

# 웨이블 시험노력을 이용한 개발 소프트웨어의 최적발행 모델에 관한 연구

최 규 식<sup>†</sup> · 장 윤 승<sup>†</sup>

## 요 약

본 논문에서는 소프트웨어 시험 단계중에 발생하는 시험노력 소요량을 고려한 소프트웨어 신뢰도 성장 모델을 제시하여 시간종속적인 시험 노력소요량 동태를 웨이블 곡선으로 설명한다. 시험 단계중에 소요되는 시험노력의 양에 대한 결합 검출비를 현재의 결합 내용에 비례하는 것으로 가정하여 소프트웨어 신뢰도 성장 모델을 비동차 포아송 프로세스(NHPP)로 공식화하되, 이 모델을 이용하여 소프트웨어 신뢰도 척도에 대한 데이터 분석기법을 개발한다. 시험 시간의 경과와 신뢰도와의 관계, 시험비용과 신뢰도와의 관계를 연구한다. 소프트웨어의 비용을 고찰함에 있어서 조건별로 검토하여 비용이 최소로 되는 발행시각을 결정하되, 목표신뢰도를 만족시키는 최적발행시각을 정한다. 비용의 입장에서 발행 시각을 결정하는 문제와 신뢰도의 입장에서 발행 시각을 결정하는 문제를 동시에 고려하여 최적 발행시각을 결정하도록 한다.

## A Study on the Optimum Release Model of a Developed Software with Weibull Testing Efforts

Che Gyu Shik<sup>†</sup> · Chang Yun Seung<sup>†</sup>

## ABSTRACT

We propose a software-reliability growth model incorporating the amount of testing effort expended during the software testing phase. The time-dependent behavior of testing effort expenditures is described by a Weibull curve. Assuming that the error detection rate to the amount of testing effort spent during the testing phase is proportional to the current error content, a software-reliability growth model is formulated by a nonhomogeneous Poisson process. Using this model the method of data analysis for software reliability measurement is developed. After defining a software reliability, we discuss the relations between testing time and reliability and between duration following failure fixing and reliability are studied in this paper. The release time making the testing cost to be minimum is determined through studying the cost for each condition. Also, the release time is determined depending on the conditions of the specified reliability. The optimum release time is determined by simultaneously studying optimum release time issue that determines both the cost related time and the specified reliability related time.

키워드 : 소프트웨어 신뢰도 성장 모델(SRGM), NHPP, 평균치 함수(mean value function), 웨이블 시험노력 함수(Weibull testing efforts function, target reliability), 목표 신뢰도, 최적 발행시각( optimum release time)

### 1. 서 론

소프트웨어 개발 과정에서 품질 좋고 신뢰성 있는 소프트웨어를 효과적이고 효율성 있게 개발하기 위하여 소프트웨어의 신뢰도 측정 및 관리가 중요하다. 제품신뢰도를 특성화하기 위해서는 정량적인 측정과 관리가 필수적이며, 특히, 소프트웨어 결함데이터 분석에 근거하여 소프트웨어 시험 단계에서 소프트웨어의 신뢰도를 추정하는 것이 매우 중요하다. 그동안 시험 단계동안 소프트웨어 결함 검출현상을 설명하기 위한 여러 가지 소프트웨어 신뢰도 모델이 개발되었다. 소프

트웨어 시험에 의해서 발견되는 누적결함(또는 소프트웨어 고장간의 시간간격)과 소프트웨어 시험 시간간격 사이의 관계를 짓는 모델들을 소프트웨어 신뢰도 성장모델(SRGM ; software reliability growth model)[1]이라 한다. 이러한 모델들을 이용하여 평균 초기결함의 수, 평균 고장간 시간 간격, 임의의 시험 시간에 소프트웨어 내의 평균 잔여결함수, 소프트웨어 신뢰도 함수와 같은 소프트웨어 신뢰도 척도를 추정할 수 있다.

소프트웨어 개발에는 많은 개발자원이 소요된다. 소프트웨어 시험 단계 기간동안에는 소프트웨어의 신뢰도가 내재 결함을 검출 및 수정하는데 소요되는 개발자원의 양에 크게 의존한다. Musa 등[2]은 기존 소프트웨어 신뢰도 성장모델을 분류하는 하나의 안을 개발하였다. Yamada 등[3]은 역일 시험 시간, 시험 노력량, 시험 노력에 의해서 검출되는 소프트웨어 결

※ 본 연구는 한국과학기술연구원(R01-2000-00273) 지원으로 수행되었음.  
† 정 회 원 : 건양대학교 정보전자통신공학부 교수  
논문접수 : 2001년 2월 26일, 심사완료 : 2001년 9월 24일

함의수 사이의 관계를 명시적으로 설명할 수 있는 간단하고도 새로운 모델을 제시하였다. 시험 노력은 시험 단계에서 소요되는 인력, CPU시간, 실행시험 케이스 등등에 의해서 측정된다.

본 논문에서는 웨이블곡선으로서 시험노력의 시간중속 거동을 설명한다. 소프트웨어 시험에서 결함 검출비가 현재의 내재 결함수에 비례하고, 임의의 시험 시간에서 그 비례가 현재의 시험 노력 여하에 달려 있다는 것을 가정하여 비동차 포아송 과정(Non-Homogeneous Poisson Process ; NHPP)[4]에 바탕을 둔 신뢰도성장모델을 개발한다.

