정보기술응용연구
제 3 권 제 3 호
2 0 0 1 년 9 월

소프트웨어 개발시 일정테스트노력과 웨이불 테스트 노력의 비교 연구

최규식* , 김종기** , 장원석*

요 약

본 논문에서는 소프트웨어 테스트 단계중에 발생되는 테스트노력 소요량을 고려한 소프트웨어 신뢰도 성장 모델을 제시하여 시간종속적인 테스트 노력소요량 동태를 일정 테스트 노력일 때와 웨이불 테스트 노력일 때를 비교하여 연구한다. 테스트 단계중에 소요되는 테스트 노력의 양에 대한 결함 검출비를 현재의 결함 내용에 비례하는 것으로 가정하여 모델을 비동차 포아송 과정으로 공식화하며, 이 모델을 이용하여 소프트웨어 신뢰도 척도에 대한 데이터 분석기법을 개발하도록 한다. 테스트 시간의 경과와 신뢰도와의 관계도 심도 있게 연구한다. 목표신뢰도를 만족시키는 최적발행시각을 정한다.

개발 후 테스트를 시작하기 전의 신뢰도가 어떠한 조건에 있는가를 검토하여 각 조건에 따른 최적 발행시각을 결정한다. 일정 테스트 노력 곡선과 웨이불 테스트 노력 곡선 모두에 대해서 그 조건은 목표신뢰도를 초과하는 경우, 목표신뢰도를 초과하지는 못하지만 어느 조건 이상인 경우, 어느 조건 이하인 경우로 대별되며, 이 중에서 이상적인 경우는 두 번째 조건인 경우이다.

key words : SRGM, NHPP, 평균치 함수, 목표 신뢰도, 결함 검출, 일정테스트노력 함수, 웨이불 테스트노력 함수, 목표 신뢰도, 최적 발행시각

본 연구는 한국과학재단 목적기초연구(R01-2000-00273) 지원으로 수행되었음
*) 건양대학교 IT 학부
**) 건양대학교 첨단과학부

1. 서론

최근 운영시스템, 제어프로그램, 적용프로그램과 같은 여러 가지 소프트웨어 시스템이 더욱더 복잡화 및 대형화되고 있기 때문에 신뢰성이 높은 소프트웨어 시스템을 개발하는 일이 매우 중요하며, 따라서 소프트웨어 제품 개발에 있어서 소프트웨어의 신뢰도가 핵심사항이라고 할 수 있다. 소프트웨어 개발에서 중요한 문제는 개발된 소프트웨어의 테스트를 언제 중단하여 발행할 것이냐를 결정하는 것이다. 이 발행시기를 결정하는 적절한 방법중의 하나는 테스트중인 소프트웨어의 결함수를 예측할 수 있는 방법론을 확립하여, 이 방법에 의해 테스트 단계에서 소프트웨어 내의 결함을 검출하여 줄임으로써 목표신뢰도를 만족하도록 하는 방법이다.

제품 신뢰도를 특성화하기 위해서는 정량적인 측정과 관리가 필수적이며, 특히, 소프트웨어 결함데이타 분석에 근거하여 소프트웨어 테스트 단계에서 소프트웨어의 신뢰도를 추정하는 것이 매우 중요하다. 그동안 테스트 기간중의 소프트웨어 결함 검출현상을 설명하기 위한 여러 가지 소프트웨어 신뢰도 모델이 개발되었다. 소프트웨어 테스트에 의해서 발견되는 누적결함(또는 소프트웨어 고장간의 시간간격)과 소프트웨어 테스트 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 테스트 시간에 소프트웨어 내의 평균 잔여결함수, 소프트웨어 신뢰도 함수와 같은 소프트웨어 신뢰도 척도를 추정할 수 있다.

소프트웨어 개발에는 많은 개발자원들이 소요된다. 소프트웨어 테스트 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 테스트 시간, 테스트 노력량, 테스트 노력에 의해서 검출되는 소프트웨어 결함의 수 사이의 관계를 명시적으로 설명할 수 있는 간단하고도 새로운 모델을 제시하였다. 테스트 노력은 테스트 단계에서 소요되는 인력, CPU시간, 실행테스트 케이스 등등에 의해서 측정된다.

