 의 생존함수는 다음과 같다.

$$\begin{aligned} P\{T_N > t\} &= P\{N(t) < N\} \\ &= \sum_{k=0}^{N-1} P\{N(t) = k\} \\ &= \sum_{k=0}^{N-1} \frac{[m(t)]^k e^{-m(t)}}{k!} \end{aligned} \quad (5)$$

따라서 N 번째 오류가 발생될 때까지의 기대시간은

$$\begin{aligned} E[T_N] &= \int_0^\infty P\{T_N > t\} dt \\ &= \sum_{k=0}^{N-1} \int_0^\infty \frac{[m(t)]^k e^{-m(t)}}{k!} dt \end{aligned} \quad (6)$$

이다. 소프트웨어 생명주기 상의 총 비용은 T-모형에서와 마찬가지로 세 가지로 구성된다. 시험기간 중 검출되는 오류의 개수는 N 이므로 오류수정에 소요되는 총 비용은 c_1N 이고, 출시 후 예상되는 오류수정의 총 비용은 $c_2(a - N)$ 이다. 따라서 수명주기 동안의 총비용은 식(6)을 이용하여

$$C(N) = c_1N + c_2(a - N) + c_3 \sum_{k=0}^{N-1} \int_0^\infty \frac{m(t)^k e^{-m(t)}}{k!} dt \quad (7)$$

로 구해진다. 식(7)은 유영관(2011)에서 경미오류가 존재하지 않고 모두 중대오류만 존재하는 경우의 총 비용식과 일치한다.

III. (N,T) 소프트웨어 출시모형

(N,T) 모형 하에서 소프트웨어 시험은 N 개의 오류가 검출되어 수정되거나, 일정한 시험시간 T 의 경과 중 먼저 도래하는 시점까지 계속된다. 이 모형 하에서의 평균 시험기간을 구하기 위해 다음과 같이 정의하자.

$$T_M = \min\{T, T_N\} \quad (8)$$

그러면 평균 시험기간은 식(4), (5), (6)을 이용하여 다음의 수식 전개로 구할 수 있다.

$$\begin{aligned} E[T_M] &= E[\min\{T, T_N\}] \\ &= E[T_M | T_N > T] P\{T_N > T\} \\ &\quad + E[T_M | T_N \leq T] P\{T_N \leq T\} \\ &= TP\{T_N > T\} + E[T_N] P\{T_N \leq T\} \\ &= TP\{N(T) < N\} + E[T_N] P\{N(T) \geq N\} \\ &= T \sum_{k=0}^{N-1} \frac{m(T)^k e^{-m(T)}}{k!} \\ &\quad + \sum_{k=0}^{N-1} \int_0^\infty \frac{m(t)^k e^{-m(t)}}{k!} dt \cdot \sum_{k=N}^\infty \frac{m(T)^k e^{-m(T)}}{k!} \\ &= \sum_{k=0}^{N-1} \int_0^\infty \frac{m(t)^k e^{-m(t)}}{k!} dt + \sum_{k=0}^{N-1} \frac{m(T)^k e^{-m(T)}}{k!} \\ &\quad \times \left[T - \sum_{k=0}^{N-1} \int_0^\infty \frac{m(t)^k e^{-m(t)}}{k!} dt \right] \end{aligned} \quad (9)$$

식(9)는 다음과 같이 특별한 경우를 포함하고 있음을 알 수 있다.

$$E[T_M] = \begin{cases} T & N \rightarrow \infty \text{ (T-정책)} \\ \sum_{k=0}^{N-1} \int_0^\infty \frac{m(t)^k e^{-m(t)}}{k!} dt & T \rightarrow \infty \text{ (N-정책)} \end{cases} \quad (10)$$

이제 수명주기 상의 총 비용을 계산하여 보자.