신뢰도와 비용을 고려한 연구로서 Okumoto와 Goel[5]은 전체평균 소프트웨어 비용을 최소화시키는 비용 - 최적 SRP(software reliability process)를 발표하였다. Yamada와 Osaki[6]는 전체 평균 비용을 최소화시키고 소프트웨어 신뢰도를 만족시키는 전체평균비용 - 신뢰도 · 최적 SRP를 도입하였다. 이러한 연구결과를 참조하여 Hou, Kuo, Chang[7]은 지수 곡선과 로지스틱 곡선에 적용하는 연구를 수행하였다. 소프트웨어공학에서 신뢰도 분야는 아직까지 큰 비중을 차지하고 있지 못하지만, 앞으로 소프트웨어 산업의 중요성이 더욱 높아진다고 전망해볼 때, 최소의 비용으로 개발 소프트웨어의 신뢰도를 극대화할 수 있는 기법이 계속 연구된다면 소프트웨어의 비용을 크게 줄일 수 있을 것으로 본다.

본 논문에서는 테스트노력으로서 웨이블곡선을 이용하여 비용과 신뢰도에 대한 두 개의 기준을 동시에 고려하여 발행시기를 결정하는 최적발행정책에 대해서 연구한다. 연구방법으로서 소프트웨어 신뢰도를 고려하여 목표신뢰도를 만족시키면서 발행시기를 결정하는 방법과 총비용이 최저로 되는 발행시점을 결정하는 방법을 동시에 연구하여 최적발행시기를 어떻게 결정하는 것이 합리적인가를 고찰한다.

<표 1> 사용되는 기호 및 의미

$N(t)$ : 시각 $t$ 까지 검출되는 소프트웨어의 누적결함 갯수
$a$ : 초기부터 소프트웨어내에 존재하고 있는 결함의 갯수
$b, b_1, b_2$ : 일반적인 경우, 시험시간중, 발행후 운영시간중의 결함 검출비, $b_1 \geq b_2$
$m(t)$ : $E[N(t)]$ , 평균치 함수
$R(x t)$ : 소프트웨어의 신뢰도. 시각 $t(x \geq 0)$ 에서 결함이 검출된 후 $(t, t+x)$ 에서 고장이 일어나지 않을 확률
$R_0$ : 목표신뢰도, $0 < R_0 < 1$
$c_1$ : 시험시간중에 검출되는 결함을 수정하는 단위비용, $c_1 > 0$
$c_2$ : 운영시간에 검출되는 결함을 수정하는 단위비용, $c_1 > c_2$
$c_3$ : 단위시간당 시험비용, $c_3 > 0$
$C(T)$ : 전체 평균 소프트웨어 비용
$T_{LC}$ : 소프트웨어의 수명주기
$T$ : 전체 시험 시간
$T^*$ : 최적 소프트웨어 시험시간
$T_1$ : $dC(T)/dT=0$ 을 만족시키는 유일 해 $T$
$T_2$ : $R(x T)=R_0$ 를 만족시키는 유일 해 $T$
$d$ : $\exp[-\alpha\gamma(1 - e^{-\beta T T_2})]$ , $0 < d < 1$
$\alpha$ : 소프트웨어 시험에서 필요로 하는 시험노력 소요 총량
$\beta, m$ : 척도모수, 형상모수

2절에서는 NHPP를 기준으로 한 소프트웨어 신뢰도 성장 모델에 대해서 각종 테스트 노력 곡선에 대해서 간략히 검토한 후 고장확률 및 신뢰도를 연구하고, 3절에서는 시험 시간중 및 발행운영 시간중에 검출되는 결함의 수정비용과 시험비를 연구한다. 4절에서는 여러 조건을 고려하여 각 조건에 맞는 비용-신뢰도 최적발행정책을 검토하여 연구한다. 본 논문에서 사용된 용어의 의미를 <표 1>에 정리하였다.

## 2. 시험 노력 함수

주어진 기간에 맞추어 소프트웨어 시스템을 개발할 때 시간, 자금, 인력과 같은 자원들이 소요된다. 일반적으로 이러한 소요자원의 시간중속 동태를 예측할 때 레일레이 곡선을 사용해왔다. 특히, 소프트웨어 시험에까지 이르는 자원들이 소프트웨어 신뢰도에 상당한 영향을 미친다. 소프트웨어 개발 자원 전체의 약 40~50%가 시험단계에서 소요된다. Yamada 등[3]은 시험 기간중의 시험노력과 소프트웨어 개발 노력 모두가 레일레이 곡선으로 설명될 수 있다는 것을 가정하여 소프트웨어 시험에 쓰이는 시험노력의 양을 고려한 소프트웨어 신뢰도 성장 모델을 제시하였다. 그들은 레일레이 곡선의 대안으로서 지수곡선도 제안하였다.

### 2.1 레일레이 곡선

레일레이 곡선은 식 (1)로 표현되며,

$$w_R(t) = a\beta \cdot t \cdot \exp\left[-\frac{\beta}{2} t^2\right] \tag{1}$$

로 표시되며, 그 적분 형태는 식 (2)와 같다.

$$W_R(t) = a\left(1 - \exp\left[-\frac{\beta}{2} t^2\right]\right) \tag{2}$$

### 2.2 지수 곡선

$$w_R(t) = a\beta \cdot \exp[-\beta t] \tag{3}$$

로 표시되며, 그 적분 형태는 다음과 같다.

$$W_R(t) = a(1 - \exp[-\beta t]) \tag{4}$$

본 논문에서는 시각  $t$ 에서의 시험노력 형상을 기술하기 위해 시험노력함수로서 웨이블곡선을 이용한다.

$$w(t) = a \cdot \beta \cdot m \cdot t^{m-1} \cdot \exp[-\beta t^m] \tag{5}$$

(5)의 적분형태

$$W(t) = a(1 - \exp[-\beta t^m]) \tag{6}$$

는 시각  $(0, t)$ 에서의 누적 시험 노력량을 나타낸다.