그동안 많은 연구가와 참여자들에 의해서 보편적으로 널리 연구되고 사용되는 SRGM의 부류는 NHPP(nonhomogeneous Poisson process) 모델이며, 이러한 부류의 모델은 실제로 많은 장점을 가지고 있어서 그간 많은 관심을 끌어왔다.

본 논문에서는 일정한 값으로서의 테스트노력과 웨이불곡선으로서 테스트노력을 시간종속 거동적으로 설명한다. 소프트웨어 테스트에서 결함 검출비가 현재의

내재 결함수에 비례하고, 임의의 테스트 시간에서 그 비례가 현재의 테스트 노력 여하에 달려 있다는 것을 가정하여 비동차 포아송 과정(NHPP)[4]에 바탕을 둔 신뢰도성장모델을 개발한다.

2장에서는 NHPP를 기준으로 한 SRGM에 대해서 고장확률 및 신뢰도를 연구하고, 3장에서는 테스트 시간중 두 가지 안에 대해서 목표신뢰도를 맞추어서 발행하는 발행 시각 결정 방법에 대해서 연구한다. 4장에서는 여러 조건을 고려하여 각 조건에 맞는 최적발행정책을 연구한다.

[표-1] 사용되는 기호 및 의미

$N(t)$: 시각 t까지 검출되는 소프트웨어의 누적결함 갯수
a : 초기부터 소프트웨어내에 존재하고 있는 결함의 갯수
b, b_1, b_2 : 일반적인 경우, 테스트시간중, 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_o : 목표신뢰도, $0<R_o<1$
T_{LC} : 소프트웨어의 수명주기
T : 전체 테스트 시간
T^* : 최적 소프트웨어 테스트 시간
T_1 : 일정테스트노력에서 목표신뢰도를 만족시키는 유일 해 T
T_2 : 웨이불테스트노력에서 목표신뢰도를 만족시키는 유일 해 T
d : 상수, $e^{bT_{LC}}$, $d>1$
g : 상수, $\exp[-\alpha\gamma(1-e^{-\beta T_{LC}^m})]$, $0<d<1$
α : 소프트웨어 테스트에서 필요로 하는 테스트노력 소요 총량
β, m : 척도모수, 형상모수
s : 결함이 발견되는 시각
$w(t)$: 웨이불 곡선

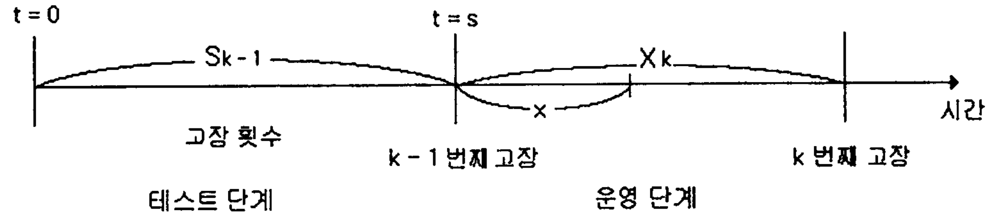
2. 소프트웨어의 신뢰도

소프트웨어 신뢰도는 규정된 환경 하에서 주어진 시간에 소프트웨어를 결함없이 운영할 수 있는 확률인 것으로 정의하며, 다음과 같이 조건확률로 표현할 수 있다.[2]

$$R(x|s) = \Pr\{X_k > x | S_{k-1} = s\} \qquad (1)$$

이는 소프트웨어를 개발하여 결함검출 테스트를 시작하여 계속 s 유니트 시간

동안 (k-1)번 째 결함을 발견하여 수정한 후, k번 째 결함이 발견되기 전까지 운영될 x 시간동안에 고장 없이 소프트웨어가 동작할 확률인 것이다. [그림-1]에 이 관계를 표시하였다.