- (1) 시험기간 중의 오류수정 총비용

N_M 을 시험기간 중의 오류개수라고 하자. 그러면 이 확률변수의 기댓값은 다음과 같이 구해진다.

$$\begin{aligned}
 E[N_M] &= E[N_M | T_N > T] P\{T_N > T\} \\
 &\quad + E[N_M | T_N \leq T] P\{T_N \leq T\} \\
 &= m(T) P\{T_N > T\} + NP\{T_N \leq T\} \\
 &= m(T) P\{N(T) < N\} + NP\{N(T) \geq N\} \\
 &= m(T) \sum_{k=0}^{N-1} \frac{m(T)^k e^{-m(T)}}{k!} \\
 &\quad + N \left[1 - \sum_{k=0}^{N-1} \frac{m(T)^k e^{-m(T)}}{k!} \right] \\
 &= N + (m(T) - N) \sum_{k=0}^{N-1} \frac{m(T)^k e^{-m(T)}}{k!}. \tag{11}
 \end{aligned}$$

따라서 시험기간 중의 오류수정 총비용은 다음과 같다.

$$c_1 E[N_M] = c_1 \left[N + (m(T) - N) \sum_{k=0}^{N-1} \frac{m(T)^k e^{-m(T)}}{k!} \right] \tag{12}$$

(2) 출시 후 오류수정 총비용

시험기간 이후 남아 있는 오류의 개수는

$$a - E[N_M] = a - N - (m(T) - N) \sum_{k=0}^{N-1} \frac{m(T)^k e^{-m(T)}}{k!} \tag{13}$$

이므로 출시 후 오류수정 총 비용은 다음과 같다.

$$c_2 \left[a - N - (m(T) - N) \sum_{k=0}^{N-1} \frac{m(T)^k e^{-m(T)}}{k!} \right] \tag{14}$$

(3) 시험비용

시험기간에 비례하여 발생하는 시험비용은 식(9)를 이용하면 다음과 같이 구해진다.

$$\begin{aligned}
 &c_3 \left[\sum_{k=0}^{N-1} \int_0^\infty \frac{m(t)^k e^{-m(t)}}{k!} dt \right. \\
 &\left. + \sum_{k=0}^{N-1} \frac{m(T)^k e^{-m(T)}}{k!} \left[T - \sum_{k=0}^{N-1} \int_0^\infty \frac{m(t)^k e^{-m(t)}}{k!} dt \right] \right] \tag{15}
 \end{aligned}$$

이상으로부터 (N,T)모형하의 수명주기 총 비용은 식(12), (14), (15)를 모두 더하여 다음과 같이 표현된다.

$$\begin{aligned}
 C(N, T) &= c_1 \left[N + (m(T) - N) \sum_{k=0}^{N-1} \frac{m(T)^k e^{-m(T)}}{k!} \right] \\
 &\quad + c_2 \left[a - N - (m(T) - N) \sum_{k=0}^{N-1} \frac{m(T)^k e^{-m(T)}}{k!} \right]
 \end{aligned}$$

$$\begin{aligned}
 &+ c_3 \left[\sum_{k=0}^{N-1} \int_0^\infty \frac{m(t)^k e^{-m(t)}}{k!} dt + \sum_{k=0}^{N-1} \frac{m(T)^k e^{-m(T)}}{k!} \right. \\
 &\quad \left. \times \left[T - \sum_{k=0}^{N-1} \int_0^\infty \frac{m(t)^k e^{-m(t)}}{k!} dt \right] \right] \tag{16}
 \end{aligned}$$

구해진 총 비용식 (16)은 T-모형과 N-모형을 특별한 경우로 포함하고 있다. 즉, $T \rightarrow \infty$ 이면 총비용식 (16)은

$$C(N, \infty) = c_1 N + c_2 (a - N) + c_3 \sum_{k=0}^{N-1} \int_0^\infty \frac{m(t)^k e^{-m(t)}}{k!} dt$$

가 되어 식(7)과 일치하고, $N \rightarrow \infty$ 이면 식(16)은

$$C(\infty, T) = c_1 m(T) + c_2 (a - m(T)) + c_3 T$$

가 되어 식(3)과 일치함을 알 수 있다.

IV. 결론

본 논문에서는 시험시간과 검출된 오류개수를 동시에 이용하여 소프트웨어의 최적 출시시점을 구할 수 있는 모형을 제시하였다. 제안된 모형 하에서 소프트웨어는 규정된 수의 오류가 검출되거나 규정된 시험시간의 도래 중 먼저 발생하는 시점까지 시험된 후 출시된다. 비체차 포아송과정을 이용하여 소프트웨어 신뢰도 성장모형을 가정하고, 수명주기 상의 총 비용을 구하는 방법을 제시하였다. 도출된 총비용식으로부터 시험시간만을 규정하거나 오류개수만을 규정하여 출시시점을 정하는 경우를 특별한 경우로 분리해 낼 수 있었다.

