2005년도 대한전기학회 하계학술대회 논문집 2005. 7. 18 - 20

소프트웨어 신뢰도의 매개변수 도출

최규식
건양대학교 의공학과

A Study on the Edequate Parameter Estimation of S W

Che Gyu Shik
Konyang University

Abstract - 몇몇 논문을 통하여 테스트 노력을 고려한 소프트웨어의 신뢰도 평가가 중요한 인자인 것으로 발표되었다. 여러 산업 현장의 경험 데이타에 의하면 그 형태가 지수함수형, 레일레이형, 웨이불형, 로지스틱형 테스트 노력 함수 중 하나인 것으로 보고되었다. 그러므로 연구자들은 각각의 소프트웨어 테스트 형태에 따라 이중 하나의 적합한 형태를 사용해왔다. 테스트 노력이 어떤 형태의 그래프를 따르냐에 의해 신뢰도 예측 및 성장에 대한 평가가 달라지고 그에 따라 인도시기도 변한다.

따라서, 본 논문에서는 이 네 가지 형태의 테스트 노력을 가진 소프트웨어의 신뢰도 성장에 필요한 각종 파라미터를 구하는 방법에 대하여 고찰한다. 개발 현장에서 관찰된 테스트 노력 데이터와 결함검출을 비교하여 어느 형태의 테스트 노력 곡선이 그 경우에 적합한가를 연구하는 한편, 목표 신뢰도에 맞는 발행시기를 결정하는 문제를 연구한다.

1. 서 론

소프트웨어 신뢰도는 규정된 시간 동안 주어진 환경에서 소프트웨어가 고장 없이 원활하게 동작할 확률로서 소프트웨어의 품질을 평가할 때의 핵심 사항이다. 소프트웨어 신뢰도는 고객의 입장에서 본 소프트웨어 품질 관점이다. 이는 단순히 프로그램 설계보다는 실제적인 운전과 관련이 있으므로, 정적이라기 보다는 동적이다. 그러므로 소프트웨어 시스템의 신뢰도를 측정하고 계산 및 성장시키는 것이 매우 중요하다. 이는 개발 기간동안 모든 테스트 자원을 계획하고 제어하는데 쓰일 수 있어서, 우리에게 소프트웨어의 정확성을 정량적으로 확인시켜 주는 것이다. 소프트웨어 신뢰도를 측정하기 위한 공통적인 접근법은 소프트웨어 고장으로부터 구한 가용한 데이터로부터 산출된 파라미터를 가진 분석모델을 이용하는 것이다.

한편, 소프트웨어 개발에 있어서 상당한 양의 테스트 자원이 소프트웨어의 테스트에 쓰인다. 이러한 테스트 기간에 대한 테스트 자원의 소요곡선을 테스트 노력 곡선으로 표현할 수 있다. 테스트 노력은 테스트 기간중의 실행 테스트 케이스의 수, 인력의 양, CPU시간 등이다. 소프트웨어 테스트 기간 동안 소프트웨어의 신뢰도는 잠재 소프트웨어 결함을 검출하고 수정하는 데에 소요되는 개발자원의 양에 전적으로 의존한다. 그러나, 많은 연구가들의 연구에서는 테스트 단계의 테스트 자원의 소모율은 일정한 것으로 가정하거나 또는 그러한 테스트 노력을 고려하지도 않았었다.

2항에서는 소프트웨어의 신뢰도에 대한 일반적인 정의 및 의미를 기술하며, 3항에서는 각종 테스트 노력 함수에 대해서 고찰한다. 4항에서는 이와 같은 테스트 노력 함수를 고려한 신뢰도를 구함에 있어서 파라미터를 구하는 방법에 대해서 연구하고, 실례를 들어서 파라미터를 구하고 테스트 노력곡선과 신뢰도를 산출한다. 5항에서는 결론으로서 본 연구의 결과를 고찰하고 앞으로 연구해야 할 분야를 기술하였다.