웨이블 시험 노력함수 (1)이나 (2)에서의  $a, \beta, m$ 은 최소 자승법으로 구한다.  $a, \beta, m$ 은  $(t_k, w_k)$ 형태의  $n$ 개 관찰 데이터 쌍으로부터 결정한다.

### 3. 소프트웨어 신뢰도 성장 모델

#### 3.1 모델에 대한 고찰

소프트웨어 개발 시험 기간동안 시스템 내의 결함을 검출하여 수정한다. 소프트웨어 고장은 시스템 내에 잔존하고 있는 소프트웨어의 결함에 의해서 프로그램 동작이 제대로 되지 않는 것을 말한다. 평균치 함수를 다음과 같이 정의한다.

$$m(t) = a(1 - \exp[-rW(t)]) \quad (7)$$

소프트웨어 결함검출 현상을 통계적으로 모델링할 때에 식 (7)의  $m(t)$ 에 의한 NHPP에 근거하여  $N(t)$ 의 평균치함수를 정의하면 웨이블 시험노력함수를 고려한 소프트웨어 신뢰도 성장모델을 만들 수 있다.

$$\Pr\{N(t) = n\} = \text{poim}(n; m(t)) \quad (8)$$

NHPP 고장강도함수는 평균치 함수의 미분 형태이다.

$$\lambda(t) = dm(t)/dt = a \cdot r \cdot w(t) \cdot \exp[-rW(t)] \quad (9)$$

식 (8)로부터  $N(t)$ 의 제한적인 분포가 평균치 함수

$$m(\infty) = a(1 - \exp[-ra]) \quad (10)$$

을 가진 포아송분포라는 것을 보여주고 있다.

#### 3.2 소프트웨어 신뢰도 척도

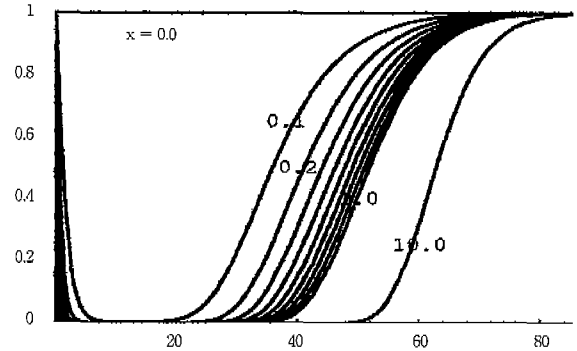
식 (7)의  $m(t)$ 를 가진 NHPP 모델에 의해서 소프트웨어 신뢰도 평가에 대한 두 가지 정량적 평가 척도를 얻을 수 있다. 시각  $t$ 에서의 신뢰도는 다음과 같다.

$$R(x|t) = \exp\{-a(\exp[-r \cdot W(t)] - \exp[-r \cdot W(t+x)])\} \quad (11)$$

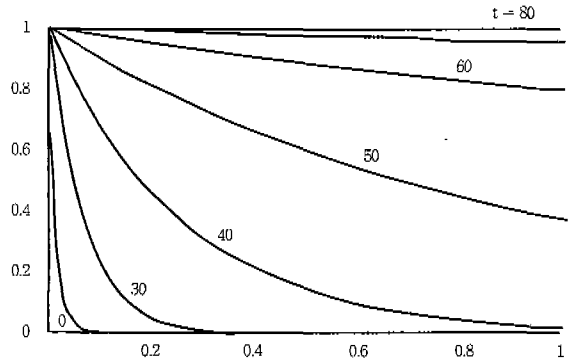
식 (11)로 표시된 신뢰도의 특성을 이해하기 위해 시험시간과 신뢰도의 관계 및 최종 검출 결함 수정 후 경과시간과 신뢰도의 관계를 그림으로 나타내면 (그림 1), (그림 2)와 같다.

(그림 1)의 경우, 최종 결함 수정 후 경과되는 각각의 시간에 대해서 발행시각과 신뢰도 성장과의 관계를 보여주고 있다. 일정한 경과시간  $x$ 에 대해서 시험시간 및 발행시기를 늦추면 늦출수록 신뢰도가 성장함을 알 수 있다. 또한, 비록 신뢰도가 성장하여 목표신뢰도 이상이 될 수 있으나, 결함 수정 후 경과시간이 길어지면 길어질수록 결함 발견 확률이 높아 신뢰도가 저하된다는 것도 알 수 있다. 그림에서  $x = 0.0$  일 때는 시험 시간에 관계없이 신뢰도가 1이나,  $x$ 의 값이 커

지면 커질수록 곡선이 횡축의 우측으로 이동하여 신뢰도가 저하됨을 알 수 있다. 결함 수정 후 경과시간을 어떤 범위로 하여 신뢰도 성장 기준을 잡는가도 중요한 문제이다.