[그림-1] 고장발생 표현

2.1 일정테스트 노력곡선

NHPP의 표준 이론으로부터 평균치 함수를

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

로 정의할 때 임의의 $t \geq 0$과 $x > 0$에서

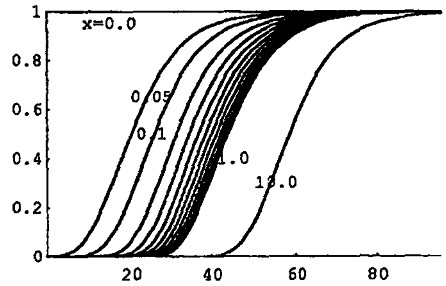
$$\Pr\{N(t+x)-N(t)=k\} = \frac{[m(t+x)-m(t)]^k}{k!}\exp\{-[m(t+x)-m(t)]\} \tag{3}$$

이므로, 식(1)의 신뢰도는 다음과 같이 표현할 수 있다.

$$R(x|t) \equiv \Pr\{N(t+x)-N(t)=0\} = \exp\{-[m(t+x)-m(t)]\}$$
$$= \exp[-a(1-e^{-bx})e^{-bt}] = \exp[-m(x)e^{-bt}] \tag{4}$$

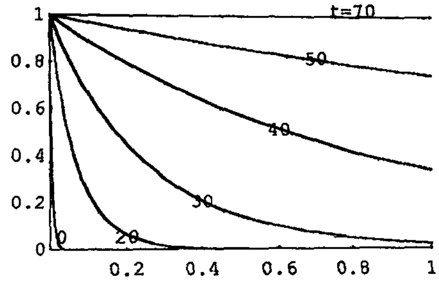
즉, 어느 시각 t부터 (t+x) 시각까지 새로운 결함이 발견되지 않을 확률을 신뢰도로 정의하며, 이는 평균치 함수의 차이를 지수함수의 지수로 취한 형태를 하고 있다.

식(4)로 표시된 신뢰도의 특성을 이해하기 위해 테스트시간과 신뢰도의 관계 및 최종 검출 결함 수정 후 경과시간과 신뢰도와 관계는 [그림-2], [그림-3]과 같다.



[그림-2] 발행시각과 신뢰도와의 관계

[그림-2]의 경우, 최종 결함 수정 후 경과되는 각각의 시간에 대해서 테스트 시간과 신뢰도 성장과의 관계를 보여주고 있다. 일정한 경과시간 x에 대해서 테스트시간 및 발행시기를 늦추면 늦출수록 신뢰도가 s 형 곡선으로 성장함을 알 수 있다. 그림에서 x = 0.0일 때는 테스트 시간에 관계없이 신뢰도가 1이나, x의 값이 커지면 커질수록 곡선이 횡축의 우측으로 이동하여 신뢰도가 저하됨을 알 수 있다. 결함 수정 후 경과시간을 어떤 범위로 하여 신뢰도 성장 기준을 잡는가도 중요한 문제이다.



[그림-3] 결함 수정후 경과시간과 신뢰도와의 관계

이와는 대조적으로 [그림-3]의 경우, 주어진 각각의 테스트 시간 t에 대해서 최종 결함 수정 후 경과시간 x와 신뢰도 성장과의 관계를 보여주고 있다. 일정한 테스트시간 t에 대해서 경과시간 x가 작으면 작을수록 신뢰도가 높으며, 경과시간 x가 증가함에 따라 신뢰도가 급격히 감소된다. 이 그림에서 보듯 테스트 시간이 길면 길수록 경과시간에 대한 신뢰도의 저하가 작아짐을 알 수 있다.

상기 식(4)에서 정의한 테스트 단계의 신뢰도 의미를 고찰해보기로 한다.

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

우리가 원하는 목표신뢰도를 R_o라 하면 식(4)로부터

$$\exp[-m(x)e^{-bt}] = R_o \quad (5)$$

가 된다.

주어진 기간에 맞추어 소프트웨어 시스템을 개발할 때 시간, 자금, 인력과 같은 자원들이 소요된다. 특히, 소프트웨어 테스트에까지 이르는 자원들이 소프트웨어 신뢰도에 상당한 영향을 미친다. 소프트웨어 개발 자원 전체의 약 40-50%가 테스트단계에서 소요된다.

