

# 소프트웨어 테스트 정지시각 결정 방법에 대한 연구

최규식\*, 장원석\*, 김종기\*\*

\* 건양대학교 정보전자통신공학부

\*\* 건양대학교 첨단과학부

요약 - 소프트웨어의 신뢰도모델을 적용하는 중요한 일 중의 하나는 소프트웨어 발행시각을 결정하는 일이다. 이 문제에 대해서 여러 연구자들이 그동안 많은 연구논문을 발표하였다. 본 논문에서는 최적발행시각을 결정하는 문제를 심도 있게 다루고자 하였다. 특정한 발행시각을 결정하는데 있어서 신뢰도 요건 및 비용 모델 모두를 고려하여 고출한다. 신뢰도 목표로서 허용치 이내의 고장강도를 사용해야 하며, 최적 발행방책은 순차적인 접근법에 근거하여 검토되어야 한다.

## 1. 서론

소프트웨어를 테스트하고 디버깅하는 최종 목적은 해당 소프트웨어가 요구하는 신뢰도에 이르게 하기 위한 것이다. 소프트웨어의 테스트 단계에서는 언제 테스트를 중단하여 소프트웨어 제품을 판매해야 하는가 라는 것에 대해서 관심이 크며, 따라서, 그 때를 아는 것이 중요한 요소가 될 것이다. 소프트웨어 테스트 기간이 길어지면 길수록 소프트웨어가 좀더 신뢰성 있게 될 것이다. 그러나, 다른 제품들에서와 마찬가지로 소프트웨어 제품을 될 수 있는 한 빨리 판매하는 것이 중요하므로, 테스트 시간을 무작정 길게 할 수는 없다. 그러므로, 연구 분석에 의해 최적발행시각을 결정하는 것이 매우 중요하다.

최적발행시각을 결정하는 것은 결국 최적화 문제로 귀착된다. 이와 관계되는 인자들을 분석하기 위해서 여러 가지 상이한 기준을 도입할 수도 있다. 첫째, 그 요건이 고장수목(fault-tree)이거나 또는 다른 신뢰도목표이면, 신뢰도요건에 이르는 최소시간을 결정하는 일이 문제로 된다. 둘째, 전체 소프트웨어 비용을 고려해야 할 경우, 최적발행시각이 여기에 해당되는 적절한 비용함수에 의해서 결정되어야 한다는 것이다. 그래서, 전체적인 예상 소프트웨어 비용이 최소화되도록 한다.

본 논문에서는 여러 가지 소프트웨어 발행방책에 대해서 고찰을 하였다. 신뢰도 요건과 비용 모델 모두를 이용하여 특수한 최적발행시각을 결정하도록 한다. 그 다음에 최적 발행시각을 결정하는 중요한 방향에 대해서 검토한다.

## 2. 신뢰도 기준에 근거한 최적발행시각

본 장에서는 상이한 기준에 근거를 둔 몇 가지 소프트웨어 발행 방책에 대해서 고찰한다. 신뢰도요건은 허용치 이내의 잔여결함수로 주어지거나 허용치 이내의 고장강도 레벨로 주어질 수 있다. 보통 소프트웨어를 발행할 것인가 또는 테스트를 계속할 것인가를 결정하는 것이 필요하다. 이것이 상이한 여러 소프트웨어 모델에 근거하여 일반적으로 결정론적 문제로 공식화하는 정지규칙문제이다. 모델파라미터가 일반적으로 알려져 있지 않아서 이를 산출해야 하므로 정지 규칙 또한 순서적이어야 한다.

### 2.1 잔여 소프트 결함의 수에 근거한 정지규칙

우리가 소프트웨어를 발행해야 할까, 아니면 테스트를 계속해야 할까를 결정해야 한다고 할 때, 비록 잔여결함의 수가 소프트웨어 신뢰도의 특성적 척도는 아니지만 이에 대한 조사가 필요하다. 소프트웨어의 발행시간은 잔여결함의 수가 규정된 저 레벨에 이르도록 요구하여 쉽게 결정할 수도 있다.