(그림 1) 발행시각과 신뢰도와의 관계



(그림 2) 결함 수정후 경과시간과 신뢰도와의 관계

이와는 대조적으로 (그림 2)의 경우, 주어진 각각의 시험 시간  $t$ 에 대해서 최종 결함 수정 후 경과시간  $x$ 와 신뢰도 성장과의 관계를 보여주고 있다. 일정한 시험시간  $t$ 에 대해서 경과시간  $x$ 가 작으면 작을수록 신뢰도가 높으며, 경과시간  $x$ 가 증가함에 따라 신뢰도가 급격히 감소된다. 이 그림에서 보듯 시험 시간이 길면 길수록 경과시간에 대한 신뢰도의 저하가 작아짐을 알 수 있다.

상기 식 (11)에서 정의한 시험 단계의 신뢰도 의미를 고찰해보기로 한다.

$R(x|t)$ 는 시각  $t$ 에서 최종적으로 결함을 발견하여 수정한 후  $x$ 단위시간 동안 새로운 결함이 발견되지 않을 확률이다. 소프트웨어를 개발하여 결함시험을 하면 할수록 결함을 발견하여 수정하는 빈도가 작아지므로 신뢰도가 성장되며, 결함 수정 후 경과시간이 길어지면 길수록 결함 발견 확률이 높아지기 때문에 소프트웨어의 신뢰도는 낮아진다. 한편, 시험 단계에서는 얼마나 오랜 시간동안 결함이 발견되지 않느냐가 중요한 것이 아니라, 현 단계에서 소프트웨어 내에서 발견되지 않고 잔존하는 결함의 수가 얼마나 되는가가 더 중요하다.

소프트웨어를  $t = T$ 에서 발행하는 경우, 발견되는 누적

분포는

$$m(T) = a(1 - e^{-\gamma W(T)})$$

이므로, 잔여 분포

$$\bar{a} = a - a(1 - e^{-\gamma W(T)}) = a \cdot e^{-\gamma W(T)}$$

이다. 이것이 소프트웨어를 발행해서 운전하는 초기 결함수  
이므로

$$\begin{aligned} m(T+x) &= a \cdot e^{-\gamma W(T)} \cdot (1 - e^{-\gamma W(x)}) + m(T) \\ &= a(1 - e^{-\gamma W(T) - \gamma W(x)}) \end{aligned} \quad (12)$$

이다. 그러므로,

$$\begin{aligned} R(x|T) &= \exp[-m(T+x) + m(T)] \\ &= \exp[-a e^{-\gamma W(T)} (1 - e^{-\gamma W(x)})] \\ &= \exp\{-a e^{-a\gamma(1 - e^{-\beta T^*})} [1 - e^{-a\gamma(1 - e^{-\beta x^*})}]\} \end{aligned} \quad (13)$$

이고, 여기서

$$R(x|0) = \exp\{-a [1 - e^{-a\gamma(1 - e^{-\beta x^*})}]\} \quad (14)$$

이다.

따라서, 시각 T에서 발행된 소프트웨어의 신뢰도는 경과  
시간 x에 대해서 지수함수적으로 감소한다. 감소비를 줄이기  
위해서는 발행시각을 늦추거나 결함검출비를 높여야 한다.

#### 4. 비용-신뢰도 최적 발행 정책

소프트웨어 프로젝트 관리자에게 있어서는 소프트웨어 시  
험을 중단하고 시스템을 사용자에게 인계하는 최적발행시각  
을 결정하는 것이 중요하다. 이것을 최적 소프트웨어 발행문  
제라고 부른다. 이는 시스템의 신뢰도와 투입된 시험자원 사  
이의 관계를 고려하여 공식화할 수 있다. 목표신뢰도를 맞추  
면서 전체 소프트웨어 비용이 최소가 되도록 하는 것이 중  
요하다. 시험 단계중의 시험 노력 소요량 비용과 발행 전후  
의 결함수정비용을 소프트웨어 비용인자로 계산한다. (그림  
3)에서는 시험 기간을 횡축으로 하여 비용과 신뢰도의 관계  
를 동시에 나타낸 것이다. 아울러 최적 소프트웨어 발행문제  
를 다음과 같이 정의한다.

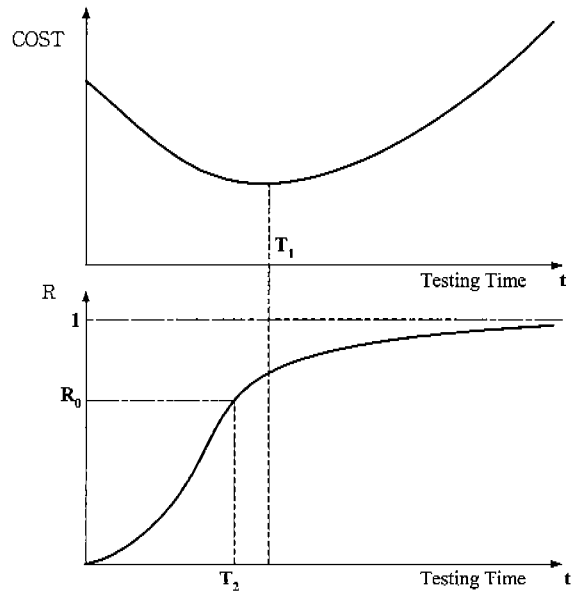
$$c_2 > 0, c_1 > c_3 > 0, x \geq 0, 0 < R_0 < 1 \text{인 경우에 대해서}$$

$$R(x|T) \geq R_0, T \geq 0 \text{인 조건하에 } C(T) \text{를 최소화} \quad (15)$$

이러한 방법으로 하여 비용-신뢰도 최적 소프트웨어 발행시  
각에 대한 해를 구할 수 있다.