2.2 웨이불 테스트노력 곡선

본 논문에서는 시각 t에서의 테스트노력 형상을 기술하기 위해 테스트 노력함수로서 웨이불곡선을 이용한다.

$$w(t) = \alpha \cdot \beta \cdot m \cdot t^{m-1} \cdot \exp[-\beta t^m] \quad (6)$$

(6)의 적분형태

$$W(t) = \alpha(1 - \exp[-\beta t^m]) \quad (7)$$

는 시각 (0,t]에서의 누적 테스트 노력량을 나타낸다.
웨이불 테스트 노력함수 (6)이나 (7)에서의 α, β, m은 최소자승법으로 구한다. 또, α, β, m은 (t_k, w_k)형태의 n개 관찰 데이터 쌍으로부터 결정한다.

소프트웨어 고장은 시스템 내에 잔존하고 있는 소프트웨어의 결함에 의해서 프로그램 동작이 제대로 되지 않는 것을 말한다. 평균치 함수를 다음과 같이 정의한다.

$$m(t) = a(1 - \exp[-rW(t)]) \quad (8)$$

소프트웨어 결함검출 현상을 통계적으로 모델링할 때에 식(8)의 $m(t)$에 의한 NHPP에 근거하여 $N(t)$의 평균치함수를 정의하면 웨이불 테스트 노력함수를 고려한 소프트웨어 신뢰도성장모델을 만들 수 있다.

$$\Pr\{N(t) = n\} = poim(n; m(t)) \quad (9)$$

NHPP 고장강도함수는 평균치 함수의 미분 형태이다.

$\lambda(t) = dm(t)/dt = a \cdot r \cdot w(t) \cdot \exp[-rW(t)]$ (10)

식(9)로부터 $N(t)$의 제한적인 분포가 평균치 함수

$m(\infty) = a(1-\exp[-r\alpha])$ (11)

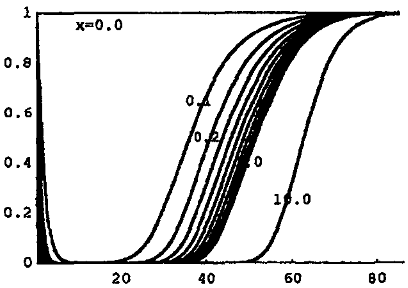
을 가진 포아송분포라는 것을 보여주고 있다. 즉, 어느 소프트웨어에 평균적으로 존재하는 결함의 수는 테스트를 무한정 계속할 때 발견되는 결함의 수와 같다는 것을 의미한다.

식(8)의 $m(t)$를 가진 NHPP 모델에 의해서 소프트웨어 신뢰도 평가에 대한 두 가지 정량적 평가 척도를 얻을 수 있다. 시각 t에서의 신뢰도는 다음과 같다.

$R(x \mid t) = \exp\{-a(\exp[-r \cdot W(t)]$

$-\exp[-r \cdot W(t+x)])\}$ (12)

식(12)로 표시된 신뢰도의 특성을 이해하기 위해 테스트시간과 신뢰도의 관계 및 최종 검출 결함 수정 후 경과시간과 신뢰도의 관계는 [그림-4]와 같다.



[그림-4] 발행시각과 신뢰도와의 관계

이 그림에서는 최종 결함 수정 후 경과되는 각각의 시간에 대해서 발행시각과 신뢰도 성장과의 관계를 보여주고 있다. 일정한 경과시간 x에 대해서 테스트시간 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. 이것은 일정 테스트 노력일 경우와 유사하다.

또, 주어진 각각의 테스트 시간 t에 대해서 최종 결함 수정 후 경과시간 x와 신뢰도 성장과의 관계는 [그림-3]에서 보는 바와 유사하다.

상기 식(12)에서 정의한 테스트 단계의 신뢰도 의미를 고찰해보기로 한다.

