

테스트노력이 균일한 소프트웨어의 최적발행

최규식* · 김종기**

Optimum Release of Uniform Testing Software

Che Gyu Shik* · Kim Jong Ki**

요 약

소프트웨어를 개발 및 발행함에 있어서 테스트와 디버깅 작업을 얼마나 오랫동안 수행하여 발행하느냐에 따라 운영중에 발생하는 결함 및 수정비용이 달라지므로 이 시기를 결정하는 것이 매우 중요하다. 테스트노력이 균일할 수도 비균일할 수도 있으나, 일반적으로 테스트노력은 일정한 것으로 가정하므로 본 논문에서도 일정테스트인 경우에 대해서 고찰한다.

1. 서론

소프트웨어의 발행시기를 결정하는 적절한 방법 중의 하나는 개발 후 테스트중인 소프트웨어의 결함수를 예측할 수 있는 방법론을 확립하여, 이 방법에 의해 테스트 단계에서 소프트웨어 내의 결함을 검출하여 줄임으로써 목표신뢰도를 만족하도록 하고, 개발비 및 발행 전후에 발생하는 테스트비용을 합친 총 비용을 가능한 한 최소화시킬 수 있는 방안이 무엇인가를 연구하여 종합적으로 개발 소프트웨어의 발행시기를 결정하는 것이다.

Okumoto와 Goel은 전체평균 소프트웨어 비용을 최소화시키는 비용-최적 SRP(software reliability process)를 발표하였다. Yamada와 Osaki는 전체 평균 비용을 최소화시키고 소프트웨어 신뢰도를 만족시키는 전체평균비용-신뢰도-최적 SRP를 도입하였다. 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 지금까지 연구된 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.

2. 소프트웨어의 신뢰도

소프트웨어 신뢰도는 규정된 환경 하에서 주어진 시간에 소프트웨어를 결함 없이 운영할 수 있는

확률인 것으로 정의하며, 다음과 같이 조건확률로 표현할 수 있다.

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

식(1)의 시간간격 X_k 가 소프트웨어의 테스트로서 테스트중이고 테스트공정이 NHPP를 따른다면 NHPP의 표준 이론으로부터 평균치 함수를

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

로 정의할 때 임의의 $t \geq 0$ 과 $x > 0$ 에서

$$\Pr\{N(t+x) - N(t) = k\} =$$

$$\frac{[m(t+x) - m(t)]^k}{k!} \exp\{-[m(t+x) - m(t)]\}$$

이므로, 식(1)의 신뢰도는 다음과 같이 표현할 수 있다.

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

식(4)로 표시된 신뢰도의 특성을 이해하기 위해 테스트시간과 신뢰도의 관계 및 최종 검출 결함수 정 후 경과시간과 신뢰도의 관계를 그림으로 나타내면 그림 1, 그림 2와 같다.

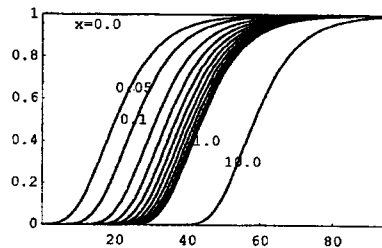


그림 1 발행시각과 신뢰도와의 관계

그림 1의 경우, 최종 결함 수정 후 경과되는 각각의 시간에 대해서 테스트 시간과 신뢰도 성장과의 관계를 보여주고 있다. 일정한 경과시간 x 에 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 결함이 발견될 확률이 높아 신뢰도가 저하된다는 것도 알 수 있다. 그림에서 $x = 0.0$ 일 때는 테스트 시간에 관계없이 신뢰도가 1이나, x 의 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.

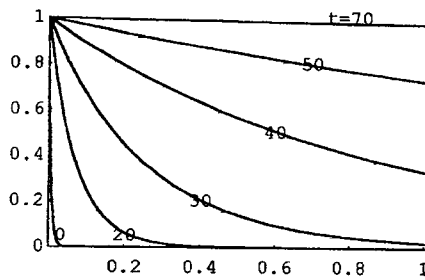


그림 2 결함 수정후 경과시간과 신뢰도와의 관계

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

원하는 목표신뢰도를 R_0 라 하면 식(4)로부터

$$\exp[-m(x)e^{-bx}] = R_0 \quad (5)$$

가 되어야 한다.

