

S/W의 테스트 신뢰도와 운영 신뢰도에 대한 연구

최규식, 김종기

che@kytis.konyang.ac.kr, jkkim@kytis.konyang.ac.kr

요약

S/W 신뢰도는 보통 NHPP와 같은 신뢰도모델에 의해서 평가한다. S/W 시스템은 테스트 단계에서 품질이 향상되고 운영단계에서는 거의 변화가 없다. 신뢰도를 테스트 단계에서 추정할 것인가 아니면 운영단계에서 추정할 것인가의 여부에 따라서 그 추정방법이 달라져야 한다. 본 논문에서는 두 가지의 상이한 신뢰도개념 즉 운영신뢰도와 테스트 신뢰도를 명확히 구분하여 자세히 고찰한다.

1. 서론

과거 20여년 동안 많은 SRGM이 제안되었다. 많은 연구가와 참여자들에 의해서 보편적으로 널리 연구되고 사용되는 SRGM의 부류는 NHPP 모델이며, 이러한 부류의 모델은 실제로 많은 장점을 가지고 있어서 그간 많은 관심을 끌어왔다.

S/W 신뢰도는 관련 SRGM으로부터 유도되어 왔다. 여기에 두 가지의 상이한 신뢰도 개념이 있는데 즉, 테스트 신뢰도와 운영 신뢰도가 바로 그것이다. 테스트 신뢰도는 테스트 단계에서 에러가 발생하지 않을 확률이며, 운영 신뢰도는 운영 단계에서 에러가 발생하지 않을 확률이다. 테스트 단계에서는 소프트웨어결함이 발견되는 대로 제거되어 유사한 에러가 다시 나타나지 않기 때문에 고장발생율은 테스트 시간 여부에 달려있다. 반면에, 운영테스트 기간 중에는 결함을 제거하는 것이 테스트 단계와 같을 수가 없어서, 사용자가 사용하는 전 기간에 걸쳐서 일정한 고장발생율을 겪게 된다. 이러한 두 가지 신뢰도개념은 명백히 다르며, 이들을 구분 짓는데 중요한 사항이 되어 상황에 따라 적절히 사용해야 한다.

명명법

- $m(t)$ 시간간격 $[0, t]$ 에서 관찰되는 예상 결함의 수
- T_0 $m(t)$ 의 변곡점 즉, $[d^2 m(t)/dt^2]_{t=T_0}=0$ 을 만족하는 점
- $\lambda(t)$ 시각 t 에서의 순간 고장강도, $\lambda(t)=dm(t)/dt$
- λ_r 소프트웨어 발행시점의 순간고장강도
- R_0 목표 S/W 신뢰도레벨
- a 테스트를 시작할 때의 소프트웨어결함의 수
- b 소프트웨어에서 잔여결함의 검출비
- c_1 테스트기간중의 예상 결함제거 비용
- c_2 운영기간중의 예상 결함제거 비용

c_s 단위 테스트시간당 예상 비용
 T 소프트웨어 발행시각
 $C(T)$ 소프트웨어개발의 총 예상비용

2. 테스트 및 운영신뢰도 개념

S/W 신뢰도는 규정된 환경 하에서 주어진 기간에 대한 무결함 소프트웨어운영확률인 것으로 정의한다. 대부분의 경우에 있어서 소프트웨어의 결함이력이 알려져 있으므로 S/W 신뢰도는 다음과 같이 조건확률로 표현할 수 있다.

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

이는 x 유닛의 s 유닛 기간 동안 주어진 고장이력에서 다음 고장간격 기간 동안의 신뢰도를 표현한다. 그림 1 참조. 시각 t 까지의 누적고장수를 $N(t)$ 로 표현하고 $\{N(t); t \geq 0\}$ 평균치함수 $m(t)$ 를 가진 NHPP로 모델화할 수 있다. 본 항에서는 두 개의 신뢰도 개념을 정의하고 이들의 차이를 추정신뢰도로 고찰하고자 한다.

