

소프트웨어 발행의 문제점에 관한 연구

최규식, 김용경

건양대학교

che@konyang.ac.kr

충남 논산시 내동 26

041-730-5283

키워드 : 비용-신뢰도, NHPP, 평균치함수, 최적발행, SRGM

1. 서론

컴퓨터시스템은 여러 가지 복잡하고 민감한 시스템을 제어하는데 광범위하게 쓰이고 있다. 최근에 와서는 운영시스템, 제어프로그램, 적용프로그램과 같은 여러 가지 소프트웨어 시스템이 더욱더 복잡화 및 대형화되고 있기 때문에 신뢰성이 높은 소프트웨어 시스템을 개발하는 일이 매우 중요하며, 소프트웨어 제품 개발에 있어서 소프트웨어의 신뢰도가 핵심사항이다. 1970년대 이후 소프트웨어의 신뢰성을 향상시키기 위한 여러 가지 소프트웨어의 신뢰도 모델이 제시되고 검토되었으며, 특히, 소프트웨어 개발 후 테스트단계에 적용하는 신뢰도를 추정하고 예측하는 모델이 많이 개발되었다. 소프트웨어가 주어진 시간간격동안 고장이 발생하지 않을 확률 즉, 신뢰도는 소프트웨어의 테스트과정을 계속해서 반복 및 수정하면 더욱 더 증가된다. 그러한 결함검출현상을 설명해주는 소프트웨어 신뢰도 모델을 소프트웨어 신뢰도 성장 모델(SRGM)이라 한다.

본 논문에서는 지금까지 연구된 비용과 신뢰도에 대한 두 개의 기준을 동시에 고려하여 발행시기를 결정하는 최적발행정책에 대해서 연구해보고, 특히, 이러한 연구결과의 한계에 대해서 연구하고자 한다. 연구 방법으로서 소프트웨어 신뢰도를 만족시키는 조건하에서 전체평균 소프트웨어 비용을 최소화시키는 최적 소프트웨어 발행정책을 검토하며, 이러한 최적 비용-신뢰도 발행정책에 대한 연구가 현실적으로 타당한 것인가, 일반적으로 적용될 수 있는 합리성을 가지고 있는가를 연구한다.

$N(t)$: 시각 t 까지 검출되는 소프트웨어의 누적에러 갯수

a : 초기부터 소프트웨어내에 존재하고 있는 에러의 갯수

b, b_1, b_2 : 에러검출비, 일반적인 경우, 테스트기간중, 발행후 운전기간중, $b_1 \geq b_2$

$m(t)$: $E[N(t)]$, 평균치 함수

$R(x|t)$: 소프트웨어의 신뢰도, 시각 $t(x \geq 0)$ 에서 에러가 검출된 후 $(t, t+x)$ 에서 고장이 일어나지 않을 확률

R_0 : 목표신뢰도, $0 < R_0 < 1$

c_1 : 테스트기간중에 검출되는 에러를 수정하는 단위비용, $c_1 > 0$

c_2 : 운전기간에 검출되는 에러를 수정하는 단위비용, $c_1 > c_2$

c_3 : 단위시간당 테스트 비용, $c_3 > 0$

$C(T)$: 전체 평균 소프트웨어 비용

T_{LC} : 소프트웨어의 수명주기

T : 전체 테스트 시간

T^* : 최적 소프트웨어 테스트 시간

T_1 : $dC(T)/dT=0$ 을 만족시키는 유일 해 T

T_2 : $R(x/T)=R_0$ 를 만족시키는 유일 해 T

2. SRGM

소프트웨어 개발단계에서 테스트하는 소프트웨어 시스템에 대해서 고찰해보기로 한다. 소프트웨어 고장은 “시스템 내에 잔존하는 에러에 의해 프로그램 운전이 원치 않게 중지되는 것”으로 정의한다.

소프트웨어의 테스트에 의해서 검출되는 누적에러의 수를 기술할 때, 일반적으로 에러의 검출시간단위로써 역일시간(calendar time)이나 장비의 실행시간과 같은 테스트시간을 사용한다. NHPP의 평균치함수로 불리는 $M(t)$ 의 기대치는 $m(t)$ 로 정의한다. NHPP에 근거한 SRGM을 아래와 같이 정의한다.

$$\Pr\{N(t) = n\} = \frac{\{m(t)\}^n}{n!} \exp[-m(t)] \quad (1)$$

여기서,

$$m(t) = a[1 - \exp(-bt)], \quad a > 0, \quad b > 0, \quad t \geq 0 \quad (2)$$

이다. $a (= m(\infty))$ 를 최종적으로 검출될 기대누적에러의 수 즉, 추정할 최초의 기대에러라고 정의하면 다음과 같은 식을 유도할 수 있다.

$$\lim_{t \rightarrow \infty} \Pr\{N(t) = n\} = \frac{a^n}{n!} \exp(-a) \quad (3)$$

이는 $N(t)$ 가 오랜기간동안 테스트를 한 후 평균치 a 를 가진 Poisson을 따른다는 것을 의미한다.

