로지스틱 특성곡선을 이용한 발행시기 연구

최규식
건양대학교 의공학과

요약

소프트웨어 개발 후 인도 전 테스트 단계중에 발생되는 테스트 노력 소요량을 고려한 소프트웨어 신뢰도 성장 모델을 제시하여 테스트 노력소요량 동태를 시간함수인 로지스틱 곡선으로 설명한다. 테스트 단계중에 소요되는 테스트노력의 양에 대한 결함 검출비를 현재의 결함 내용에 비례하는 것으로 가정하여 소프트웨어 신뢰도 성장 모델을 비동차 포아송 프로세스(NHPP)로 공식화하여, 이 모델을 이용하여 소프트웨어 신뢰도 척도에 대한 데이터 분석기법을 개발한다. 그간 여러 문헌에서 소프트웨어 신뢰도 향상 모델을 연구할 때 소프트웨어 테스트 중에 소요되는 테스트노력의 양으로서 지수함수 곡선, 레일레이 곡선, 웨이불 곡선을 사용해 왔다. 그러나, 모든 소프트웨어 개발 환경에서 지금까지 제시된 그러한 곡선중 하나에 의해서 테스트노력 소요 곡선을 표현하는 것은 적절하지 못하다는 것이 밝혀지고 있다. 본 논문에서는 로지스틱 테스트노력 곡선이 소프트웨어의 개발/테스트 노력곡선으로 적절하게 표현될 수 있다는 것과 실제 데이터를 근거로 하여 적용하여서 예측성이 매우 좋은 능력을 가지고 있다는 것을 보이고자 한다.

1. 서 론

그동안 테스트 단계동안 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)[1]이라 한다. Musa 등[2]은 기존 소프트웨어 신뢰도 성장모델을 분류하는 알고리즘을 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.

신뢰도와 비용을 고려한 연구로서 Okumoto와 Goel[5]은 전체 평균 소프트웨어 비용을 최소화시키는 비용-최적 SRP를 발표하였다. Yamada와 Osaki[6]는 전체 평균 비용을 최소화시키고 소프트웨어 신뢰도를 만족시키는 전체평균비용-신뢰도-최적 SRP를 도입하였다. 이러한 연구결과를 참조하여 Hou, Kuo, Chang[7]은 지수 곡선과 로지스틱 곡선에 적용하는 연구를 수행하였다.

2. 테스트 노력 함수의 배경

2.1 웨이불형 테스트노력 함수

Yamada등[3]이 제안한 웨이불형 테스트 노력 함수에 의하면 소프트웨어의 테스트 노력이 테스트 단계 전체에 걸쳐서 일정하다고 가정해서는 안된다고 한다.

1) 지수함수곡선 : (0, t]에서 소모되는 누적 테스트 노력은

$$W(t)=N[1-\exp(-\beta t)] \qquad (2.1)$$

로서 웨이불함수의 m=1인 경우에 해당된다.

2) 레일레이 곡선 : 소모되는 누적 테스트 노력은

$$W(t)=N[1-\exp(-\frac{\beta}{2}\cdot t^2)] \qquad (2.2)$$

로서 웨이불함수의 m=2인 경우에 해당된다.

3)웨이불 곡선 : 소모되는 누적 테스트 노력은

$$W(t)=N[1-\exp(-\beta t^m)] \qquad (2.3)$$

로서 웨이불 함수의 일반적인 경우, 즉 m=3, 4, …인 경우에 해당된다.

2.2 로지스틱형 테스트 노력 함수

(0, t]에서의 누적 테스트 노력 소모는

$$W(t))=\frac{N}{1+A\cdot\exp(-\alpha t)} \qquad (2.4)$$

이다. 현재의 테스트 노력 소모량은 W(t)의 미분치로서

$$w(t)=\frac{dW(t)}{dt}=\frac{NA\alpha\cdot\exp(-\alpha t)}{[1+A\cdot\exp(-\alpha t)]^2}$$

$$=\frac{NA\alpha}{[\exp(\alpha\frac{t}{2})+A\cdot\exp(-\frac{\alpha t}{2})]^2} \qquad (2.5)$$

와 같이 표현된다.

3. 로지스틱 테스트노력 SRGM

3.1 소프트웨어 신뢰도 척도

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

$$R(x \mid t) = \exp[-m(t+x)+m(t)] \qquad (3.1)$$

3.2 로지스틱형 성장모델

통계적인 예측에 의하여 현재의 결함 내용은 어느 경우이든 유한하므로 m(t)는 t에 대한 증가함수이며 m(0)=0이다. 이러한 가정 하에

$$r = \lim_{\Delta t \to 0}\left(\frac{m(t+\Delta t)-m(t)}{[a-m(t)]\,w(t)\Delta t}\right) \qquad (3.2)$$

이며, 현재의 테스트 노력 비용에 의해서 검출되는 결함의 수가 잔여결함의 수에 비례하면

$$\frac{dm(t)}{dt}\cdot\frac{1}{w(t)} = r[a-m(t)] \qquad (3.3)$$

과 같이 방정식을 세울 수 있다.