$$T^* = \max\{T_1, T_2\} \quad (16)$$

여기서,  $T_1$ 은 비용을 최저로 하는 발행시간,  $T_2$ 는 목표신뢰  
도를 만족하는 발행시간이다. 비용을 될 수 있는 한 적게 들  
이고 목표 신뢰도를 맞추기 위한 최적 발행시각은 상기 두  
시간 중에 큰 값으로 결정되는 경우이다. 비용을 최저로 하  
면서 목표신뢰도를 맞추기 위한 최적발행시각은 그 두 시간  
중에 큰 값을 취해야 한다는 것을 의미한다.



(그림 3) 비용-신뢰도 곡선

#### 4.1 비용을 고려한 발행시간

총 소프트웨어비용을 아래와 같이 정의한다.

$$\begin{aligned} C(T) &= c_1 \cdot m(T) + c_2\{m(T_{LC}) - m(T)\} \\ &\quad + c_3 \int_0^T w(x) dx \end{aligned} \quad (17)$$

즉, 테스트 기간중에 발생하는 결함을 수정하는 비용과 운영  
기간중에 발생하는 결함을 수정하는 비용, 그리고 테스트 기  
간동안의 테스트 비용을 합친 비용이 소프트웨어 개발 및  
운영에 소요되는 총비용이다.

식 (17)을 다시 쓰면 아래와 같다.

$$\begin{aligned} C(T) &= c_1 a(1 - e^{-\gamma W(T)}) + c_2 \{a(e^{-\gamma W(T)} \\ &\quad - e^{-\gamma W(T_{LC})})\} + c_3 \int_0^T w(x) dx \end{aligned} \quad (18)$$

식 (18)을 T로 미분하여 0으로 놓는다.

$$\begin{aligned} \frac{\partial C(T)}{\partial T} &= w(T)(c_2 - c_1) \left\{ \frac{c_3}{c_2 - c_1} - a\gamma \cdot \exp(-\gamma W(T)) \right\} \\ &= w(T)(c_2 - c_1) \left\{ \frac{c_3}{c_2 - c_1} - a\gamma \cdot \exp[-a\gamma(1 - e^{-\beta T^*})] \right\} \end{aligned} \quad (19)$$

식 (19)를 0으로 하여 비용을 최저로 하는 발행시각을  $T_1$

라 하면

$$T_1 = \left\{ -\frac{1}{\beta} \ln \left[ 1 + \frac{1}{a\gamma} \ln \frac{c_3}{a\gamma(c_2 - c_1)} \right] \right\}^{\frac{1}{m}} \quad (20)$$

이므로

$$0 < \left\{ -\frac{1}{\beta} \ln \left[ 1 + \frac{1}{a\gamma} \ln \frac{c_3}{a\gamma(c_2 - c_1)} \right] \right\}^{\frac{1}{m}} < T_{LC}$$

인 범위 즉,

$$1) \quad 1 > \frac{c_3}{a\gamma(c_2 - c_1)} > d$$

에서 양의 유일 해  $T^* = T_1$  이 존재한다.

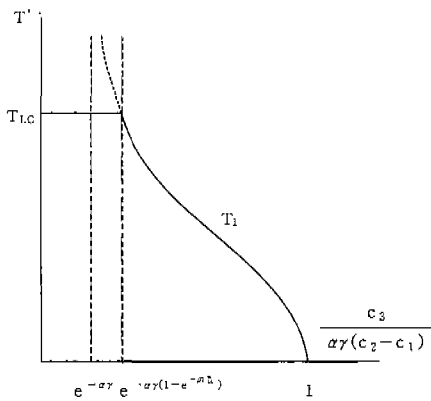
$$2) \quad \frac{c_3}{a\gamma(c_2 - c_1)} \geq 1$$

이면  $T_1 \leq 0$ 이므로  $T^* = T_1 = 0$ 이다.

$$3) \quad \frac{c_3}{a\gamma(c_2 - c_1)} \leq d$$

이면  $T_1 \geq T_{LC}$ 이므로  $T^* = T_1 = T_{LC}$ 이다.

이와 같은 내용을 (그림 4)에 표시하였다.



(그림 4) 비용과 발행시각 곡선

즉,  $\frac{c_3}{a\gamma(c_2 - c_1)} > 1$ 인 조건은 비용이 단조증가하는 경

우로서 결함시험을 하면 할수록 비용이 증가되어 시험 없이 발행하는 것이 최적인 것을 의미한다.

$1 \geq \frac{c_3}{a\gamma(c_2 - c_1)} > d$ 인 조건은 비용최저점이 결함시험

과 소프트웨어의 전 수명기간 사이에 존재하는 경우이다. 이러한 경우는 본 논문이 추구하고자 하는 이상적인 경우로서 목표신뢰도를 만족시키는 발행시기와 총 비용을 최저로 하는 발행시기 중 큰 값을 취하는 것이 이상적이다.

$\frac{c_3}{a\gamma(c_2 - c_1)} \leq d$ 인 조건은 비용이 단조감소하는 경우

이다. 이러한 경우는 결함시험을 하면 할수록 총 비용이 감소된다. 이러한 경우는 소프트웨어의 결함시험에 의해서 신뢰도가 높아지고, 따라서 드물게 결함이 검출되어도 그 수정 비용이 미미한 경우에 해당된다.