$m(t)$ 를 가진 NHPP로부터 유도되는 정량적 신뢰도척도의 하나는 소프트웨어 신뢰도이다. 소프트웨어 신뢰도는 다음과 같이 정의한다.

$$R(x,t) = \exp[m(t+x) - m(t)] \quad (4)$$

여기서, $m(t+x) - m(t) = a(1 - e^{-b(t+x)}) - a(1 - e^{-bt})$
 $= ae^{-bt}(1 - e^{-bx}) = m(x)e^{-bt}$

이므로,

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

이다.

3. 소프트웨어의 수정 비용

Okumoto와 Goel은 소프트웨어 신뢰도 성장모델을 이용하여 비용 및 신뢰도기준에 근거한 최적 소프트웨어 발행정책을 연구하였으며, Yamamoto와 Osaki는 비용과 신뢰도를 동시에 고려하여 소프트웨어 시스템의 최적발행시각을 결정하고자 하였다. Rong-Huei Hou, Sy-Yen Kuo, Yi-Ping Chang은 이러한 연구 결과를 지수형 성장곡선과 로지스틱 성장곡선에 적용하는 연구를 하였다.

3.1 테스트 기간중의 수정비용

테스트 기간중에 검출되는 총 에러의 수에 에러당 수정비용을 곱한 값이 된다.

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

3.2 운전 기간중의 수정비용

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

3.3 테스트 기간중의 테스트 비용

전 테스트기간에 단위시간당 테스트비용을 곱한 값이 된다.

$$c_3 T \quad (8)$$

3.4 총비용

상기와 같은 논리에 의해서 총 비용은 테스트기간중의 에러수정비용, 운전기간중의 에러수정비용, 테스트 기간중의 테스트비용을 합한 값이 된다.

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

수치해석적인 반복법에 의하여 최적발행시각 $T=T_1$ 을 구한다. 특별히 소프트웨어 발행 전후의 에러검출 비율이 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.

즉, (9b)에서 최소값을 구하기 위한 조건은

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

이다. 그런데, $T_1 > 0$ 이므로, $\ln \frac{ab(c_2 - c_1)}{c_3} > 0$ 즉, $ab(c_2 - c_1) > c_3$ 일 때에만 (10a)가 성립되며, 그 외에는 구할 수 없는 외부 최소값이 존재하여 그 경계치

$$T_1 = 0 \quad (10b)$$

에서 최소가 된다.

마찬가지로, 유사한 내부해법이 존재하여 신뢰도를 요건에 최근접시키는 유일한 시각이 존재한다. 최적 소프트웨어 발행시각은 미리 규정된 소프트웨어 신뢰도 $R(x, T) = R_0$ 를 만족시키는 최근접이 되는 시각이다.

여기서, x 는 최근에 검출된 에러를 수정한 후 경과되는 시간이다.

발행시각 T 에서의 신뢰도는 (5)에서

$$R(x, T) = \exp[-a(1 - e^{-b_2 x})e^{-b_1 T}] = R_0 \quad (11)$$

$$T_2 = \frac{1}{b_1} \left[\ln m(x) - \ln \left(\ln \frac{1}{R_0} \right) \right] \quad (12)$$

이고, 여기서 R_0 는 소프트웨어가 추구하는 목표신뢰도이다.

특히, $b_1 = b_2 = b$ 인 경우는 단순화되어

$$T_2 = \frac{1}{b} \left\{ \ln m(x) - \ln \left[\ln \frac{1}{R_0} \right] \right\} \quad (13a)$$

이다. 그런데,

$$R(x|0) = \exp[-m(x)] = \exp[-a(1 - e^{-bx})]$$

로서, x 가 증가함에 따라 $R(x|0)$ 가 감소한다. 즉, $R(x|0) < R_0$ 일 때에만 (13a)가 성립되며, 그 외에는 구할 수 없는 외부 해가 존재하며, 경계치

$$T_2 = 0 \quad (13b)$$

에서 최근접이 된다.