$R(x|t)$는 시각 t에서 최종적으로 결함을 발견하여 수정한 후, x 유니트의 경과시간동안 새로운 결함이 발견되지 않을 확률이다. 소프트웨어를 개발하여 결함테스트를 하면 할수록 결함을 발견하여 수정하는 빈도가 작아지므로 신뢰도가 성

장되며, 결함 수정 후 경과시간이 길어지면 질수록 결함 발견 확률이 높아지기 때문에 소프트웨어의 신뢰도는 낮아진다. 한편, 테스트 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 분포는

$\overline{a}=a-a(1-e^{-\gamma W(T)})=a\cdot e^{-\gamma W(T)}$이다. 이것이 소프트웨어를 발행해서 운전하는 초기 결함수이므로

$$m(T+x)=a\cdot e^{-\gamma W(T)}\cdot(1-e^{-\gamma W(x)})+m(T)=a(1-e^{-\gamma W(T)-\gamma W(x)}) \quad (13)$$

이다. 그러므로,

$$\begin{aligned} R(x|T) &= \exp[-m(T+x)+m(T)] \\ &= \exp[-ae^{-\gamma W(T)}(1-e^{-\gamma W(x)})] \\ &= \exp\{-ae^{-\alpha\gamma(1-e^{-\beta T^{m}})}[1-e^{-\alpha\gamma(1-e^{-\beta x^{m}})}]\} \end{aligned} \quad (14)$$

이고, 여기서

$$R(x|0)=\exp\{-a[1-e^{-\alpha\gamma(1-e^{-\beta x^{m}})}]\} \quad (15)$$

이다.

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

3. 발행시각 결정

3.1 일정테스트 노력

식(14)로부터

$$T_1=\frac{1}{b}\ln\frac{m(x)}{\ln\frac{1}{R_o}}$$

이므로

$$0<\frac{1}{b}\ln\frac{m(x)}{\ln\frac{1}{R_o}}<T_{LC}$$

인 범위 즉,

$$1<\frac{m(x)}{\ln\frac{1}{R_o}}<d$$

$\ln \frac{1}{R_o} < m(x) < d \cdot \ln \frac{1}{R_o}$

$\ln R_o > -m(x) > d \cdot \ln R_o$

$R_o > e^{-m(x)} > R_o^d$

이므로,

1) $R_o > R(x|0) > R_o^d$에서 양의 유일 해 $T^* = T_1$가 존재한다.

2) $R(x|0) > R_o$이면 $T_1 < 0$이므로 $T^* = T_1 = 0$이다.

3) $R(x|0) < R_o^d$이면 $T_1 > T_{LC}$이므로 $T^* = T_1 = T_{LC}$이다

1)의 경우에 대해서 좀더 검토해보기로 한다.

$$\exp(-m(x)) = R(x|0) \tag{16}$$

이므로,

$$m(x) = -\ln R(x|0) = \ln \frac{1}{R(x|0)} \tag{17}$$

으로 되어 식(4)로부터 T_1을 다음과 같이 구할 수 있다.

$$T^* = T_1 = \frac{1}{b} \ln \left\{ \frac{\ln \frac{1}{R(x|0)}}{\ln \frac{1}{R_o}} \right\} \tag{18}$$

여기서, 편의상 $R(x|0) = t$ 라고 가정하고 식(18)을 t에 관하여 두 번 미분하면

$$\frac{dT_1^2}{dt^2} = \frac{1}{b} \left\{ \frac{\ln \frac{1}{t} - 1}{(t \cdot \ln \frac{1}{t})^2} \right\} \tag{19}$$

이 되어 함수의 변곡점은 목표신뢰도 R_o의 값에 관계 없이 항상

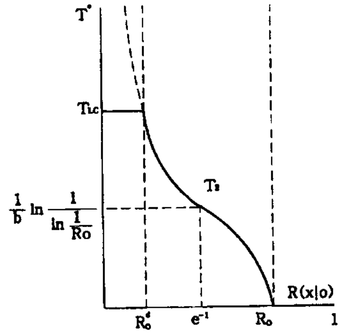
$$t = R(x|0) = \frac{1}{e} \tag{20}$$

인 값으로 결정된다.