경계조건 m(0)=0인 조건에서 (3.3)을 풀면

$$m(t) = a(1-e^{-r[W(t)-W(0)]})$$
$$= a\{1-\exp[-r\cdot W^*(t)\}$$
$$W^*(t) = \frac{N}{1+A\cdot\exp(-\alpha t)} - \frac{N}{1+A} \qquad (3.4)$$

이고, 고장강도는

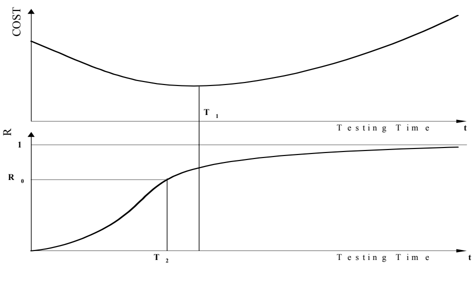
$$\lambda(t) = \frac{dm(t)}{dt} = arw(t)\cdot\exp[-rW^*(t)]$$

(3.5)이다.

4. 비용-신뢰도를 고려한 인도 기법

<그림 4.1>에서는 테스트 기간을 횡축으로 하여 비용과 신뢰도의 관계를 동시에 나타낸 것이다. 최적 소프트웨어 인도문제를 다음과 같이 정의한다.

$c_2>0,\ c_1>c_3>0,\ x\geq 0,\ 0<R_0<1$인 경우에 대해서 $R(x|T)\geq R_0,\ T\geq 0$인 조건하에 $C(T)$를 최소화 (4.1)



<그림 4.1> 비용-신뢰도 관계 곡선

이러한 방법으로 하여 비용-신뢰도최적 소트웨어 인도시각에 대한 해를 구할 수 있다.

$$T^* = \max\{T_1,\ T_2\} \qquad (4.2)$$

4.1 비용 기준에 의한 인도 시각

소프트웨어 테스트/개발 단계 기간 동안의 테스트 노력 비용과, 인도 전후의 오류 수정 비용은 다음과 같다.

$$C(T) = C_1 m(T) + C_2[m(T_{LC})-m(T)]$$
$$+ C_3\int_0^T w(x)dx \qquad (4.3)$$

비용이 최소가 되는 시각을 구하기 위해

$$\exp\left[-\frac{Ar(e^{\alpha T_{LC}}-1)}{(1+A)(A+e^{\alpha T_{LC}})}\right] \leq \frac{C_3}{ar(C_2-C_1)} \leq 1$$

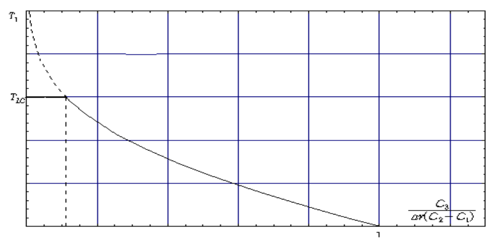
에서 유일하고도 유한한 양의 해 $T*=T_1$이 존재하고 그 값은 (4.3)과 같다.

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

$\frac{C_3}{ar(C_2-C_1)} < \exp\left[-\frac{Ar(e^{\alpha T_{LC}}-1)}{(1+A)(A+e^{\alpha T_{LC}})}\right]$

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

이와 같은 내용을 <그림 4.2>에 표시하였다.



<그림 4.2> 조건에 따른 인도시각

$\exp\left[-\frac{Ar(e^{\alpha T_{LC}}-1)}{(1+A)(A+e^{\alpha T_{LC}})}\right] \leq \frac{C_3}{ar(C_2-C_1)} \leq 1$ 이면 비용최저점이 결함시험과 소프트웨어의 전 수명기간 사이에 존재하는 경우이다.

$\frac{C_3}{ar(C_2-C_1)} < \exp\left[-\frac{Ar(e^{\alpha T_{LC}}-1)}{(1+A)(A+e^{\alpha T_{LC}})}\right]$ 인 조건은 비용이 단조감소하는 경우이다.

주어진 조건에서 MLE와 LSE를 이용하여 관련 파라미터를 구하고 이 파라미터를 이용하여 방정식(4.)으로부터 비용 최적 인도시각 T_1을 결정한다.

4.2 신뢰도 기준에 의한 인도 시각

위의 방정식으로부터

$$\log R(x|0) = -m(x) = -a(e^{-rW^*(0)} - e^{-rW^*(x)})$$

에서 유일하고도 유한한 양의 해 $T^*=T_1$이 존재하고 그 값은 (4.3)과 같다.

$$R(x|0)=\exp\left[-a\left(1-e^{-r\left(\frac{N}{1+Ae^{-\alpha x}}-\frac{N}{1+A}\right)}\right)\right] \quad (4.4)$$

과, 또 이 결과로부터

$$e^{-\alpha x}=\frac{1}{A}\left\{\frac{1+A}{1-\frac{1+A}{rN}\log[1+\frac{1}{a}\log R(x|0)]}-1\right\} \quad (4.5)$$

을 얻는다.