4. 최적 소프트웨어 발행

소프트웨어 테스트로부터 구한 소프트웨어의 신뢰도를 어떤 규정치로 유지하는 제한 하에 전체 평균 소프트웨어의 비용을 최소로 하는 최적 소프트웨어 발행정책에 대해서 고려해보기로 한다. 최적 소프트웨어 발행문제는 아래와 같이 공식화할 수 있다.

$$\begin{aligned} c_2 > 0, c_1 > c_3 > 0, x \geq 0, 0 < R_0 < 1 \text{인 경우에 대해서} \\ R(x|T) \geq R_0, T \geq 0 \text{인 조건하에 } C(T) \text{를 최소화} \end{aligned} \quad (14)$$

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

$$T^* = \max\{T_1, T_2\} \quad (15)$$

여기서, T_1 은 (10)에서, T_2 는 (13)에서 구한 값이다.

- 1) $ab > c_3/(c_2 - c_1)$ 이고 $R(x|0) < R_0$ 이면 (10)과 (13)을 만족시키는 양의 유일한 T_1 과 T_2 가 각각 존재한다.
- 2) $ab > c_3/(c_2 - c_1)$ 이고 $R(x|0) \geq R_0$ 이면 $T^* = T_1$
- 3) $ab \leq c_3/(c_2 - c_1)$ 이고 $R(x|0) < R_0$ 이면 $T^* = T_2$
- 4) $ab \leq c_3/(c_2 - c_1)$ 이고 $R(x|0) \geq R_0$ 이면 $T^* = 0$

여기서, 식(10)과 식(13)을 근거로 하여 도출된 식(15)가 현실적으로 타당한가를 연구해보고자 한다. 상기식이 타당성을 가지려면 현실적으로 적용가능한 식이 되어야 한다. 비용-신뢰도 발행문제에 있어서 가장 이상적인 것은 그림 1에서 보는 바와 같이 목표신뢰도를 만족시키면서 소프트웨어 전 수명기간동안의 수정 비용이 최소가 되는 경우 즉, 최적치 연구에서 $T_1 > T_2$ 가 되어 $T^* = T_1$ 로 결정되는 경우이다.

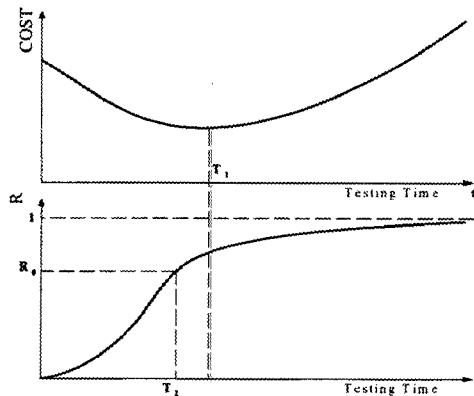


그림 1 비용-신뢰도 최적관계 곡선

문제를 단순화시키기 위해서 특별히 테스트기간중의 에러검출비와 운전기간중의 에러검출비가 같다고 가정하여 $b_1=b_2=b$ 라고 하고 우선 상기 조건 1)을 고찰해보기로 한다.

$R(x) < R_0$ 인 조건 하에서 $T_1 > T_2$ 인 경우를 고찰하는 것이므로, 상기 조건을 만족하려면

$$\frac{1}{b} \ln \frac{ab(c_2 - c_1)}{c_3} > \frac{1}{b} \left\{ \ln \frac{m(x)}{\ln \frac{1}{R_0}} \right\} \quad (16)$$

로서

$$\frac{ab(c_2 - c_1)}{c_3} > \frac{m(x)}{\ln \frac{1}{R_0}} \quad (17)$$

이고, $R(x) < R_0$ 인 조건을 고려하여 이 식을 정리하면

$$\exp[-m(x)] < R_0 < \exp\left[-\frac{c_3}{ab(c_2 - c_1)} m(x)\right] \quad (18)$$

이 된다. 즉, 조건 1)을 만족시키면서 $T_1 > T_2$ 가 되려면 식(18)과 같은 목표신뢰도의 제한을 받게 된다.

그리고, 조건 2)의 경우는 $T_2=0$ 즉, 소프트웨어가 개발완료되는 순간의 신뢰도가 목표신뢰도를 만족하기 때문에 테스트할 필요 없이 곧바로 소프트웨어를 발행하는 경우이다. 이 때에는 T_1 의 결정이 의미를 가진다고 볼 수 있다. 그러나, 이 때에도 $R(x) \geq R_0$ 로부터

$$R_0 \leq \exp[-m(x)] \quad (19)$$

의 목표신뢰도 제한을 받는다. 따라서, 목표신뢰도를 만족시키면서 전 수명기간동안의 소프트웨어 수정비용이 최소가 되는 경우 즉, $T^* = T_1$ 으로 되는 때는 $T_2=0$ 인 경우뿐이고, 그 신뢰도 또한 식(19)와 같은 제한을 받게 된다. 그리고, 소프트웨어가 개발되자마자 테스트 없이 곧바로 발행된다는 것도 현실에 맞지 않는다.