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.

$$R(x|s)=\Pr\{X_k>x|S_{k-1}=s\} \tag{2.1}$$

이는 소프트웨어를 개발하여 결함검출 테스트를 시작한 후 계속 s 유니트 시간 동안 $(k-1)$번 째 결함을 발견하여 수정한 후 k번 째 결함이 발견되기 전까지 운영될 x 시간동안에 고장 없이 소프트웨어가 동작할 확률인 것이다.

식(2.1)의 시간간격 X_k가 소프트웨어의 테스트로서 테스트중이고 테스트노력이 일정하다고 가정하고 테스트 공정이 NHPP를 따른다면 NHPP의 표준 이론으로부터 평균치 함수를

$$m(t)=a(1-e^{-bt}) \tag{2.2}$$

로 정의할 때[6] 임의의 $t\geq 0$과 $x>0$에서

$$\Pr\{N(t+x)-N(t)=k\}=\frac{[m(t+x)-m(t)]^k}{k!}\exp\{-[m(t+x)-m(t)]\} \tag{2.3}$$

이므로, 테스트 노력이 일정한 경우 식(2.1)의 신뢰도는 다음과 같이 표현할 수 있다.

$$R(x|t)\equiv\Pr\{N(t+x)-N(t)=0\}=\exp[-m(x)e^{-bt}] \tag{2.4}$$

3. 테스트 노력 함수

3.1 평균치 함수 및 신뢰도

초기치를 고려한 누적 테스트 노력 함수를 $W^*(t)$로 가정하여 소프트웨어를 시각 t에서 발행하는 경우, 발견되는 누적 결함 분포는 평균치 함수

$$m(t)=a(1-e^{-\gamma W^*(t)}) \tag{3.1}$$

이므로,

$$R(x|t)=\exp[-m(t+x)+m(t)]$$
$$=\exp[-a\cdot e^{-\gamma W^*(t)}(1-e^{-\gamma W^*(x)})] \tag{3.2}$$

이 된다.

따라서, 시각 $t=T$에서 발행된 소프트웨어의 신뢰도는 경과 시간 x에 대해서 지수 함수적으로 감소한다.

목표 신뢰도를 $R(x|T)=R_0$라 하면

$$W^*(t)=\frac{1}{\gamma}\log\frac{1-e^{-\gamma W^*(x)}}{\frac{1}{a}\log\frac{1}{R_0}} \tag{3.3}$$

에서 최적 인도 시각 $T^*=T$를 구한다.

3.2 웨이불형 함수

야마다가 제안한 웨이불형 테스트 노력 함수에 의하면 소프트웨어의 테스트 노력이 테스트 단계 전체에 걸쳐서 일정하다고 가정하면 비현실적이다. 그리고, 순간적

인 테스트 노력이 결국은 테스트 수명주기 동안 감소한다. 그 이유는 누적 테스트 노력이 유한치에 접근하기 때문이다. 그 어떤 소프트웨어 개발회사도 소프트웨어 개발에 무한정으로 자원을 투입하지 않기 때문에 이러한 가정이 합리적이라 할 수 있다. 여러 관련 문헌에 의하면 테스트 노력이 웨이불형 분포로 설명될 수 있고, 아래와 같은 3개의 케이스를 가진다는 것을 보여주고 있다.

1) 지수함수형 곡선 : (0, t]에서 소요되는 누적 테스트 노력은

$$W(t)=N[1-e^{-\beta t}] \tag{3.4}$$

로서 웨이불 함수의 m=1인 경우에 해당되며, 신뢰도는

$$R(x|t)=\exp[-a\cdot e^{-rN(1-e^{-\beta t})}\cdot(1-e^{-rN(1-e^{-\beta t})})] \tag{3.5}$$

로 표현된다.

2) 레일레이형 곡선 : 소요되는 누적 테스트 노력은

$$W(t)=N[1-\exp^{-\frac{\beta}{2}\cdot t^2})] \tag{3.6}$$

로서 웨이불함수의 m=2인 경우에 해당된다.
신뢰도는

$$R(x|t)=\exp[-a\cdot e^{-rN(1-e^{-\frac{\beta}{2}t^2})}\cdot(1-e^{-rN(1-e^{-\frac{\beta}{2}t^2})})] \tag{3.7}$$

로 표현된다.