$$S(N,\ A,\ \alpha)=\sum_{k=1}^{n}[W_k-W(t_k)] \quad (4.6)$$

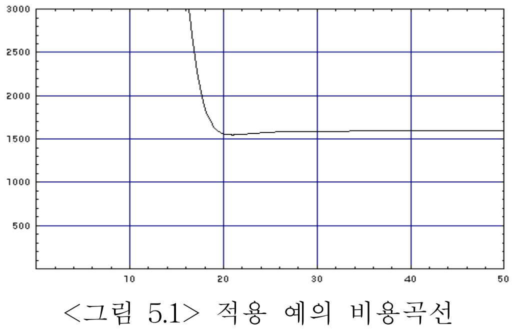
주어진 조건에서 MLE와 LSE를 이용하여 관련 파라미터를 구하고 이 파라미터를 이용하여 방정식 (4.6)으로부터 비용 최적 인도시각 T_2를 결정한다.

5. 적용 예

N=29.1095, A=4624.89, α=0.493515, a=138.165, r=0.145098, C_1=1, C_2=100, C_3=50, T_{LC}=100, R_0=0.95, x=1이다.

5.1 비용

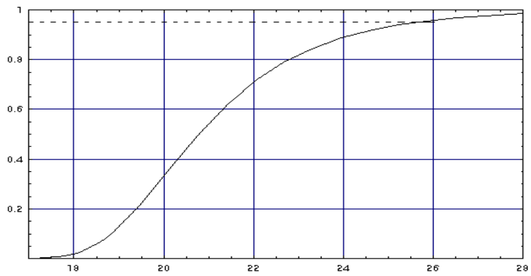
조건1)에 해당하므로 방정식(4.3)을 이용하여 그 시각을 구하면 T_1=20.04가 되고 방정식(4.3)을 이용하여 최저비용을 구하면 C=1,548.8이 된다. 이는 <그림 5.1>에서도 명백하게 드러난다. 이 때의 신뢰도를 구하면 0.5453으로서 고객이 요구하는 목표 신뢰도와 거리가 멀다. 즉, 비용을 최저로 하는 시점에서 소프트웨어의 테스트를 중지하여 고객에게 인도하게 되면 소프트웨어의 신뢰도가 54.53%밖에 되지 않아서 목표신뢰도 95%에 이르기 위해서는 좀더 경과시간을 두고 테스트를 계속하여 신뢰도를 향상시켜야 한다.



<그림 5.1> 적용 예의 비용곡선

5.2 신뢰도

목표신뢰도 95%를 만족하는 시각을 구하면 T_2=26이 된다. 이는 소프트웨어의 신뢰도를 목표치까지 상승시키기 위해서 그 시간이 26으로 될 때까지 소프트웨어의 테스트를 계속하여 신뢰도를 향상시켜야 함을 의미한다. 따라서, (4.2)의 조건에 따라 $T^*=T_2$=26이 되어야 하며, 이 때의 총 비용은 1,584.4로서 최저비용보다 35.6만큼 증가된다.



<그림 5.2> 적용 예의 신뢰도 곡선

이 예에서는 T_{LC}=100이라는 수명 기간 동안 경과시간 x=1일 때의 목표 신뢰도 R_0=0.95를 유지하기 위한 시간을 구함에 있어서 비용을 최저로 하는 시간을 구하고 목표 신뢰도를 만족하는 시간을 구해봤을 때 $T_1<T_2$인 경우로서 인도시간이 비용최저에 의해서 결정된 것이 아니고 목표 신뢰도를 맞추는 시점에서 결정되었다.

6. 결 론

비용면에서는 개발, 테스트, 인도, 인도 후 A/S에 대한 총 비용을 구하는 방법을 연구하고 이를 최소 비용으로 줄일 수 있는 기법을 제시하였다. 이는 조건에 따라서 그 시기가 달라진다. 즉, $\frac{C_3}{ar(C_2-C_1)}>1$ 이면 비용이 단조증가하는 경우로서 결함시험을 하면 할수록 비용이 증가되어 시험 없이 인도하는 것이 최적인 것을 의미한다.

$\exp\left[-\frac{Ar(e^{\alpha T_{LC}}-1)}{(1+A)(A+e^{\alpha T_{LC}})}\right]\le\frac{C_3}{ar(C_2-C_1)}\le 1$ 이면 비용 최저점이 결함시험과 소프트웨어의 전 수명기간 사이에 존재하는 경우이다.

$\frac{C_3}{ar(C_2-C_1)}<\exp\left[-\frac{Ar(e^{\alpha T_{LC}}-1)}{(1+A)(A+e^{\alpha T_{LC}})}\right]$ 인 조건은 비용이 단조감소하는 경우이다.

또한, 신뢰도 기준에 의한 인도수법을 고찰한다. 일반적으로 소프트웨어 인도시간문제는 소프트웨어

시스템의 신뢰도와 연동되어 있다. 소프트웨어 시스템의 신뢰도가 적정한 수준에 이르면 소프트웨어를 인도한다. 마지막 고장이 발생한 후 조건 확률함수를 이용하여 목표신뢰도를 만족시키는 시간을 결정하는 방정식을 제시하였다.

참고문헌

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

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

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

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

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

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