조건 1)과 2)를 만족시킬 수 있는 목표신뢰도의 한계치를 고찰해보기 위해서 [1]에서 인용한 모델파라미터인 $a=1348$, $b=0.124$, $x=0.1$, $R_0=0.7$, $c_1=1$, $c_2=5$, $c_3=100$, $T_{LC}=100$ 에 적용해보기로 한다.

이러한 값들을 이용하여 $C(T)$ 를 최소화시키는 비용최소화 기준과 $R(x/T)=R_0$ 를 만족시키면서 소프트웨어 전 수명기간동안의 수정비용이 최소가 되는 경우 즉, 최적치 연구에서 $T_1 > T_2$ 가 되어 $T^* = T_1$ 로 결정되는 경우의 목표신뢰도 한계를 고찰해보면, 각각에 대해서 $\exp[-m(x)] \rightarrow 0$, $\exp\left[-\frac{c_3}{ab(c_2 - c_1)} m(x)\right] \rightarrow 0$ 이므로, $R_0 \approx 0$ 이다. 즉, 비용을 최소로 하면서 원하는 정도의 목표신뢰도를 얻는다는 것은 어렵다. 따라서, 일반적으로 통용되는 목표 신뢰도, 예를 들면 $R_0=95\%$ 와 같은 높은 정도의 신뢰도를 얻으려면, 그 최적발행시각이 최저비용에 의해서 결정되는 것이 아니고 목표신뢰도에 의해서 결정되기 때문에, 비용을 최저로 하고자 한 [4]와 [5]의 연구결과는 재검토되어야 한다고 본다.

상기와 같은 알고리즘에 의해 비용-신뢰도를 산출할 경우, 원하는 목표 신뢰도를 얻고 동시에 발행비용을 최소로 하기 위해서는 소프트웨어를 이상적으로 잘 개발하여 소프트웨어 내에 에러의 수가 현저하게 적어지게 하거나, 에러의 검출비가 낮아서 $b \rightarrow 0$ 로 되도록 해야 한다.

5. 결론

소프트웨어 시스템을 개발하여 발행하기 전에 소프트웨어 시스템 내에 존재하고 있는 에러를 검출하여 수정하기 위한 테스트를 수행할 때, 언제 테스트를 끝내고 발행해야 하는가 하는 문제가 발생하게 된다. 이러한 최적 발행시기를 결정하기 위해 여러 연구자들이 전 수명기간중의 투입비용과 신뢰도를 동시에 고려

하여 연구를 수행하였다.

본 논문에서는 이러한 발표논문을 중심으로 하여 소프트웨어의 비용-신뢰도 최적발행시기 결정방법에 관한 한계를 연구하였다. 비용-신뢰도 발행 문제에 있어서 가장 이상적인 것은 목표신뢰도를 만족시키면서 수명기간 동안의 소프트웨어 수정비용이 최소가 되도록 발행시기를 결정하는 것이다. 그러나, 연구 결과 이러한 경우는 목표신뢰도가 비현실적으로 극히 낮은 경우이거나, 소프트웨어가 개발되는 순간의 신뢰도가 목표신뢰도를 만족하여 테스트할 필요 없이 곧바로 소프트웨어를 발행하는 경우뿐인 것으로 나타났다. 이러한 경우는 극히 발생확률이 적은 비현실적인 경우에 해당된다. 그 외에는 전 수명기간동안의 수정비용과 관계 없이 목표신뢰도가 항상 소프트웨어의 발행시기를 결정하게 되므로 비용-신뢰도 최적발행에 대한 그 동안의 연구를 재검토해볼 필요가 있다.

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. H.S. Koch, P. Kubat, "Optimal release time of computer software", IEEE Trans. Software Eng'g, vol SE-9, 1983 May, pp323-327
3. M. Xie, "On the determination of optimum software release time", Proc. 2nd Int' Symp. Software Reliability Eng'g, 1991, pp218-224
4. S. Yamada, S. Osaki, "Cost-reliability optimal release policies for software systems", IEEE Trans. Reliability, vol R-34, 1985 Dec., pp422-424
5. Rong-Huei Hou, Sy-Yen Kuo, Yi-Ping Chang, "Optimal Release Policy for Hyper-Geometric Distribution Software-Reliability Growth Model", IEEE Trans Reliability, vol 45, 1996 Dec., pp646-651
6. S. Yamada, H. Ohtera, H. Narihisa, " Software Reliability Growth Models with Testing-Effort", IEEE Trans Reliability, vol R-45, 1986 April, pp19-23