3) 웨이불형 곡선 : 소요되는 누적 테스트 노력은

$$W(t)=N[1-e^{-\beta t^m})] \tag{3.8}$$

로서 웨이불 함수의 일반적인 경우, 즉 m>2인 경우에 해당된다.
신뢰도는

$$R(x|t)=\exp[-a\cdot e^{-rN(1-e^{-\beta t^m})}\cdot(1-e^{-rN(1-e^{\beta t^m})})] \tag{3.9}$$

로 표현된다.

3.3 로지스틱형 함수

실제 테스트 노력 데이타가 여러 가지 소요 패턴을 나타내므로 때때로 테스트 노력 비용을 지수함수나 레일레이 곡선만으로 설명하기는 어렵다. 웨이불형 곡선은 일반적인 소프트웨어 개발 환경 하에서 데이터에 잘 맞고, 소프트웨어 신뢰도 모델링에 널리 쓰이지만 m>3일 때 피크현상을 가진다. 그 대안으로 제시된 것이 로지스틱형 테스트노력 함수이다.

$$W(t)=\frac{N}{1+A\cdot e^{-\alpha t}} \tag{3.10}$$

이고, 신뢰도는

$$R(x|t)=\exp[-a\cdot e^{-rN(\frac{1}{1+Ae^{-\alpha t}}-\frac{1}{1+A})}] \tag{3.11}$$

로 표현된다.

3.4 발행시각 결정

SRGM을 분석하여 목표 신뢰도에 가장 근접하는 유일한 시각을 구할 수 있다. 최적 소프트웨어 발행 시각은 미리 규정된 소프트웨어 목표 신뢰도에 가장 근접하는 시각이다. 목표신뢰도를 만족시키는 발행시각을 구하면 다음과 같다.

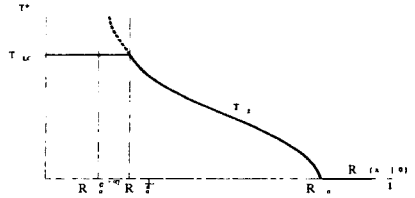
목표신뢰도를 R_o라 하면

$$\exp\{-ae^{-\alpha\gamma(1-e^{\beta T^m})}[1-e^{-\alpha\gamma(1-e^{\beta T^m})}]\}=R_o \tag{3.12}$$

이다. 이를 만족하는 발행시각을 T^*라 하면

$$T^*=\left\{-\frac{1}{\beta}\ln\left[1+\frac{1}{\alpha\gamma}\ln\frac{\ln R_o}{\ln R(x|0)}\right]\right\}^{\frac{1}{m}} \tag{3.13}$$

로 표현된다. $R(x|0)$와 목표신뢰도에 이르는 시각 T^*와의 관계는 그림 3.1과 같다.



(그림 3.1) 목표신뢰도와 발행시각 곡선

4. 파라미터 적용

4.1 파라미터 산출법

위의 각 경우에서 정의된 테스트 노력 함수에서 파라미터 N, A, a, β는 최소자승법(LSE)으로 산출한다. 최대가능성 평가자(MLE)는 한 집합의 동시방정식을 풀어서 파라미터를 산출하며, s-신뢰 구간을 구동하는데 더 좋은 방법이다. LSE는 실제로 관찰/획득한 것과 예측한 것 사이의 차이를 제곱해서 더한 총값을 최소화하는 것이다. LSE는 중간 크기의 표본에 최적인 것으로 알려지고 있으며, 최적점 산출을 제공한다. 최소자승법을 적용하기 위한 산출 공식$S(N, A, a)$은 다음과 같다.

$$S(N,\ A,\ a)=\sum_{k=1}^{n}[W_k-W(t_k)]^2 \tag{4.1}$$

W_k = (0, t]기간동안 실제로 소요되는 누적 테스트 노력
$W(t_k)$ = 테스트 노력 함수에 의해서 산출된 누적 테스트 노력
S를 N, A, a에 관하여 미분하여 그 편미분치를 0으로 놓으면, 그리고 이 항들을 재정비하면 이러한 형태의 비선형 최소 자승 문제를 푼다.