이 때

$$T_1 = \frac{1}{b} \left\{ \ln \frac{1}{\ln \frac{1}{R_o}} \right\} \tag{21}$$

이다. 이러한 관계를 [그림-5]에 표시하였다.



[그림-5] 목표신뢰도와 발행시각과의 관계

3.2 웨이불테스트

마찬가지로, 신뢰도를 요건에 최근접시키는 유일한 시각이 존재한다. 최적 소프트웨어 발행시각은 미리 규정된 소프트웨어 목표신뢰도를 만족시키는 최근접이 되는 시각이다.

발행시각 T에서의 신뢰도는 식(14)와 같으므로 여기서 목표신뢰도를 만족시키는 발행시각을 구하면 다음과 같다.

목표신뢰도를 R_o라 하면 식(14)에서

$$\exp\{-ae^{-\alpha\gamma(1-e^{-\beta T^m})}[1-e^{-\alpha\gamma(1-e^{-\beta x^m})}]\}=R_o \tag{22}$$

이다. 식(22)를 만족하는 발행시각을 T_2라 하면

$$T_2=\left\{-\frac{1}{\beta}\ln[1+\frac{1}{\alpha\gamma}\ln\frac{\ln R_o}{\ln R(x|0)}]\right\}^{\frac{1}{m}} \tag{23}$$

이므로

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

인 범위 즉,

1) $R_o>R(x|0)>R_o^{\frac{1}{g}}$

에서 양의 유일 해 $T^*=T_2$이 존재한다.

이 값은 식(23)으로 표현된다.

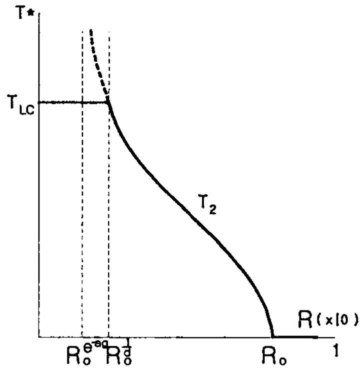
2) $R(x|0)\geq R_o$

이면 $T_1\leq 0$이므로 $T^*=T_2=0$이다.

3) $R(x|0) < R_o^{1/g}$

이면 $T_1 \geq T_{LC}$이므로 $T^* = T_2 = T_{LC}$이다.

이와 같은 내용을 [그림-6]에 표시하였다.



[그림-6] 목표신뢰도와 발행시각 곡선

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

$R_o > R(x|0) > R_o^{1/g}$는 소프트웨어를 개발한 후 테스트 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.

$R(x|0) < R_o^{1/g}$는 소프트웨어의 전 수명기간에 걸쳐서 테스트를 해도 목표신뢰도를

만족시키지 못하는 경우이다.

4. 최적 발행 시각

4.1일정 테스트 노력

$R(x|0)$은 소프트웨어를 개발하여 테스트를 거치지 않은 상태에서 x 시간까지 시간이 경과할 때의 소프트웨어 신뢰도를 나타내는 것이다.

1) $R(x|0) > R_o$
는 소프트웨어 개발 즉시 신뢰도가 목표신뢰도를 만족하기 때문에 더 이상 결함 발견을 위한 테스트를 할 필요가 없는 경우로서 비용을 최저로 하는 시기에 맞추어 발행시기를 결정해야 하는 경우이다. 그러나, 이러한 경우는 현실적으로 고려하기 어려운 경우이다.

2) $R_o > R(x|0) > R_o^d$
는 소프트웨어를 개발한 후 테스트 및 결함 수정을 통하여 목표신뢰도를 만족시키는 경우이다. 이러한 경우는 목표신뢰도를 만족시키는 발행시기와 총 비용을 최저로 하는 발행시기 중 큰 값을 취하는 것이 이상적이다.

3) $R(x|0) < R_o^d$
는 소프트웨어의 전 수명기간에 걸쳐서 테스트를 해도 목표신뢰도를 만족시키지 못하는 경우로서 소프트웨어 개발에 실패한 경우로 볼 수 있다.