다음과 같은 조건을 만족시키는 처음  $n$ 에 대해서 테스트를 중지시켜야 하는 것이 합리적인 규칙이다.

$$P(\hat{N}_0 > n) \leq \rho_0 \quad (1)$$

$n$  : 현재까지 검출된 결함의 수

$\hat{N}_0$  : 초기결함의 기대수

$\rho_0$  : 목표 위험도

잔여결함  $N_r$ 의 기대치는

$$\hat{N}_r = \hat{N}_0 - n \quad (2)$$

이다. 현실적으로 소프트웨어의 모든 결함을 제거시킬 때까지 테스트를 계속하는 것은 합리적이지 못하다. 확률이란 잔여결함의 수가 목표결함의 수보다 적을 때 테스트를 중단하는 것이다. 또 다른 확률은 잔여결함의 수가 초기결함의 목표비율보다 적을 때 테스트를 중단하는 것이다.

## 2.2 소프트웨어 고장강도 요건에 근거한 정지규칙

소프트웨어 결함의 수가 소프트웨어 신뢰도의 양호한 척도가 아니며, 주로 이론적인 면에서 관심이 있는 분야이기 때문에 최적 소프트웨어 발행시각은 고장강도 함수에 근거하여 결정되어야 한다. 이는 즉 시각  $t$ 에서 소프트웨어를 발행해야 하는가 아닌가를 현재의 고장강도가 허용 저레벨에 이르렀는가에 의해서 결정해야 한다.

$n$ 번째 결함을 제거한 후의 추정 고장강도를  $\hat{\lambda}(n+1)$ 이라 하면 어떤 목표치  $\lambda_0$ 에 대하여

$$\hat{\lambda}(n+1) \leq \lambda_0 \quad (3)$$

일 때 테스트를 중단해야 한다. 한 예로서 고엘과 오꾸모토 모델을 들기로 하자. 이전의 고장 데이터를 이용하여 파라미터를 구하고자 한다면 테스트시간  $t_s$  후의 신뢰도는

$$(t+t_s) = \exp\{-m(t)e^{-bt}\} \quad (4)$$

이고  $m(t)$ 는 평균치 함수로서

$$m(t) = a(1 - e^{-bt}) \quad (5)$$

이다.

상기와 같은 관계로부터 목표신뢰도  $R$ 을 얻기 위한 최적 테스트시간을 쉽게 결정할 수 있다.

$$\begin{aligned} t_s &= \frac{1}{b} \{ \ln m(t) - \ln(-\ln R) \} \\ &= \frac{1}{b} \{ \ln[a(1 - e^{-bt})] - \ln(-\ln R) \} \end{aligned} \quad (6)$$

NHPP 모델이 아닌 다른 평균치 함수를 이용해도 최적발행시각을 계산할 수 있다.

## 2.3 멱수형 마코프모델 적용

우리는 잔여결함의 수에 종속되는 파라미터로

서 멱수형 함수를 일반적으로 사용할 수 있다.

이는 고장간 시간들이 독립이고 파라미터

$$\lambda(i) = \emptyset(N_0 - i + 1)^\alpha \quad (7)$$

를 가진 지수형분포인 것으로 가정할 수 있다.

$N_0$  : 초기고장의 수

$\emptyset$  : 비례상수

$\alpha$  : 테스트품질에 달려 있는 파라미터

$(i-1)$  : 검출되어 제거된 결함의 수

JM모델은  $\alpha=1$ 이다. 그러나, 테스트 초기에 대다수의 결함이 검출되므로  $\alpha > 1$ 이라는 것을 예상할 수 있다. 이러한 일반적인 모델에 이르는 약간의 크기에 편향된 표본 상수를 발견할 수 있다. 예를 들어 결함이 기하학적 분포크기를 가지며, 결함검출확률이 그 크기에 비례한다면  $\alpha=2$ 이다.

## 2.4 순차절차

일반적으로 모델파라미터는 알려지지 않기 때문에 수집 데이터를 이용하여 구하는 수밖에 없다. 때로는 파라미터를 평가하는 자체가 문제가 된다. Forman과 Sin은 JM 모델에 근거한 최적 테스트시간 결정문제를 연구하였다. 그들은 잔여결함의 수를 평가하는 통상의 MLE에 근거한 최적테스트 시간이 적절하지 않다는 사실을 발견하였다. 그들은 이 경우에 좀더 양호한 통계적 특성을 가진 상대가능성함수(relative likelihood function)를 사용하도록 제안하였다.