또한, 상기에서 정의한 테스트 노력 함수의 산출 파라미터를 이용하여 MLE에 의해서 $m(t)$의 신뢰도 성장 파라미터 a와 r을 구할 수 있다. 비동차포아송과정(NHPP)의 표준이론으로부터 임의의 $t\geq0$, $x>0$에서

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

$$=\frac{\{m(t+x)-m(t)\}^k}{k!}\cdot e^{-m(t+x)+m(t)} \tag{4.2}$$

이므로, 여기에 $m(t+x)=m(t_k)$, $m(t)=m(t_{k-1})$, $k=m_k-m_{k-1}$를 대입하면

$$\Pr\{N(t_k)-N(t_{k-1})=m_k-m_{k-1}\}$$

$$=\frac{\{m(t_k)-m(t_{k-1})\}^{m_k-m_{k-1}}}{(m_k-m_{k-1})!}\cdot e^{-m(t_k)+m(t_{k-1})} \tag{4.3}$$

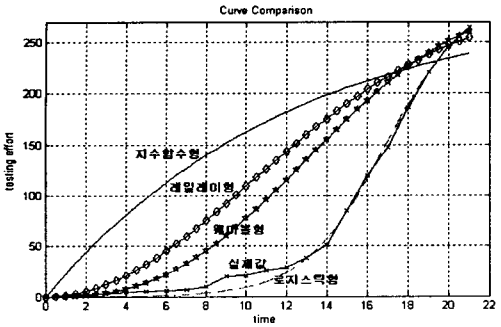
4.2 적용 예

3항에서 연구한 NHPP 모델의 수치 적용 예를 보이기 위해서 실제적인 소프트웨어 에레 데이타에서 조사된 데이터집합을 가지고 다음과 같은 값을 산출하여 적용하였다.

이 데이터를 이용하여 각 경우의 파라미터에 의한 테스트 노력곡선이 실제의 현상과 어떤 연관관계를 가지는가를 검토하기 위하여 추이곡선을 그렸으며, 그 결과는 그림 4.1과 같다.

<표 4.1> 실제 데이터

t_k	1	2	3	4	5	6	7	8	9	10	–
m_k	8	16	25	37	41	46	52	59	64	80	–
w_k	1	1	1	1	1	1	1	3	10	1	–
t_k	11	12	13	14	15	16	17	18	19	20	21
m_k	89	98	107	121	128	131	133	133	135	136	137
w_k	5	3	9	13	34	35	27	39	35	25	19



(그림 4.1) 테스트 노력 곡선 비교

4가지 경우의 테스트 노력 곡선에 대한 계수를 구하여 실제 관측된 데이터 곡선과 비교하였을 때, 그림에서 보는 바와 같이 테스트 중간에서 지수함수형과 레일레이형, 웨이불형 모두가 실제와 큰 차이를 나타내고 있다. 이와 대조적으로 로지스틱형은 테스트 초창기만 약간 차이를 보이고 있으나, 테스트 시간이 경과함에 따라 실제 현상과 거의 근접한 것을 알 수 있다. 초창기에 차이가 나는 것은 테스트 초기에는 테스트 여건이나 테스트자의 숙련도 등이 미숙하거나 불확실한 이유 때문으로 사료된다.

이러한 각종 파라미터에 의한 테스트노력곡선을 이용하여 각 경우의 신뢰도 성장 곡선을 그리면 그림 4.2와 같다. 단, 여기에서 목표신뢰도 R_0=0.95, 경과시간 x=1로 가정하였다.