3. 소프트웨어의 수정 비용

테스트 기간중의 결함 수정 비용은 검출된 결함 하나하나를 수정하는데 비용이 발생되므로, 테스트 기간중 검출되는 총 결함의 수에 결함당 수정비용을 곱한 값이 된다.

$$c_1 m_1(T) = c_1 a(1 - e^{-b_1 T}) \quad (6)$$

운영중에 검출되는 결함의 수정비용은 발행 후 수명이 끝나는 시점까지 발생하는 결함에 대해서 수정하는데 드는 비용이므로

$$c_2 \{m_2(T_{LC}) - m_1(T)\} = c_2 a e^{-b_1 T} (1 - e^{-b_2(T_{LC}-T)}) \quad (7)$$

와 같이 표현된다.

테스트 기간 중에 발생하는 비용은 전 테스트기간에 단위시간당 테스트비용을 곱한 값이 된다.

$$c_3 T \quad (8)$$

상기와 같은 논리에 의해서 총 비용은 테스트기간중의 결함 수정 비용, 운영기간중의 결함 수정 비용, 테스트기간중의 테스트 비용을 합한 값이 된다.

$$C(T) = c_1 m_1(T) + c_2 \{m_2(T_{LC}) - m_1(T)\} + c_3 T \\ = c_1 a(1 - e^{-b_1 T}) + c_2 a e^{-b_1 T} (1 - e^{-b_2(T_{LC}-T)}) + c_3 T \quad (9a)$$

최적 소프트웨어 발행시각은 전체 평균 소프트웨어 비용을 최소로 하는 테스트시간이다.

$C(T)$ 의 최저값을 구하기 위해 식(9a)를 T 로 미분하여 정리한다.

$$(c_1 - c_2) a b_1 e^{-b_1 T} + \\ c_2 a (b_1 - b_2) e^{-b_2 T_{LC}} e^{-(b_1 - b_2) T} + c_3 = 0$$

이 식을 만족시키는 $T > 0$ 인 범위의 T 값을 구하면 $C(T)$ 에 대한 최소값이 된다. 그러나, 이 식에서 보통의 해석적인 방법으로는 T 값을 구하기가 쉽지 않다. 따라서, 수치해석적인 반복법에 의하여 최적발행시각 $T=T_1$ 을 구한다. 특별히 소프트웨어 발행 전후의 결함검출비율이 같을 경우에는 $b_1 = b_2 = b$ 가 되기 때문에 상기식이 단순화되어 이 때 식(9a)는

$$C(T) = c_1 m(T) + c_2 \{m(T_{LC}) - m(T)\} + c_3 T \quad (9b)$$

와 같이 된다.

$$T_1 = -\frac{1}{b} \ln \frac{c_3}{ab(c_2 - c_1)} \quad (10)$$

마찬가지로, 유사한 해법이 존재하여 신뢰도를 요건에 최근접시키는 유일한 시각이 존재한다. 최적 소프트웨어 발행시각은 미리 규정된 식(5)의 소프트웨어 목표신뢰도를 만족시키는 최근접이 되는 시각이다.

$$T_2 = \frac{1}{b_1} [\ln m(x) - \ln(\ln \frac{1}{R_0})] \quad (11a)$$

단, R_0 는 개발 소프트웨어가 추구하는 목표신뢰도이다. T_2 의 값이 커질수록 신뢰도가 성장된다. 특히, $b_1 = b_2 = b$ 인 경우는 단순화되어

$$T_2 = \frac{1}{b} \left\{ \ln \frac{m(x)}{\ln \frac{1}{R_0}} \right\} \quad (11b)$$

로 된다.