또 다른 연구논문에서 이들은 통계적 가설이론에 근거한 정지규칙을 제안하였다. 이 가설은 아래와 같다.

$$H_0: N^* = 0, \text{ vs } H_1: N^* > 0 \quad (8)$$

이 절차는 추가시간  $t_s$  동안 소프트웨어를 테스트하고 고장이 발생되면  $H_0$ 를 거절하는 것이다. 가설  $N=n$ 과  $N > n$  사이의 최적테스트 시간은 후속 추가시간  $T(n+1)$ 동안 그 이상의 고장이 검출되지 않으면 테스트를 중단하는 것이다. 그렇지 않으면 테스트를 계속하여  $T(n+2)$ 시간이 된다.

일반적으로 최적발행방책은 다음과 같은 접근

법에 의한다.

1. 안정된 산출결과에 필요한 시간인 목표시간  $T_p$  동안 소프트웨어를 테스트한다.
2. 모델파라미터를 산출한다.
3. 전에 검토한 절차를 이용하여 최적 테스트시간  $T$ 를 결정한다.
4.  $T=0$ 이면 소프트웨어를 발행하고 그렇지 않으면 또 다른 시간 단위  $T$  동안 테스트를 계속하여 단계 2로 복귀한다.

### 3. 비용기준에 근거한 최적발행시각

실제에 있어서 소프트웨어 매니저는 비용의 입장에서 본 최적 발행시각을 결정하는 일에 비중을 크게 두는 것이 일반적이다. 전 수명기간 동안 소프트웨어의 예상비용은 극히 필수적인 과제이다.

고엘과 오푸모토가 제안한 모델은 단순한 비용함수에 근거를 두었으며, 후에 여러 연구자에 의해서 다각도로 연구·검토되었다. 기준이 되는 소프트웨어 신뢰도 모델은 단순한 평균치 함수를 가진 NHPP 모델이다.

평균치함수를  $m(t)$ 라 하면 비용함수는

$$C(T) = c_1 m(T) + c_2 [m(\infty) - m(T)] + c_3 T \quad (9)$$

로 정의한다.

$c_1$  : 테스트기간중 결함을 고치는 기대 비용

$c_2$  : 운전기간중 결함을 고치는 기대 비용

$c_3$  : 소프트웨어 테스트 기간중의 테스트 비용과 발행지연 비용을 포함한 소프트웨어 테스트의 기대 비용

이러한 발행방책을 일반화시키는 방법으로 두 가지가 있다. 다른 소프트웨어 신뢰도 모델들은 위에서 제시된 비용함수에 쉽게 적용할 수 있다. 기본이 되는 소프트웨어 모델로서 다른 NHPP 모델은 다른 연구에서 발견할 수 있다. 또 다른 방법은 단순 비용함수를 일반화 시키는 연구가 있다. 이러한 두 가지 경우를 하나로 결합하는 것도 가능하다.

### 4. 기타 다른 최적 발행 방책

#### 4.1 주어진 목표신뢰도에서 기대 비용 최소화

일반적으로 최적 발행 방책은 비용과 신뢰도

를 동시에 고려하여 결정된다. 상기 두 항을 결합하여 주어진 신뢰도 요건에서 전체 기대 비용을 최소화하거나 주어진 총비용 하에서 목표신뢰도를 최대화하여 최적 발행시각을 얻는다.

실제에 있어서 보편적인 상황은 목표 신뢰도가 설정되고 소프트웨어를 발행하기 전에 신뢰도 요건이 달성되는 것이다. 이러한 목표 신뢰도하에 총 기대 비용을 최소화시켜 최적발행시각을 결정하는 것이다. 다음과 같은 최적화 문제를 풀어서 발행시각  $T$ 를 구한다.