#### 4.2 목표신뢰도를 고려한 발행시각

마찬가지로, 목표 신뢰도에 가장 근접하는 유일한 시각이 존재한다. 최적 소프트웨어 발행시각은 미리 규정된 소프트웨어 목표신뢰도에 가장 근접하는 시각이다. 발행시각  $T$ 에서의 신뢰도는 식 (13)과 같으므로 여기서 목표신뢰도를 만족시키는 발행시각을 구하면 다음과 같다.

목표신뢰도를  $R_0$ 라 하면 식 (13)에서

$$\exp\{-ae^{-a\gamma(1-e^{-a\gamma T^*})}[1 - e^{-a\gamma(1-e^{-a\gamma T^*})}]\} = R_0 \quad (21)$$

이다.

식 (21)을 만족하는 발행시각을  $T_2$ 라 하면

$$T_2 = \left\{ -\frac{1}{\beta} \ln \left[ 1 + \frac{1}{a\gamma} \ln \frac{\ln R_0}{\ln R(x|0)} \right] \right\}^{\frac{1}{m}} \quad (22)$$

이므로

$$0 < \left\{ -\frac{1}{\beta} \ln \left[ 1 + \frac{1}{a\gamma} \ln \frac{\ln R_0}{\ln R(x|0)} \right] \right\}^{\frac{1}{m}} < T_{LC}$$

인 범위 즉,

$$1) \quad R_0 > R(x|0) > R_0^{\frac{1}{d}}$$

에서 양의 유일 해  $T^* = T_2$ 이 존재한다.

$$2) \quad R(x|0) \geq R_0$$

이면  $T_2 \leq 0$ 이므로  $T^* = T_2 = 0$ 이다.

$$3) \quad R(x|0) < R_0^{\frac{1}{d}}$$

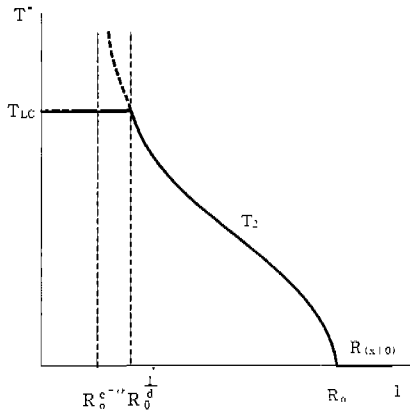
이면  $T_2 \geq T_{LC}$ 이므로  $T^* = T_2 = T_{LC}$ 이다.

이와 같은 내용을 (그림 5)에 표시하였다.

즉,  $R(x|0) \geq R_0$ 는 소프트웨어 개발 즉시 신뢰도가 목표 신뢰도를 만족하기 때문에 더 이상 결함 발견을 위한 시험을 할 필요가 없는 경우로서 비용을 최저로 하는 시기에 맞추어 발행시기를 결정해야 하는 경우이다.

$R_0 > R(x|0) > R_0^{\frac{1}{d}}$ 는 소프트웨어를 개발한 후 시험 및 결함 수정을 통하여 목표신뢰도를 만족시키는 경우이다. 이러한 경우는 본 논문이 추구하고자 하는 이상적인 경우로서 목표신뢰도를 만족시키는 발행시기와 총 비용을 최저로 하는 발행시기 중 큰 값을 취하는 것이 이상적이다.

$R(x|0) < R_0^{\frac{1}{d}}$ 는 소프트웨어의 전 수명기간에 걸쳐서 시험을 해도 목표신뢰도를 만족시키지 못하는 경우이다.



(그림 5) 목표신뢰도와 발행시각 곡선

4.3 최적 발행 시각

이상의 내용을 요약하면 다음 <표 2>과 같다.

4.3.1 비용

이 표를 비용면에서 고찰해보기로 한다.

1)  $\frac{c_3}{a\gamma(c_2 - c_1)} > 1$

이면 비용이 단조증가하는 경우로서 결함시험을 하면 할수록 비용이 증가되어 시험 없이 발행하는 것이 최적인 것을 의미한다. 그러나, 이러한 경우는 현실적으로 고려하기 어려운 경우이다.

2)  $1 \geq \frac{c_3}{a\gamma(c_2 - c_1)} > d$

는 비용최저점이 결함시험과 소프트웨어의 전 수명기간 사이에 존재하는 경우이다. 이러한 경우는 목표신뢰도를 만족시키는 발행시각과 총 비용을 최저로 하는 발행시각 중 큰 값을 취하는 것이 이상적이다.

3)  $\frac{c_3}{a\gamma(c_2 - c_1)} \leq d$

는 비용이 단조감소하는 경우이다. 이러한 경우는 결함시험을 하면 할수록 총 비용이 감소되는 경우이다.

4.3.2 신뢰도

다참가지로 신뢰도면에서 고찰해보기로 한다.  $R(x|0)$ 은 소프트웨어를 개발하여 시험을 거치지 않은 상태에서  $x$  시간까지 시간이 경과할 때의 소프트웨어 신뢰도를 나타내는 것이다.

1)  $R(x|0) \geq R_0$ .

는 소프트웨어 개발 즉시 신뢰도가 목표신뢰도를 만족하기 때문에 더 이상 결함 발견을 위한 시험을 할 필요가 없는 경우로서 비용을 최저로 하는 시기에 맞추어 발행시각을 결정해야 하는 경우이다. 그러나, 이러한 경우는 고려하기 어려운 경우이다.