4. 비용-신뢰도 최적 소프트웨어 발행

소프트웨어 테스트로부터 구한 소프트웨어의 신뢰도를 어떤 규정치로 유지하는 제한 하에 전체 평균 소프트웨어의 비용을 최소로 하는 최적 소프트웨어 발행정책에 대해서 고려하기로 한다. 그림 3

에서는 테스트 기간을 횡축으로 하여 비용과 신뢰도의 관계를 동시에 나타낸 것이다. 최적 소프트웨어 발행문제는 아래와 같이 공식화할 수 있다.

$c_2 > 0, c_1 > c_3 > 0, x \geq 0, 0 < R_0 < 1$ 인 경우에 대해서 $R(x)T \geq R_0, T \geq 0$ 인 조건하에 $C(T)$ 를 최소화(12)

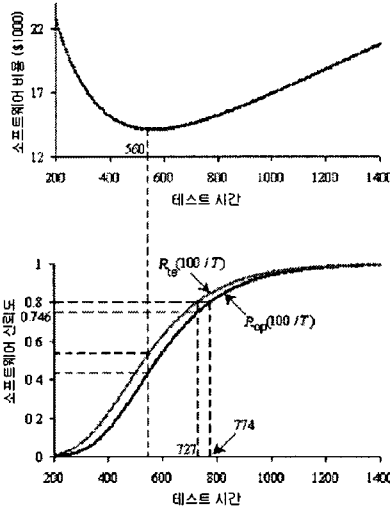


그림 3 비용-신뢰도 곡선

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

$$T^* = \max \{ T_1, T_2 \} \quad (13)$$

여기서, T_1 은 (10)에서, T_2 는 (11)에서 구한 값이다.

4.1 비용을 고려한 발행시각

비용면에서 식(10)으로부터

$$T_1 = -\frac{1}{b} \ln \frac{c_3}{ab(c_2 - c_1)}$$

이므로

$$0 < -\frac{1}{b} \ln \frac{c_3}{ab(c_2 - c_1)} < T_{LC}$$

인 범위 즉,

$$1) \ 1 > \frac{c_3}{ab(c_2 - c_1)} > \frac{1}{d}$$

에서 양의 유일 해 $T^* = T_1$ 이 존재한다.

$$2) \ \frac{c_3}{ab(c_2 - c_1)} > 1 \text{ 이면 } T_1 < 0 \text{ 이므로 } T^* = T_1 = 0 \text{ 이다.}$$

$$3) \ \frac{c_3}{ab(c_2 - c_1)} < \frac{1}{d} \text{ 이면 } T_1 > T_{LC} \text{ 이므로 } T^* = T_1 = T_{LC} \text{ 이다.}$$

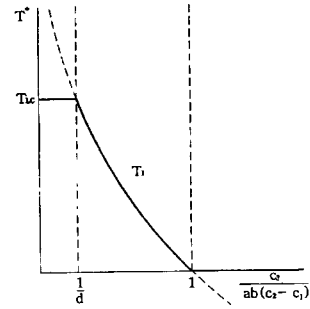


그림 4. 비용과 발행시각과의 관계

4.2 목표신뢰도를 고려한 발행시각

목표신뢰도 면에서 식(12)로부터

$$T_2 = \frac{1}{b} \ln \frac{m(x)}{\ln \frac{1}{R_0}}$$

이므로

$$0 < \frac{1}{b} \ln \frac{m(x)}{\ln \frac{1}{R_0}} < T_{LC}$$

인 범위 즉,

$$1 < \frac{m(x)}{\ln \frac{1}{R_0}} < d$$

$$\ln \frac{1}{R_0} < m(x) < d \cdot \ln \frac{1}{R_0}$$

$$\ln R_0 > -m(x) > d \cdot \ln R_0$$

$$R_0 > e^{-m(x)} > R_0^d$$

이므로,

1) $R_0 > R(x) > R_0^d$ 에서 양의 유일 해 $T^* = T_2$ 가 존재한다.

2) $R(x) > R_0$ 이면 $T_2 < 0$ 이므로 $T^* = T_2 = 0$ 이다.

