

O테스트가 균일한 소프트웨어의 발행 신뢰도

최규식, 장원석

건양대학교 IT학부

전화 : 041-730-5290 / 핸드폰 : 011-446-5290

Che Gyu Shik, Chang Won Seok

Konyang University

E-mail : wschang@konyang.ac.kr

1. 서론

본 논문에서는 소프트웨어의 신뢰도에 대한 정의를 하고, 테스트 시간의 경과와 신뢰도와 관계, 결합 수정 후 경과되는 시간과 신뢰도와 관계를 연구한다. 소프트웨어의 비용을 고찰함에 있어서 조건별로 검토하여 비용이 최소로 되는 발행시각을 결정한다. 목표 신뢰도 입장에서 발행시각을 결정함에 있어서 개발 후 테스트를 시작하기 전의 신뢰도의 조건에 따른 최적발행시각을 정한다. 비용의 입장에서 발행 시각을 결정하는 문제와 신뢰도의 입장에서 발행 시각을 결정하는 문제를 동시에 고려하여 최적 발행시각을 결정하도록 한다. 그리고, 각각의 조건 및 한계를 연구한다. 이상적인 범위의 조건일 때에 최적 발행시각의 경향이 어떠한 추세를 보이는지 연구한다.

2 소프트웨어의 수정 비용

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

$$c_1 m_1(T) = c_1 a(1 - e^{-b_1 T}) \quad (2.1)$$

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

$$c_2 \{m_2(T_{LC}) - m_1(T)\} = c_2 a e^{-b_1 T} (1 - e^{-b_2(T_{LC} - T)}) \quad (2.2)$$

와 같이 표현된다.

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

$$c_3 T \quad (2.3)$$

총 비용은 테스트기간중의 결합 수정 비용, 운영기간중의 결합 수정 비용, 테스트기간중의 테스트 비용을 합한 값이 된다.

$$C(T) = c_1 m_1(T) + c_2 \{m_2(T_{LC}) - m_1(T)\} + c_3 T \quad (2.4a)$$

최적 소프트웨어 발행시각은 전체 평균 소프트웨어 비용을 최소로 하는 테스트시간이다.

특히 소프트웨어 발행 전후의 결합검출비율이 같을 경우에는 $b_1 = b_2 = b$ 가 되기 때문에 상식식이 단순화되어 이 때 식(4.4a)는

$$C(T) = c_1 m(T) + c_2 \{m(T_{LC}) - m(T)\} + c_3 T \quad (2.4b)$$

와 같이 되므로, 다음과 같은 식이 유도된다.

$$T_1 = -\frac{1}{b} \ln \frac{c_3}{ab(c_2 - c_1)} \quad (2.5)$$

3 소프트웨어의 신뢰도 요건

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

발행시각 T 에서의 신뢰도는 다음과 같다.

$$T_2 = \frac{1}{b_1} [\ln m(x) - \ln(\ln \frac{1}{R_0})] \quad (3.1a)$$

특히, $b_1 = b_2 = b$ 인 경우는 단순화되어

$$T_2 = \frac{1}{b} \left\{ \ln \frac{m(x)}{\ln \frac{1}{R_0}} \right\} \quad (3.1b)$$

로 된다.

4 비용-신뢰도 최적 소프트웨어 발행

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

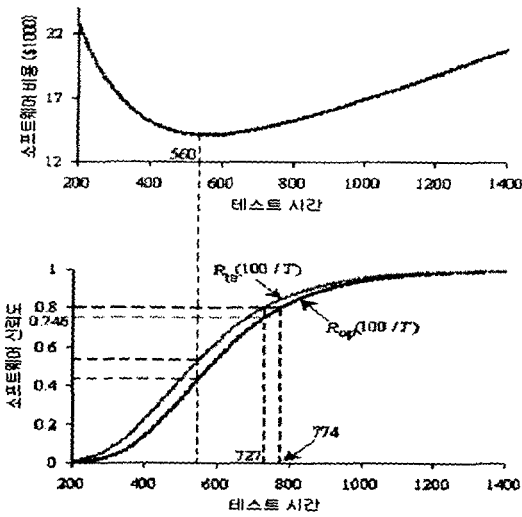


그림 4-1 비용-신뢰도 곡선

3.1 비용을 고려한 발행시각

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

에서 양의 유일 해 $T^* = T_I$ 이 존재한다.

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

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

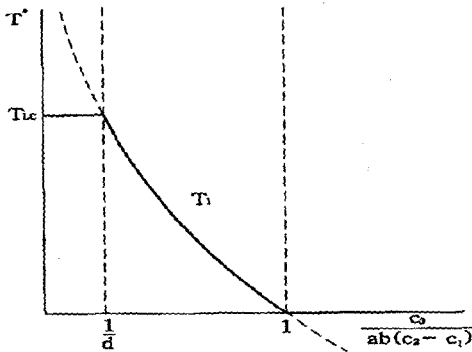


그림 4-2 비용과 발행시각과의 관계

4.2 목표신뢰도를 고려한 발행시각

1) $R_0 > R(x|0) > R_0^d$ 에서 양의 유일 해 $T^* = T_2$ 가 존재한다.

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

3) $R(x|0) < R_0^d$ 이면 $T_2 > T_{Ic}$ 이므로 $T^* = T_2 = T_{Ic}$ 이다

1)의 경우에 대해서 좀더 검토해보기로 한다.

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

으로 되어

$$T_2 = \frac{1}{b} \ln \left\{ \frac{\ln \frac{1}{R(x|0)}}{\ln \frac{1}{R_0}} \right\}$$

로도 표현할 수 있다.