따라서, 이 중에서 가장 이상적인 범위는
$R_o > R(x|0) > R_o^d$
인 경우이다.

4.2 웨이블 테스트 노력

$R(x|0)$은 소프트웨어를 개발하여 테스트를 거치지 않은 상태에서 x 시간까지 시간이 경과할 때의 소프트웨어 신뢰도를 나타내는 것이다.

1) $R(x|0) \geq R_o$
는 소프트웨어 개발 즉시 신뢰도가 목표신뢰도를 만족하기 때문에 더 이상 결함 발견을 위한 테스트를 할 필요가 없는 경우로서 비용을 최저로 하는 시기에 맞추어 발행시기를 결정해야 하는 경우이다. 그러나, 이러한 경우는 현실적으로 고려하기 어려운 경우이다.

2) $R_o > R(x|0) > R_o^{\frac{1}{g}}$
는 소프트웨어를 개발한 후 테스트 및 결함 수정을 통하여 목표신뢰도를 만족시키는 경우이다. 이러한 경우는 목표신뢰도를 만족시키는 발행시기와 총 비용을 최저로 하는 발행시기 중 큰 값을 취하는 것이 이상적이다.

3) $R(x|0) < R_o^{\frac{1}{g}}$

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

따라서, 이 중에서 가장 이상적인 경우는

$R_o > R(x|0) > R_o^{\frac{1}{g}}$

인 경우이다. 그 외의 범위에서는 비용이나 목표신뢰도 어느 한 쪽 또는 양 쪽 모두가 적절한 해법이 없거나 제시하기 어려운 경우에 속하여 최적발행시각을 결정하기 어려우므로, 어느 한 쪽의 의도에 의해서 결정되어야만 한다.

4.3 실제 예

α=6759.6, β=4.5343×10 ^{3}, γ=1.5791×10^{-3}, m=0.9032, R_o=0.9, a=200, b=0.0248, x=0.1, T_{LC}=100이라 하자.

1) 일정 테스트 노력

$d = e^{bT_{LC}}$=11.9417, $R(x|0) = e^{-m(x)} = e^{-a(1-e^{-bx})}$==0.3280, $R_o^d = 0.9^{11.9417}$=0.2842

이므로 식(16)의 조건 1)에 해당된다. 이 때 T_1=95.12가 된다.

2) 웨이불 테스트 노력

$g = e^{-\alpha\gamma(1-e^{-\beta T_{LC}^{m}})}$=0.0679, $R_o^{\frac{1}{g}}$ =0.2119, $R(x|0) = e^{-a(1-e^{-\alpha\gamma(1-e^{-\beta x^{m}})})}$=0.2995이므로, 식(24)의 조건 1)에 해당된다. 이 때 T_2=49.55이다.

5. 결론

소프트웨어의 신뢰도에 대한 정의를 하고, 테스트 시간의 경과와 신뢰도와의 관계, 결함 수정 후 경과되는 시간과 신뢰도와의 관계를 연구하였다. 발행시각을 결정함에 있어서 개발 후 테스트를 시작하기 전의 신뢰도 $R(x|0)$가 어떠한 조건에 있는가를 검토하여 각 조건에 따른 최적 발행시각을 결정하였다. 일정 테스트

노력 곡선인 경우, 그 조건은 $R(x|0) > R_o$, $R_o > R(x|0) > R_o^d$, $R(x|0) < R_o^d$이다. 이 중에서 이상적인 경우는 $R_o > R(x|0) > R_o^d$인 경우이다. 이 범위의 조건일 때에 최적 발행시각을 결정하고자 하는 본 연구의 목적에 부합된다. 이 경우에 대해서 최적발행시각이 목표신뢰도 면에서 어떠한 경향을 보이는지 그림으로 표시하였다. 그 외의 범위에서는 비용이나 목표신뢰도 어느 한 쪽 또는 양 쪽 모두가 적절한 해법이 없거나 제시하기 어려운 경우에 속하여 최적 발행시각을 결정하기 어려우므로, 어느 한 쪽의 의도에 의해서 결정되어야만 한다.

