

테스트노력을 고려하지 않은 소프트웨어의 최적발행

A Study on the Optimum Software Release with without Testing Efforts

최규식

(Che, Gyu Shik)

Abstract - The software reliability is defined, and not only the relations between testing time and reliability, but also the relation 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 evaluating the cost for each condition. Also, the release time is determined depending on the conditions of the first reliability, considering the specified reliability. The optimum release time is determined by simultaneously studying two optimum release time issues that determine both the cost related time and the specified reliability related time. And, each condition and limitation are studied. The trend of the optimum time is also examined.

Key Words : SRGM, NHPP, mean value function, target reliability, fault detection, optimal release time

1. 서론

그동안 많은 연구가와 참여자들에 의해서 보편적으로 널리 연구되고 사용되는 SRGM(software reliability growth model)의 부류는 NHPP(nonhomogeneous Poisson process) 모델이며, 이러한 부류의 모델은 실제로 많은 장점을 가지고 있어서 그간 많은 관심을 끌어들였다.

Okumoto와 Goel[1]은 전체평균 소프트웨어 비용을 최소화시키는 비용-최적 SRP(software reliability process)를 발표하였다. Yamada와 Osaki[2]는 전체 평균 비용을 최소화시키고 소프트웨어 신뢰도를 만족시키는 전체평균비용-신뢰도-최적 SRP를 도입하였다. 이러한 연구결과를 참조하여 Hou, Kuo, Chang[3]은 지수 곡선과 로지스틱 곡선에 적용하는 연구를 수행하였다.

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.[2]

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

이는 소프트웨어를 개발하여 결함검출 테스트를 시작한 후 계속 s 유니트 시간 동안 (k-1) 번째 결함을 발견하여 수정한 후 k 번째 결함이 발견되기 전까지 운영될 x 시간 동안에 고장 없이 소프트웨어가 동작할 확률인 것이다. 그림 1 참조.

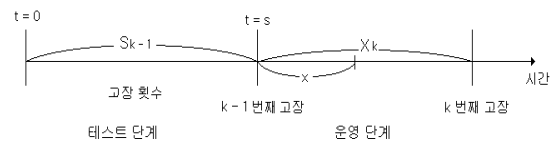


그림 1 고장발생 표현

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

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

로 정의할 때[4] 임의의 $t \geq 0$ 과 $x > 0$ 에서

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

이므로, 식(1)의 신뢰도는 다음과 같이 표현할 수 있다.[4]

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

식(4)로 표시된 신뢰도의 특성을 이해하기 위해 테스트 시간과 신뢰도의 관계 및 최종 검출 결함 수정 후 경과시간과 신뢰도의 관계를 그림으로 나타내면 그림 2, 그림 3과 같다.

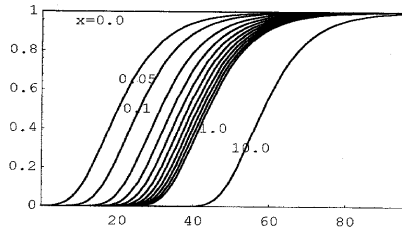


그림 2 발행시각과 신뢰도와와의 관계

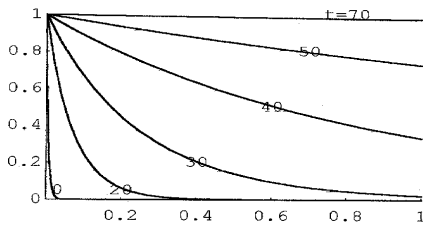


그림 3 결함 수정후 경과시간과 신뢰도와와의 관계
상기 식(4)에서 정의한 테스트 단계의 신뢰도 의미를 고찰해보기로 한다.

원하는 목표신뢰도를 R_0 라 하면 식(4)로부터

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

가 되어야 한다.[2]

3. 수정 비용

테스트 기간중의 결함 수정 비용은 검출된 결함 하나하나를 수정하는데 비용이 발생되므로, 테스트 기간중에 검출되는 총 결함의 수에 결함당 수정비용을 곱한 값이 된다.[2]

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

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.[2]

$$C(T) = c_1 m_1(T) + c_2 \{m_2(T_{LC}) - m_1(T)\} + c_3 T$$

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

최적 소프트웨어 발행시각은 전체 평균 소프트웨어 비용을 최소화 하는 테스트시간이다.

$$(c_1 - c_2)ab_1 e^{-b_1 T} + c_2 a(b_1 - b_2)e^{-b_2 T_{LC}} e^{-(b_1 - b_2)T} + c_3 = 0$$

이 식을 만족시키는 $T > 0$ 인 범위의 T 값을 구하면 $C(T)$ 에 대한 최소값이 된다. 특별히 소프트웨어 발행 전후의 결함검출비율이 같을 경우에는 $b_1 = b_2 = b$ 가 되기 때문에 상기식이 단순화되어 이 때 식(9a)는

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

(9b)

와 같이 된다.[2]

(9b)에서

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

(10)

발행시각 T 에서의 신뢰도는 식(5)와 같으므로 여기서 목표신뢰도를 만족시키는 발행시각을 구하면 다음과 같다.