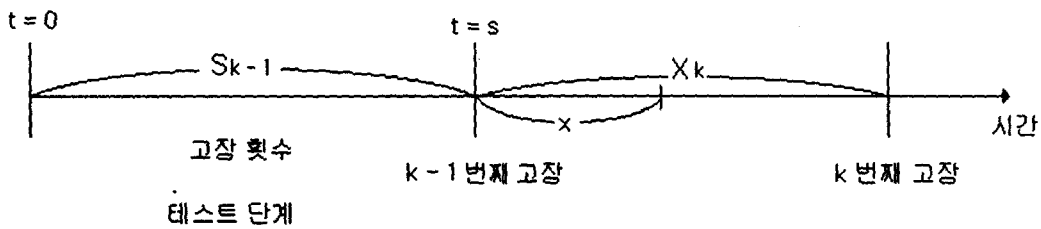


그림 1. 고장발생빈도 표현

2.1 테스트 신뢰도

테스트 단계 동안에는 소프트웨어가 개선된다. 식 (1)의 시간간격 X_k 가 소프트웨어의 테스트 단계이면 즉, 소프트웨어가 아직도 테스트중이고 테스트공정이 NHPP를 따른다면 NHPP의 표준이론으로부터 평균치 함수를 $m(t) = a(1 - e^{-bt})$ 로 정의할 때 임의의 $t \geq 0$ 과 $x > 0$ 에서

$$\Pr\{N(t+x) - N(t) = k\} = \frac{[m(t+x) - m(t)]^k}{k!} \exp\{-[m(t+x) - m(t)]\} \quad (2)$$

이므로, 식(1)의 신뢰도는 다음과 같이 계산할 수 있다.

$$R_{te}(x|s) \equiv \Pr\{N(s+x) - N(s) = 0\} \\ = \exp\{-[m(s+x) - m(s)]\} \quad (3)$$

$R_{te}(x|s)$ 가 테스트공정기간의 소프트웨어 신뢰도만을 측정할 수 있으며, 운영단계까지 연장되지 않으므로 식(3)에서 정의된 소프트웨어 신뢰도를 테스트신뢰도로 부를 수 있다.

2.2 운영신뢰도

소프트웨어를 s 단위의 시간동안 계속 테스트하여 고객에게 발행되었다면, 즉, 시간간격 X_k 가 운영단계에 있다면 다음 고장시각은 파라미터 λ_r 을 가진 지수분포를 따르며, 여기서 $\lambda_r = \lambda(s)$ 는 시각 s 에서 계산된 본래 NHPP의 고장강도함수이다. 신뢰도함수는 아래와 같다.

$$R_{op}(x|s) = \exp\left(-\int_0^x \lambda_r dt\right) = \exp[-\lambda(s)x] \quad (4)$$

식(4)에서 정의된 소프트웨어 신뢰도를 운영신뢰도라 부르며, 이는 운영단계기간의 소프트웨어 신뢰도를 측정한다. 즉, x 는 운영단계에서의 경과시간을 나타낸다. 식(3)과 (4)는 동일하지 않으며, 어느 것을 사용하느냐의 여부에 따라서 상이한 추정을 하게 된다는 사실에 유의해야 한다.

2.3 신뢰도 개념의 그래프적 표현

식(3)을 다음과 같이 다시 쓸 수 있다.

$$R_{te}(x|s) = \exp\{-[m(s+x) - m(s)]\} \\ = \exp\left[-\int_s^{s+x} \lambda(t) dt\right] = \exp(-S_{ABCD}) \quad (5)$$

여기서, S_{ABCD} 는 그림 2의 곡선 사다리꼴 ABCD의 면적이다. 이와 마찬가지로 식(4)도 다음과 같이 나타낼 수 있다.

$$R_{op}(x|s) = \exp[-\lambda(s)x] = \exp(-S_{ABCD}) \quad (6)$$

여기서, S_{ABCD} 는 그림 2의 사각형 ABC'D의 넓이를 나타낸다.

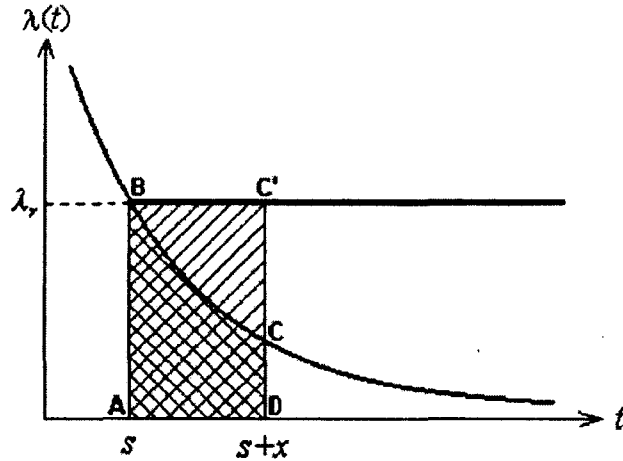


그림 2. 고장강도가 단조감소하는 경우

식(5)와 (6)으로부터 두 신뢰도 개념간의 차이를 명확하게 알 수 있다. 사실상, 두 신뢰도 정의를 아래와 같이 표현할 수 있다.