제안된 모형은 시험시간과 오류개수를 동시에 고려함으로써 의사결정자에게 좀 더 많은 선택권을 제공해 줄 수 있다는 점에서 기존의 모형들에 비해 유연하고 정교하다고 할 수 있다. 그러나 도출된 총비용 함수식이 결정변수인 N과 T의 복잡한 형태로 표현되어 있어 수학적 분석을 통한 최적화가 매우 어렵다는 점이 단점이라고 하겠다. 이에 대한 부분은 추후 연구과제로 남긴다.

참고문헌

박일광·공명복(2007), 사용단계에서 주기적 서비스 팩 배포와 불확실한 패치 배포를 고려한 소프트웨어의 최적 출시시기, *대한산업공학회지*, 33(4), 487-493.
 유영관(2011), 오류유형을 고려한 최적 소프트웨어 출시정책, *대한설비관리학회지*, 16(1), 41-47.
 이진승·나일용·홍정식이·창훈(2004), 출시 후 보수를 고려한 소프트웨어의 최적 출시시기, *대한산업공학회지*, 30(4), 261-266.
 Bai, D. and Yun, W.(1988), Optimum number of errors corrected before releasing a software system, *IEEE Tr. Reliability*, 37(1), 41-44.
 Boland, P. and Chuiv, N.(2007), Optimal times for software

- release when repair is imperfect, *Statistics and Probability Letters*, 77, 1176-1184.
- Chatterjee, S., Misra, R. and Alam, S.(1997), Joint effect of test effort and learning factor on software reliability and optimal release policy, *International J. of Systems Science*, 28, 391-396.
- Goel, A. and Okumoto, K.(1979), Time-dependent error detection rate model for software reliability and other performance measures, *IEEE Tr. Reliability*, 28, 206-211.
- Hou, R., Kuo, S. and Chang, Y.(1996), Optimal release policy for hyper-geometric distribution software reliability growth model, *IEEE Tr. Reliability*, 45(4), 646-651.
- Huang, C.(2005), Optimal release time for software systems considering cost, testing-effort, and test efficiency, *IEEE Tr. Reliability*, 54, 583- 591.
- Koch, H. and Kubat, P.(1983), Optimal release time of computer software, *IEEE Tr. Software Engineering*, 9(3), 323-327.
- Kumar, S. and Zeepongsekul, P., Balasubramanian, S.(1995), Modeling and analysis of a software system with improvements in the testing phase, *Mathematical and Computer Modelling*, 22, 183-191.
- McCabe, T.(1976), A complexity measure, *IEEE Tr. Software Engineering*, 2(4), 308-320.
- Okumoto, K. and Goel, A.(1980), Optimal release time for software system based on reliability and cost criteria, *J. of System and Software*, 1, 315-318.
- Pham, H.(2000), *Software Reliability*, Springer, Singapore.
- Pham, H. and Zhang, X.(2003), NHPP software reliability and cost models with testing coverage, *European J. of Operational Research*, 145, 443-454.
- Shinohara, Y. and Dohi, T., Osaki, S.(1997), Comparison of optimal release policies for software systems, *Computers & Industrial Engineering*, 33, 813-816.
- Yamada, S. and Osaki, S.(1985), Cost-reliability optimal release policies for software systems, *IEEE Tr. Reliability*, 34(5), 422-424.
- Xie, M. and Hong, G.(1999), Software release time determination based on unbounded NHPP model, *Computers & Industrial Engineering*, 37, 165-168.
- Ross, S.(1983), *Stochastic Processes*, New York; John Wiley & Sons.

A Software Release Policy with Testing Time and the Number of Corrected Errors

Yoo, Young Kwan*

Abstract

In this paper, a software policy considering testing time and the number of errors corrected is presented. The software is tested until a specified testing time or the time to a specified number of errors are corrected, whichever comes first. The model includes the cost of error correction and software testing during the testing time, and the cost of error correction during operation. It is assumed that the length of software life cycle has no bounds, and the error correction follows a non-homogeneous Poisson process. An expression for the total cost under the policy is derived. It is shown that the model includes the previous models as special cases.

Key Words: software release, software error, software failure, life cycle cost.

* Department of Business Administration, Halla University, Associate Professor.