2)  $R_0 > R(x|0) > R_0^{\frac{1}{d}}$

는 소프트웨어를 개발한 후 시험 및 결함 수정을 통하여 목표신뢰도를 만족시키는 경우이다. 이러한 경우는 목표신뢰도를 만족시키는 발행시각과 총 비용을 최저로 하는 발행시각 중 큰 값을 취하는 것이 이상적이다.

3)  $R(x|0) < R_0^{\frac{1}{d}}$

는 소프트웨어의 전 수명기간에 걸쳐서 시험을 해도 목표신뢰도를 만족시키지 못하는 경우로서 소프트웨어 개발에 실패한 경우로 볼 수 있다.

따라서, 이 중에서 가장 이상적인 범위는 비용면에서

$1 \geq \frac{c_3}{a\gamma(c_2 - c_1)} > d$

이고 목표신뢰도 면에서

$R_0 > R(x|0) > R_0^{\frac{1}{d}}$

인 경우이다. 이 범위의 조건일 때에 최적 발행시각을 결정

<표 2> 개발 소프트웨어의 발행시각 결정

신뢰도		비용		
		$\frac{c_3}{a\gamma(c_2 - c_1)} \geq 1$	$1 \geq \frac{c_3}{a\gamma(c_2 - c_1)} > d$	$e^{-a\gamma} < \frac{c_3}{a\gamma(c_2 - c_1)} \leq d$
$T_2$	$R(x 0) \geq R_0$	$T_1=0, T_2=0, T^*=0$	$0 < T_1 < T_{Lc}, T_2=0, T^*=T_2$	$T_1=T_{Lc}, T_2=0, T^*=T_{Lc}$
	$R_0^{\frac{1}{d}} < R(x 0) < R_0$	$T_1=0, 0 < T_2 < T_{Lc}, T^*=T_2$	$0 < T_1, T_2 < T_{Lc}, T^* = \max\{T_1, T_2\}$	$T_1=T_{Lc}, 0 < T_2 < T_{Lc}, T^*=T_{Lc}$
	$R(x 0) > R_0^{\frac{1}{d}}$	$T_1=0, T_2=T_{Lc}, T^*=T_{Lc}$	$0 < T_1 < T_{Lc}, T_2=T_{Lc}, T^*=T_{Lc}$	$T_1=T_{Lc}, T_2=T_{Lc}, T^*=T_{Lc}$

하고자 하는 본 연구의 목적에 부합된다. 이 경우에 대해서 최적발행시각이 비용 면에서, 그리고 목표신뢰도 면에서 어떠한 경향을 보이는지를 (그림 4), (그림 5)에 표시하였다. 이 두 그림에서 각각을 만족시키는 발행시각이  $T_1 > T_2$ 로 되어  $T^* = T_1$ 로 결정되는 경우가 가장 이상적이다. 그 외의 범위에서는 비용이나 목표신뢰도 어느 한쪽 또는 양쪽 모두가 적절한 해법이 없거나 제시하기 어려운 경우에 속하여 최적발행시각을 결정하기 어려우므로, 어느 한 쪽의 의도에 의해서 결정되어야만 한다.

4.4 적용 사례

Ohtera 외[8] 연구의  $\alpha = 2,253.2$ ,  $\beta = 4.5343 \times 10^{-4}$ ,  $\gamma = 1.5791 \times 10^{-3}$ ,  $m = 2.2580$ , Yamada 외[9] 연구의  $\alpha = 1,348$ ,  $b = 0.124$ ,  $x = 0.5$ ,  $c_1 = 1$ ,  $c_2 = 5$ ,  $c_3 = 4$ ,  $T_{LC} = 100$  경우에 대해서 본 논문의 발행시각 결정 방법을 적용한다.

4.4.1 비용

$d = 0.0285$ ,  $\frac{c_3}{a\gamma(c_2 - c_1)} = 0.470$ 이므로 식 (20)의 조건 1)에 해당된다.  $T_1 = 16.04$ 로서  $W(T_1) = 478.5$ ,  $W(x) = 0.2136$ ,  $W(T_{LC}) = 2,253$ ,  $C(T_1) = 5,603$ 이며, 이 때의 신뢰도는  $R(x | T_1) = 0.808$ 이므로 본 예제가 비록 비용을 최저로 하는 시점에 이르기는 하였지만 목표신뢰도에는 미치지 못한다. 즉, 비용을 최저로 하는 시각에서 신뢰도 요건은 만족되지 않는 경우이다.

4.4.2 신뢰도

$R(x | 0) = 0.635$ ,  $R_o^{\frac{1}{d}} = 0.0228$ 이므로 식 (22)의 조건 1)에 해당된다.  $T_2 = 22.8$ 이고, 이 때의 비용은  $W(T_2) = 924$ 이므로,  $C(T_2) = 6,105$ 로서 최저비용 5,603보다 증가되는 현상을 보이고 있다. 따라서, 본 예제에서는 최저비용시점에 의해서가 아니라 목표신뢰도 도달 시점에서 개발 소프트웨어가 발행되어야 할 것이다.