웨이불곡선인 경우를 고찰해보면 $R(x|0) \geq R_o$, $R_o > R(x|0) > R_o^{\frac{1}{g}}$, $R(x|0) < R_o^{\frac{1}{g}}$이다. 이 중에서 이상적인 경우는 $R_o > R(x|0) > R_o^{\frac{1}{g}}$인 경우이다. 이 범위의 조건일 때에 최적 발행시각을 결정하고자 하는 본 연구의 목적에 부합된다. 그 외의 범위에서는 비용이나 목표신뢰도 어느 한 쪽 또는 양 쪽 모두가 적절한 해법이 없거나 제시하기 어려운 경우에 속하여 최적 발행시각을 결정하기 어려우므로, 어느 한 쪽의 의도에 의해서 결정되어야만 한다.

참고문헌

[1] C. V. Ramamoorthy, F. B. Bastani, "Software reliability - Status and perspectives", IEEE Trans. on Software Eng., vol. SE-8, pp354-371, 1982 August

[2] J. D. Musa, A. Iannino, K. Okumoto, "Software Reliability : Measurement, Prediction, Application", pp230-238, 1987 March

[3] S. Yamada, H. Ohtera, H. Narihisa, "Software reliability growth models with testing- efforts", IEEE Trans. on Reliability, vol. R-35, pp19-23, 1986 April

[4] H. Ascher, H. Feigold, "Repairable Systems Reliability : Modeling, Inference, Misconceptions, and Their Causes", 1984, March

[5] Martin Trachtenberg, "The Linear Software Reliability Model and Uniform Testing", IEEE Trans. on Reliability, vol. R-34, pp8-16, 1985 April

A Comparison Study between Uniform Testing Effort and Weibull Testing Effort during Software Development

Geu-Sik Che, Jong-Ki Kim, Won-Seog Chang

Abstract

We propose a software-reliability growth model incoporating the amount of uniform and Weibull testing efforts during the software testing phase in this paper. The time-dependent behavior of testing effort is described by uniform and Weibull curves. Assuming that the error detection rate to the amount of testing effort spent during the testing phase is proportional to the current error content, the model is formulated by a nonhomogeneous Poisson process. Using this model the method of data analysis for software reliability measurement is developed.

The optimum release time is determined by considering how the initial reliability $R(x|0)$ would be. The conditions are $R(x|0) > R_o$, $R_o > R(x|0) > R_o^d$ and $R(x|0) < R_o^d$ for uniform testing efforts. Ideal case is $R_o > R(x|0) > R_o^d$.

Likewise, it is $R(x|0) \geq R_o$, $R_o > R(x|0) > R_o^{\frac{1}{g}}$ and $R(x|0) < R_o^{\frac{1}{g}}$ for Weibull testing efforts. Ideal case is $R_o > R(x|0) > R_o^{\frac{1}{g}}$.

◆저자소개◆

최규식(崔圭植)

1948년 12월 29일생. 1976년 서울대 공대 전기공학과 졸업(학사). 1983년 뉴욕공과대학 졸업(석사). 1993년 명지대학교 졸업(박사). 현재 건양대학교 공대 IT 학부 부교수

E-mail : che@konyang.ac.kr
Tel : 02-441-4732

김종기(金鐘起)

1955년 2월 10일생. 1977년 인하대 수학과 졸업(학사). 1982년 인하대 수학과 졸업(석사). 1992년 인하대 수학과 졸업(박사). 현재 건양대학교 첨단과학부 교수

E-mail : jkkim@konyang.ac.kr
Tel : 041-735-5688

장원석(張元碩)

1952년 3월 25일생. 1978년 인하대 전자공학과 졸업(학사). 1980년 인하대 전자공학과 졸업(석사). 1988년 인하대 전자공학과 졸업(박사). 현재 건양대학교 IT 학부 교수

E-mail : wschang@konyang.ac.kr
Tel : 042-485-9481