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

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

이 되어 함수의 변곡점은 목표신뢰도 R_0 의 값에 관계 없이 항상 $t = R(x|0) = \frac{1}{e}$ 인 값으로 결정된다. 이때

$$T_2 = \frac{1}{b} \left\{ \ln \frac{1}{R_0} \right\}$$

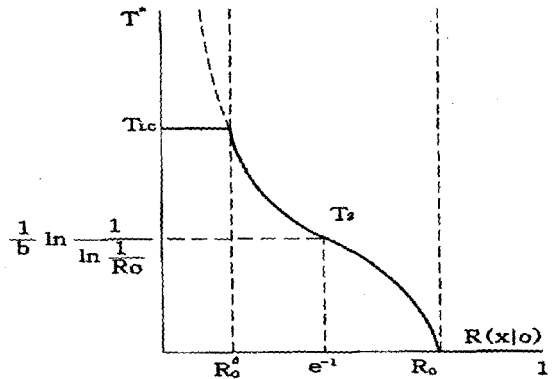


그림 4-3 목표신뢰도와 발행시각과의 관계

4.3 최적발행시각

이상의 내용을 요약하면 다음과 같다.

(1) 비용

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

이면 비용이 단조증가하는 경우로서 결합테스트를 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 결합테스트와 소프트웨어의 전 수명 기간 사이에 존재하는 경우이다. 이러한 경우는 목표 신뢰도를 만족시키는 발행시기와 총 비용을 최저로 하는 발행시기 중 큰 값을 취하는 것이 이상적이다.

$$3) \frac{C_3}{ab(c_2 - c_1)} < \frac{1}{d}$$

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

(2) 신뢰도

$R(x|0)$ 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 결합 발견을 위한 테스트를 할 필요가 없는 경우로서 비용을 최저로 하는 시기에 맞추어 발행시기를 결정해야 하는 경우이다. 그러나, 이러한 경우는 고려하기 어려운 경우이다.

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

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.

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 비용면에서

$$1) \frac{C_3}{ab(c_2 - c_1)} > \frac{1}{d}$$

이고 목표신뢰도 면에서

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

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 그림4-2, 4-3에 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 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

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

이므로

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

인 범위 즉,

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

에 서 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.

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

이 면 $T_1 > T_{LC}$ 이므로 $T^* = T_1 = T_{LC}$ 이다.

한편, 목표신뢰도면에서 볼 때

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

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

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

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

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

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

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 수명 기간 사이에 존재하는 경우이다. 이러한 경우는 목표 신뢰도를 만족시키는 발행시기와 총 비용을 최저로 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.

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

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 가능하지 않다고 보는 것이 타당할 것이다.

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

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.

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

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

참고 문헌

- [1] 강인선, 김진규, "신뢰성 공학", 한울출판사, 1997, pp183-204
- [2] 이지우, 김선진, 이성우, 정상영, "신뢰성 공학", 원창출판사, 1993, pp21-124
- [3] H. Ascher and H. Feingold, *Repairable Systems Reliability: Modeling, Inference, Misconceptions, and Their Causes*. New York: Marcel Dekker, 1984.
- [4] W. D. Brooks and R. W. Motley, "Analysis of discrete software reliability models," Rome Air Development Center, New York, Tech. Rep. RADC-TR-80-84, 1980.
- [5] E. H. Forman and N. D. Singpurwalla, "Optimal time intervals for testing hypotheses on computer software errors." *IEEE Trans. Rel.*, vol. R-28, pp. 250-253, Aug. 1979.
- [6] A. L. Goel and K. Okumoto, "Time-dependent error-detection rate model for software reliability and other performance measures," *IEEE Trans. Rel.*, vol. R-28, pp 206-211, Aug. 1979.
- [7] H. S. Koch and P. Kubat, "Optimal release time of computer software," *IEEE Trans. Software Eng.*, vol. SE-9, pp. 323-327, May 1983.
- [8] B. Littlewood, "Theories of software reliability" How good are they and how can they be improved?" *IEEE Trans. Software Eng.*, vol. SE-6, pp. 489-500, Sept. 1980.
- [9] J. D. Musa, "A theory of software reliability and its application," *IEEE Trans. software Eng.*, vol. SE-1, pp. 312-327, Sept. 1975.
- [10] W. Nelson, *Applied Life Data Analysis*. New York: Wiley, 1982.
- [11] M. Ohba, "Software quality = test coverage test accuracy," in *Proc. IPS-Japan WGSE Meet.* (in Japanese), vol. 21, 1981.
- [12] K. Okumoto and A. L. Goel, "Optimum release time for software systems based on reliability and cost criteria," *J. Syst. Software*, vol.1, pp. 315-318, 1980.
- [13] C. V. Ramamoorthy and F. B. Bastani, "Software reliability-Status and perspectives," *IEEE Trans. Software Eng.*, vol SE-*, pp. 354-371, Aug. 1982.
- [14] S. M. Ross, *Stochastic Processes*. New York: Wiley, 1983.
- [15] S. Yamada, M. Ohba, and S. Osaki, "S-shaped reliability growth modeling for software error detection," *IEEE Trans. Rel.*, vol. R-32, pp. 475-478, 484, Dec. 1983.
- [16] S. Yamada, H. Narihisa, and S. Osaki, "Optimum release policies for a software system with a scheduled software delivery time," *Int. J. Syst. Sci.*, vol. 15, no. 8, pp 905-914, Aug. 1984.
- [17] S. Yamada, M. Ohba, and S. Osaki, "S-sharped software reliability growth models and their applications," *IEEE Trans. Rel.*, vol. R-33, pp. 289-292, Oct. 1984.