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

(11a)

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 최소화 하는 최적 소프트웨어 발행정책에 대해서 고려하기로 한다. 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

$$R(x|T) \geq R_0, T \geq 0 \text{인 조건하에 } C(T) \text{를 최소화} \quad (12)$$

이러한 방법으로 하여 비용-신뢰도최적 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.

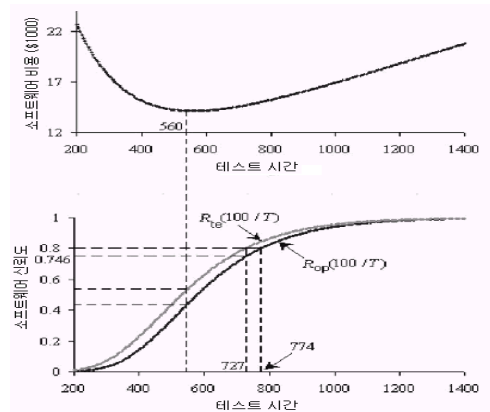


그림 4 비용-신뢰도 곡선

4.1 비용을 고려한 발행시각

그림 5 참조.

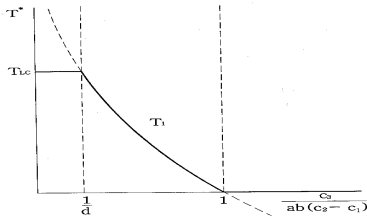


그림 5. 비용과 발행시각과의 관계

4.2 목표신뢰도를 고려한 발행시각

그림 6 참조.

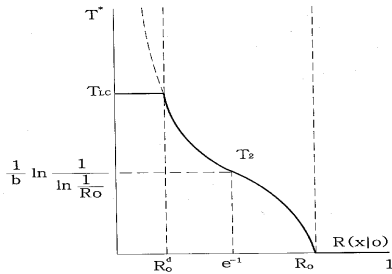


그림 6 목표신뢰도와 발행시각과의 관계

4.3 최적발행시각

이상의 내용을 요약하면 다음 표 1과 같다.

표1 개발 소프트웨어의 발행시각 결정

신뢰도	비용	T_1		
		$\frac{c_3}{ab(c_2-c_1)} > 1$	$1 > \frac{c_3}{ab(c_2-c_1)} > \frac{1}{d}$	$\frac{c_3}{ab(c_2-c_1)} < \frac{1}{d}$
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

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.

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

는 비용최저점이 결함테스트와 소프트웨어의 전 수명기간 사이에 존재하는 경우이다.

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

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 최저로 하는 시기에 맞추어 발행시각을 결정해야 하는 경우이다.

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

는 소프트웨어를 개발한 후 테스트 및 결함 수정을 통하여 목표신뢰도를 만족시키는 경우이다.

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.

비용면에서 볼 때

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

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

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

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

한편, 목표신뢰도면에서 볼 때

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

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

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

제한다.

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

$R(x|0) < R_o \exp(e^{bT_{LC}})$ 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: $\frac{C_3}{aa(C_2 - C_1)} > 1$

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

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

④: $R(x|0) > R_o$

⑤: $R_o > R(x|0) > R_o \exp(e^{bT_{LC}})$

⑥: $R(x|0) < R_o \exp(e^{bT_{LC}})$

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

이때 비용이 단조증가하는 경우로서 결합테스트를 하면 할수록 비용이 증가되어 테스트 없이 발행하는 것이 최적인 것을 의미한다.

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

는 비용최저점이 결합테스트와 소프트웨어의 전 수명기간 사이에 존재하는 경우이다.

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

는 비용이 단조감소하는 경우이다.

마찬가지로 신뢰도의 입장에서 고찰해보기로 한다.

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

$R(x|0) > R_o$

는 소프트웨어 개발 즉시 신뢰도가 목표신뢰도를 만족하기 때문에 더 이상 결합 발견을 위한 테스트를 할 필요가 없는 경우로서 비용을 최저로 하는 시기에 맞추어 발행시기를 결정해야 하는 경우이다.

$R_o > R(x|0) > R_o \exp(e^{bT_{LC}})$

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

$R(x|0) < R_o \exp(e^{bT_{LC}})$

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 발행시각을 결정하였다. 비용의 입장에서 발행시각을 결정하는 문제와 신뢰도의 입장에서 발행시각을 결정하는 문제를 동시에 고려하여 최적 발행시각을 결정하도록 하였으며, 각각의 조건 및 한계를 연구하였다. 이 결과를 표에 나타내었다. 각각을 만족시키는 발행시각이 $T_1 > T_2$ 로 되어 $T^* = T_1$ 로 결정되는 경우가 가장 이상적이다. 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] 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
 [5] Xuemei Zhang, Hoang Pham, "An analysis of factors affecting software reliability", The Journal of Systems and Software, 2000. pp43-56