3) $R(x) < R_0^d$ 이면 $T_2 > T_{LC}$ 이므로 $T^* = T_2 = T_{LC}$ 이다

1)의 경우에 대해서 좀더 검토해보기로 한다.

$$\exp(-m(x)) = R(x) \quad (14)$$

이므로,

$$m(x) = -\ln R(x) = \ln \frac{1}{R(x)} \quad (15)$$

으로 되어 식(12)는

$$T_2 = \frac{1}{b} \ln \left[\frac{\ln \frac{1}{R(x)}}{\ln \frac{1}{R_0}} \right] \quad (16)$$

로도 표현할 수 있다.

여기서, 편의상 $R(x|0) = t$ 라고 가정하고 식(16)을 t 에 관하여 두 번 미분하면

$$\frac{dT_2^2}{dt^2} = \frac{1}{b} \left\{ \frac{\ln \frac{1}{t} - 1}{(t \cdot \ln \frac{1}{t})^2} \right\} \quad (17)$$

이 되어 함수의 변곡점은 목표신뢰도 R_0 의 값에 관계 없이 항상

$$t = R(x|0) = \frac{1}{e} \quad (18)$$

인 값으로 결정된다.
이 때

$$T_2 = \frac{1}{b} \left\{ \ln \frac{1}{\ln \frac{1}{R_0}} \right\} \quad (19)$$

이다.

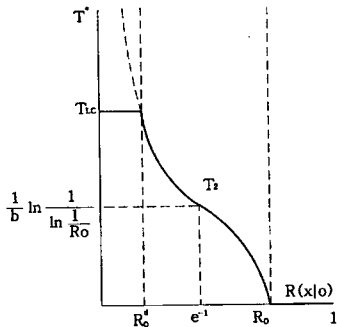


그림 5 목표신뢰도와 발행시각과의 관계

4.3 최적발행시각

이상의 내용을 요약하면 다음 표 1과 같다.

표 1 개발 소프트웨어의 발행시각 결정
Table 1 release time determination of developed software

신뢰도 \ 비용		T_1		
		$\frac{c_3}{ab(c_2 - c_1)}$	$1 > \frac{c_3}{ab(c_2 - c_1)} >$	$\frac{c_3}{ab(c_2 - c_1)} < \frac{1}{d}$
T_2	$R(x 0) > R_0$	$T_1=0, T_2=0, T^*=0$	$0 < T_1 < T_{LC}, T_2=0, T^*=T_2$	$T_1=T_{LC}, T_2=0, T^*=T_{LC}$
	$R_0 > R(x 0) > R_0^d$	$T_1=0, 0 < T_2 < T_{LC}, T^*=T_2$	$0 < T_1, T_2 < T_{LC}, T^*=\max\{T_1, T_2\}$	$T_1=T_{LC}, 0 < T_2 < T_{LC}, T^*=T_{LC}$
	$R(x 0) < R_0^d$	$T_1=0, T_2=T_{LC}, T^*=T_{LC}$	$0 < T_1 < T_{LC}, T_2=T_{LC}, T^*=T_{LC}$	$T_1=T_{LC}, T_2=T_{LC}, T^*=T_{LC}$

4.3.1 비용

이 표를 비용면에서 고찰해보기로 한다.

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

이때 비용이 단조증가하는 경우로서 결함테스트를 하면 할수록 비용이 증가되어 테스트 없이 발행하는 것이 최적인 것을 의미한다. 그러나, 이러한 경우는 현실적으로 고려하기 어려운 경우이다.

$$2) 1 > \frac{c_3}{ab(c_2 - c_1)} > \frac{1}{d}$$

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

$$3) \frac{c_3}{ab(c_2 - c_1)} < \frac{1}{d}$$

는 비용이 단조감소하는 경우이다. 이러한 경우는 결함테스트를 하면 할수록 총 비용이 감소되는 것으로서 현실적으로 있을 수 없는 경우에 속한다.

4.3.2 신뢰도

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

$$1) R(x|0) > R_0$$

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

$$2) R_0 > R(x|0) > R_0^d$$

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

$$3) R(x|0) < R_0^d$$

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

따라서, 이 중에서 가장 이상적인 범위는 비용면에서

$$1 > \frac{c_3}{ab(c_2 - c_1)} > \frac{1}{d}$$

이고 목표신뢰도 면에서

$$R_0 > R(x|0) > R_0^d$$

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 보이는지를 그림5, 6에 표시하였다. 이 두 그림에서 각각을 만족시키는 발행시각이 $T_1 > T_2$ 로 되어 $T^* = T_1$ 로 결정되는 경우가 가장 이상적이다. 그 외의 범위에서

서는 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.