5. 결론

소프트웨어의 신뢰도에 대한 정의를 하고, 시험 시간의 경과와 신뢰도와의 관계, 결함 수정 후 경과되는 시간과 신뢰도와의 관계를 연구하였다. 소프트웨어의 수정비용을 고찰함에 있어서 시험 기간중의 결함 수정 비용, 운영 기간중의 결함 수정 비용, 시험 기간중의 시험 비용으로 세분하여 검토한 후 이들을 결합하여 비용이 최소로 되는 시점을 발행시각으로 검토하였다. 목표신뢰도 입장에서 발행시각을 결정함에 있어서 개발후 시험을 시작하기 전의 신뢰도  $R(x | 0)$ 가 어떠한 조건에 있는가를 검토하여 각 조건에 따른 최적 발행시각을 결정하였다. 비용의 입장에서 발행시각을 결정하는 문제와 신뢰도의 입장에서 발행시각을 결정하는 문제를 동시

에 고려하여 최적 발행시각을 결정하도록 하였으며, 각각의 조건 및 한계를 연구하였다. 이 결과를 <표 2>에 나타내었다.

비용면에서 볼 때 그 조건은  $\frac{c_3}{a\gamma(c_2 - c_1)} > 1$ ,  $1 \geq \frac{c_3}{a\gamma(c_2 - c_1)} > d$ ,  $\frac{c_3}{a\gamma(c_2 - c_1)} \leq d$  이고, 목표신뢰도면에서  $R(x | 0) \geq R_o$ ,  $R_o > R(x | 0) > R_o^{\frac{1}{d}}$ ,  $R(x | 0) < R_o^{\frac{1}{d}}$  이다. 이 중에서 가장 이상적인 경우는 비용면에서  $1 \geq \frac{c_3}{a\gamma(c_2 - c_1)} > d$  이고 목표신뢰도면에서  $R_o > R(x | 0) > R_o^{\frac{1}{d}}$  인 경우이다. 이 범위의 조건일 때에 최적 발행시각을 결정하고자 하는 본 연구의 목적에 부합된다. 이 경우에 대해서 최적발행시각이 비용 면에서, 그리고 목표신뢰도 면에서 어떠한 경향을 보이는지 그림으로 표시하였다. 이 두 그림에서 각각을 만족시키는 발행시각이  $T_1 > T_2$ 로 되어  $T^* = T_1$ 로 결정되는 경우가 가장 이상적이다. 그 외의 범위에서는 비용이나 목표신뢰도 어느 한쪽 또는 양쪽 모두가 적절한 해법이 없거나 제시하기 어려운 경우에 속하여 최적 발행시각을 결정하기 어려우므로, 어느 한 쪽의 의도에 의해서 결정되어야만 한다.

감사의 글

본 연구는 한국과학재단 목적기초연구(R01-2000-00273) 지원으로 수행되었음.

참고 문헌

- [1] C. V. Ramamoorthy, F. B. Bastani, "Software reliability - Status and perspectives," IEEE Trans. on Software Eng., Vol. SE-8, pp.354-371, Aug. 1982.
- [2] J. D. Musa, A. Iannino, K. Okumoto, "Software Reliability : Measurement, Prediction, Application," pp.230-238, Mar. 1987.
- [3] S. Yamada, H. Ohtera, H. Narihisa, "Software reliability growth models with testing-efforts," IEEE Trans. Reliability, Vol. R-35, pp.19-23, Apr. 1986.
- [4] H. Ascher, H. Feigold, "Repairable Systems Reliability : Modeling, Inference, Misconceptions, and Their Causes," Marcel Dekker, 1984.
- [5] K. Okumoto, A. L. Goel, "Optimum release time for software systems based on reliability and cost criteria," J. System software, Vol.1, pp.315-318, 1980.
- [6] S. Yamada, S. Osaki, "Cost-reliability optimal release policies for software systems," IEEE Trans. on Reliability, Vol. R-34, pp.422-424, Dec. 1985.
- [7] Rong Huei Hou, Sy-Yen Kuo, Yi-Ping Chang, "Optimal release policy for hyper-geometric distribution software-reliability growth model," IEEE Trans. on Reliability, Vol. 45, pp.646-651, Dec. 1996.
- [8] Hiroshi Ohtera, Shigeru Yamada, "Optimum Software-Release Time Considering an Error-Detection Phenomenon during Operation," IEEE Trans. on Reliability, Vol.39, No.5, pp.596-599, 1990 Dec.
- [9] Shigeru Yamada, Shunji Osaki, "Cost-Reliability Optimal Release Policies for Software Systems," IEEE Trans. on Reliability, Vol. R-34, No.5, pp.422-424, Dec. 1985.



### 최 규 식

e-mail : che@konyang.ac.kr

1976년 서울대학교 공과대학 전기과 졸업  
(학사)

1981년 뉴욕공과대학 전기과 졸업(제어공학  
석사)

1993년 명지대학교 전기과 졸업(제어공학  
박사)

1975년~1993년 OPC 중앙연구소, KOPEC 연구소 근무

1993~현재 건양대학교 정보전자통신공학부 교수

관심분야 : PLC, 소프트웨어 신뢰도



### 장 윤 승

e-mail : yschang@konyang.ac.kr

1990년 일본기후대학 전자공학과 졸업  
(공학사)

1992년 일본기후대학 전자정보시스템공학  
전공(공학석사)

1995년 일본기후대학 전자정보시스템공학  
전공(공학박사)

1995년~현재 건양대학교 정보전자통신공학부 조교수

1998년~1999년 ARDF 세계대회 조직위원회 조직위원

관심분야 : 정보 시스템, RF통신, 위성통신 등