$R(x;T) \geq R_0$  하에서

$$EC(T) \text{ 최소화, } T \geq 0 \quad (10)$$

$C(T)$ 를 볼록함수로 가정하고, 또  $T$ 를 변수로 했을 때,  $R(x;T)$ 는 증가함수이므로 우선  $T$ 에 관하여 기대비용  $EC(T)$ 를 최소화하여  $T_0$ 를 얻는다. 그 다음 신뢰도 요건  $R_0$ 에 이르기 위한 최소시간을 계산하여  $T_1$ 을 얻는다. 그러므로, 최적발행시각은

$$T = \text{Min}(0, T_0, T_1) \quad (11)$$

으로 결정한다. 그러한 발행방책은 여러 연구자들에 의해서 연구되었다.

### 4.2 모듈 구조를 가진 소프트웨어의 발행방책

모듈 구조를 가진 소프트웨어의 최적 발행시각을 결정하는데 있어서 마수다는 통계적인 접근을 이용하는 흥미 있는 모델을 제안하였다. 어떤 일반적인 가정하에 또 다른 기간동안 더 테스트를 계속할 필요가 있는지 결정하는 것이 필요하다. 보통 소프트웨어는 여러 모듈로 구성된다. 그러므로, 소프트웨어의 발행시각은 소프트웨어의 모듈구조와 각 모듈의 신뢰도에 의해서 결정되어야 한다.

### 4.3 소프트웨어 패키지 버전의 발행시각

실제에 있어서 동일 소프트웨어가 여러 번에 걸쳐서 상이한 버전으로 발행되는 것이 일반적인 상황이다. 소프트웨어의 전 수명기간에 걸쳐서 동일 소프트웨어의 새로운 버전이 발행되는 것이 하나의 공정이라 할 수 있다. 최근에는 소

소프트웨어 고장 공정에 대한 고엘과 오키모토 모델을 이용한 새로운 소프트웨어 버전의 발행시각을 결정하는 모델을 제안하였다. 최근 버전은 이전 버전에서 검출된 결함을 제거하여 발행하는 것으로 가정한다. 그러한 상황을 기술하고 소프트웨어 패키지의 발행 방식을 결정하는 기법을 제공하기 위한 모델을 가지는 것이 유용하다.

## 5. 결론

본 연구에서는 여러 가지 소프트웨어의 발행 문제를 논하였다. 최적 발행시각은 신뢰도 요건에 이르거나 소프트웨어의 기대비용을 최소화시킴으로써 결정할 수 있다. 여러 가지 상이한 소프트웨어의 신뢰도 모델을 적용할 수 있다.

신뢰도에 있어서 중요한 특성 척도는 가능한 널리 쓰이는 고장 강도함수이다. 잔여 결함의 수는 주로 통계적인 관심사이다. 고장 강도가 목표치에 만족스럽게 이르지 못하면 소프트웨어 테스트를 계속 해야 한다.

일반적으로 신뢰도 모델과 비용 함수에 있는 그러한 여러 가지 파라미터들은 수집 데이터에 의해서 평가한다. 평가는 때때로 부정확하고 따라서, 새로운 고장 데이터가 수집되면 이를 수정한다. 파라미터에 의해서 새로운 평가를 하여 발행시각을 결정한다. 그러므로, 최적발행 방식은 순차적인 접근법에 근거하여 절차로서 공식화되어야 한다.

본 연구는 한국과학재단 목적기초연구(2000-2-30300-001-2) 지원으로 수행되었음
--

### 참고문헌

1. I. K. Okumoto, A.L. Goel, "optimum release time for software systems based on reliability and cost criteria", J. systems and software, vol.1, pp315-318,1980
2. M.Xie, "on the optimum software testing time based on a general software reliability model", research report, LiTH-IKP-R-487, Sweden, 1986
3. S. Yamada, et al., "optimum release

policies for a software system with a scheduled software delivery time", Int.J. systems sci..., vol.15, pp905-914, 1984

-----  
제목 : 소프트웨어 테스트 정지시각 결정 방법에 대한 연구

저자 : 최규식\*, 장원석\*, 김종기\*\*  
\* 건양대학교 정보전자통신공학부

\*\* 건양대학교 첨단과학부

소속기관 : 건양대학교

전화번호 : 041-730-5283

FAX : 041-736-4079

메일 : che@konyang.ac.kr