비용면에서 볼 때

$$T_1 = \left\{ -\frac{1}{\beta} \ln \left[1 + \frac{1}{\alpha\gamma} \ln \frac{C_3}{\alpha\alpha(C_2 - C_1)} \right] \right\}^{\frac{1}{m}} \text{이므로}$$

$$0 < \left\{ -\frac{1}{\beta} \ln \left[1 + \frac{1}{\alpha\alpha} \ln \frac{C_3}{\alpha\gamma(C_2 - C_1)} \right] \right\}^{\frac{1}{m}} < T_{LC} \text{인 범위 즉,}$$

1) $\frac{C_3}{\alpha\alpha(C_2 - C_1)} > \exp[-\alpha\gamma(1 - e^{-\beta T_{LC}})]$ 에서 양의 유일 해 $T^* = T_1$ 이 존재한다.

$\frac{C_3}{\alpha\alpha(C_2 - C_1)} > 1$ 이면 $T_1 < 0$ 이므로 $T^* = T_1 = 0$ 이다.

$\frac{C_3}{\alpha\alpha(C_2 - C_1)} < \exp[-\alpha\gamma(1 - e^{-\beta T_{LC}})]$ 이면 $T_1 > T_{LC}$ 이므로 $T^* = T_1 = T_{LC}$ 이다.

한편, 목표신뢰도면에서 볼 때

$$T_2 = \frac{1}{b} \ln \frac{m(x)}{\ln \frac{1}{R_0}} \text{이므로}$$

$$0 < \frac{1}{b} \ln \frac{m(x)}{\ln \frac{1}{R_0}} < T_{LC} \text{인 범위 즉,}$$

$R_0 > R(x|0) > R_0 \exp(e^{\beta T_{LC}})$ 에서 양의 유일 해 $T^* = T_2$ 가 존재한다.

$R(x|0) > R_0$ 이면 $T_2 < 0$ 이므로 $T^* = T_2 = 0$ 이다.

$R(x|0) < R_0 \exp(e^{\beta T_{LC}})$ 이면 $T_2 > T_{LC}$ 이므로 $T^* = T_2 = T_{LC}$ 이다

이상의 내용을 요약하면 다음 표와 같다.

표 소프트웨어의 발행시기 결정조건

신뢰도	비용	T_1		
		①	②	③
T_2	④	$T_1=0, T_2=0, T^*=0$	$0 < T_1 < T_{LC}, T_2=0, T^*=T_2$	$T_1=T_{LC}, T_2=0, T^*=T_{LC}$
	⑤	$T_1=0, 0 < T_2 < T_{LC}, T^*=T_2$	$0 < T_1, T_2 < T_{LC}, T^*=\max(T_1, T_2)$	$T_1=T_{LC}, 0 < T_2 < T_{LC}, T^*=T_{LC}$
	⑥	$T_1=0, T_2=T_{LC}, T^*=T_{LC}$	$0 < T_1 < T_{LC}, T_2=T_{LC}, T^*=T_{LC}$	$T_1=T_{LC}, T_2=T_{LC}, T^*=T_{LC}$

주)

①: $\frac{C_3}{\alpha\alpha(C_2 - C_1)} > 1$

②: 1) $\frac{C_3}{\alpha\alpha(C_2 - C_1)} > \exp[-\alpha\gamma(1 - e^{-\beta T_{LC}})]$

③: $\frac{C_3}{\alpha\alpha(C_2 - C_1)} < \exp[-\alpha\gamma(1 - e^{-\beta T_{LC}})]$

④: $R(x|0) > R_0$

⑤: $R_0 > R(x|0) > R_0 \exp(e^{\beta T_{LC}})$

⑥: $R(x|0) < R_0 \exp(e^{\beta T_{LC}})$

이 표를 비용의 입장에서 고찰해보기로 한다.