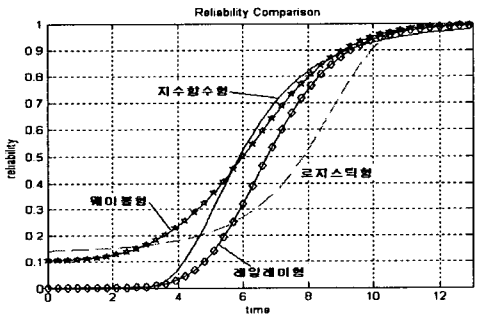


그림 4.2 SRGM 비교

그림을 검토해볼 때 지수함수형인 경우, 초창기에 신뢰도 성장이 저조하다가 T=3 부근에서 급격히 성장되어 T=10 부근에서 안정된다. 레일레이형인 경우, 지수함수형인 경우와 마찬가지로 초창기에 성장이 극히 저조하다가 T=4 부근에서 서서히 성장되어 T=10 근처에서 안정된다. 웨이불형인 경우는 테스트 초기부터 10% 정도의 신뢰도를 가지며, 테스트 시간 경과에 따라 원활하고도 완만한 성장을 한다. 이의 특징은 앞의 두 가지경우에 비하여 처음 신뢰도가 10% 정도 된다는 것이다. 로지스틱형인 경우는 테스트 초창기에 14% 정도의 신뢰도를 가지고 있으며, 타 형태의 테스트 노력에 비하여 신뢰도 성장이 극히 저조하다가 T=5 부근에서부터 서서히 성장하기 시작하여 T=7 부근에서 갑자기 성장, T=10 부근에서 안정되는 지연 S 형태를 취한다. 한편, 목표신뢰도 R_0=0.95를 만족시키는 인도시각은 지수함수형인 경우, T=10.8, 레일레이함수형인 경우, T=10.3, 웨이불형인 경우, T=9.9, 로지스틱형인 경우, T=10.4로서 모두가 대동소이하다.

5. 결 론

본 논문에서는 상기 네 가지 경우에 대한 함수의 파라미터 산출법을 연구하고, 이러한 파라미터 산출법과 함수들이 실제와 얼마나 부합되는가를 검토하기 위하여, 실제 개발 과정을 통해 수집된 경험 데이터를 가지고 각 경우에 대한 파라미터를 산출하고 이를 비교하였다.

웨이불형은 앞의 두 가지 형에 대한 보완형으로 제안되었다고 할 만큼 융통성이 있어서 실제 현상과 유사한 특성을 나타내지만, 시간 지수가 3 이상이면 문제가 있는 것으로 검토되었으므로 이를 유의할 필요가 있다. 최근에는 로지스틱형이 제안되고 있는데, 본 연구에서 비교 연구한 바에 의하면 사례의 현상에 부합하는 것으로 판명되었다.

그리고, 목표신뢰도를 정하고 산출된 파라미터에 의해서 각 경우의 발행시각을 계산해본 결과, 어느 정도 합리적인 결과를 얻었으나, 테스트 초창기에는 불규칙하고 불확실한 테스트 노력 현상 때문에 산출된 파라미터의 신뢰성이 저하된다는 것을 발견하였다.

감사의 글

이 논문은 2005학년도 건양대학교 학술연구비 지원에 의하여 이루어진 것임

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

[3] S. Yamada, J. Hishitani, S. Osaki, "Software - Reliability Growth with a Weibull Test-Effort : A Model & Application", IEEE Trans. Reliability, vol. 42, no.1, pp100-106, 1993 March

[4] Syed A. Hossain, Ram C. Dahiya, " Estimating the Parameters of a Non-homogeneous Poisson-Process Model for Software Reliability", IEEE Trans. Reliability, vol. 42, pp604-612, no.4, 1993 Dec.

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

[6] 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

[7] Chin-Yu Huang, Sy-Yen Kuo, "Analysis of Incorporating Logistic Testing-Effort Function into Software Reliability Modeling", IEEE Trans. on Reliability, vol.51, pp261-270, 2002, Sep.