$$\frac{C_3}{\alpha\alpha(C_2 - C_1)} > 1$$

이면 비용이 단조증가하는 경우로서 결합테스트를 하면 할수록 비용이 증가되어 테스트 없이 발행하는 것이 최적인 것을 의미한다. 그러나, 이러한 경우가 현실적으로 가능한가 하는 것인가를 연구해보아야 한다.

$$1) \frac{C_3}{\alpha\alpha(C_2 - C_1)} > \exp[-\alpha\gamma(1 - e^{-\beta T_{LC}})]$$

는 비용최저점이 결합테스트와 소프트웨어의 전 수명기간 사이에 존재하는 경우이다. 이러한 경우는 목표신뢰도를 만족시키는 발행시기와 총 비용을 최적으로 하는 발행시기 중 큰 값을 취하는 것이 이상적이다.

$$\frac{C_3}{\alpha\alpha(C_2 - C_1)} < \exp[-\alpha\gamma(1 - e^{-\beta T_{LC}})]$$

는 비용이 단조감소하는 경우이다. 이러한 경우는 결합테스트를 하면 할수록 총 비용이 감소되는 것으로서 현실적으로 있을 수 없는 경우에 속한다.

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

$$R(x|0) > R_0$$

는 소프트웨어 개발 즉시 신뢰도가 목표신뢰도를 만족하기 때문에 더 이상 결합 발견을 위한 테스트를 할 필요가 없는 경우로서 비용을 최적으로 하는 시기에 맞추어 발행시기를 결정해야 하는 경우이다. 그러나, 이러한 경우는 현실적으로 가능하지 않다고 보는 것이 타당할 것이다.

$$R_0 > R(x|0) > R_0 \exp(e^{\beta T_{LC}})$$

는 소프트웨어를 개발한 후 테스트 및 결합 수정을 통하여 목표신뢰도를 만족시키는 경우이다. 이러한 경우는 목표신뢰도를 만족시키는 발행시기와 총 비용을 최적으로 하는 발행시기 중 큰 값을 취하는 것이 이상적이다.

$$R(x|0) < R_0 \exp(e^{\beta T_{LC}})$$

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

5. 결론

소프트웨어의 신뢰도에 대한 정의를 하고, 테스트 시간의 경과와 신뢰도와의 관계, 결합 수정 후 경과되는 시간과 신뢰도와의 관계를 연구하였다. 소프트웨어의 수정비용을 고찰함에 있어서 테스트 기간중의 결합 수정 비용, 운영 기간중의 결합 수정 비용, 테스트 기간중의 테스트 비용으로 세분하여 검토한 후 이들을 결합하여 비용이 최소로 되는 시점을 발행시각으로 검토하였다. 목표신뢰도 입장에서 발행시각을 결정함에 있어서 개발 후 테스트를 시작하기 전의 신뢰도 $R(x|0)$ 가 어떠한 조건에 있는가를 검토하여 각 조건에 따른 최적 발행시각

을 결정하였다. 비용의 입장에서 발행시각을 결정하는 문제와 신뢰도의 입장에서 발행시각을 결정하는 문제를 동시에 고려하여 최적 발행시각을 결정하도록 하였으며, 각각의 조건 및 한계를 연구하였다. 이 결과를 표 1에 나타내었다.

Acknowledgement
본 연구는 한국과학재단 목적기초연구(R01-2000-00273) 지원사업으로 수행되었음

참고문헌

[1] K. Okumoto, A. L. Goel, "Optimum release time for software systems based on reliability and cost criteria", J. System software, vol. 1, 1980, pp315-318

[2] S. Yamada, S. Osaki, "Cost-reliability optimal release policies for software systems", IEEE Trans. on Reliability, vol. R-34, 1985 Dec., pp422-424

[3] 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, 1996 Dec., pp646-651

[4] Amrit L. Goel, Kazu Okumoto, "Time - Dependent Error - Detection Rate Model for Software Reliability and Other Performance Measure", IEEE Trans. on Reliability, vol R-28, No.3, 1979.8. pp206-211