$$R(x|s) = \exp(-S) \quad (7)$$

여기서, S는 시간간격 x와 고장강도곡선으로 둘러싸인 면적을 표시한다. 테스트 신뢰도인 경우, 소프트웨어를 테스트함에 따라, 고장강도 $\lambda(t)$ 가 시간에 따라 변하기 때문에 둘러싸인 면적은 곡선·사다리꼴 ABCD이다. 그러나, 운영신뢰도인 경우, 소프트웨어가 시각 s에서 고객에게 발행되었으므로 고장강도는 운영기간동안 변하지 않고 그것이 발행될 시점의 값과 동일한 값 λ_r 를 유지한다. 그러므로, 둘러싸인 넓이는 직사각형 ABC'D이다.

3. 두 신뢰도의 비교

운영신뢰도개념과 테스트신뢰도개념이 다르므로 실제적인 차이를 연구하여 그 특성을 비교하는 것이 중요하다. 테스트단계 기간동안에는 평균치함수 $m(t)$ 가 지수함수형, 즉 $\lambda(t)$ 가 단조감소형이거나 S-형, 즉 $\lambda(t)$ 가 처음에는 증가하다가 그 다음에 감소하는 형이다. 아래의 정리에서 두 가지 경우의 결과를 요약한다.

정리 1.1 $T \geq 0$ 와 $x > 0$ 에서 주어진 어떠한 값에 대해서도 $\lambda(t)$ 가 $t \geq 0$ 에서 절대적으로 감소하는 함수이면, $R_{op}(x|T) < R_{te}(x|T)$ 이다.

이 결과는 테스트신뢰도와 운영신뢰도의 그래프적 설명으로서도 볼 수 있다.(그림 2) 사실상, $S_{ABCD} > S_{ABC'D}$ 이므로, 식(5)와 (6)으로부터 $R_{op}(x|T) < R_{te}(x|T)$ 임이 명백하다.

개발 초기에는 소프트웨어를 배우는 단계이고 또한 그래서 불안정하므로, 테스트 단계에서는 그 결함들이 감소하기 전에 결함발생빈도가 증가되는 것으로 관찰되었다. 정리 1.2에서

이러한 상황 결과를 정리하였다.

정리 1.2 $\lambda(t)$ 가 처음에 증가하다가 그 다음에 감소한다면, 그리고 $m(t)$ 의 변곡점을 T_0 로 표시하기로 하면(그림 3 참조),

케이스 1 : $T_0 \leq T$ 이면 $R_{op}(xT) < R_{te}(xT)$ 이다.

케이스 2 : $T_0 > T$ 이면 T_1 을 $\lambda(T) = \lambda(T_1)$, $T_1 > T$ 의 해라고 할 때 다음과 같다.

(i) $T_1 \geq T+x$ 이면 $R_{op}(xT) > R_{te}(xT)$ 이다.

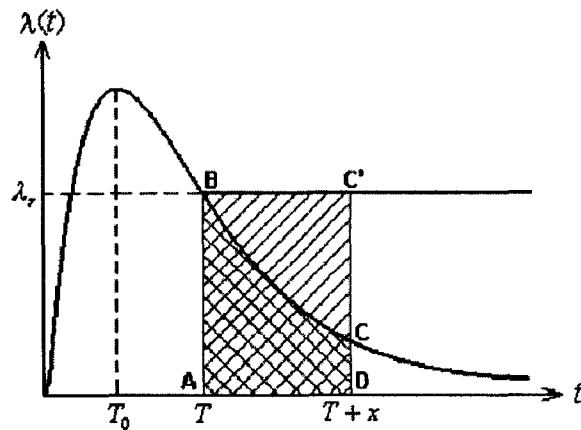
(ii) $T_1 < T+x$ 이면

$M < 0$ 일 때 $R_{op}(xT) > R_{te}(xT)$

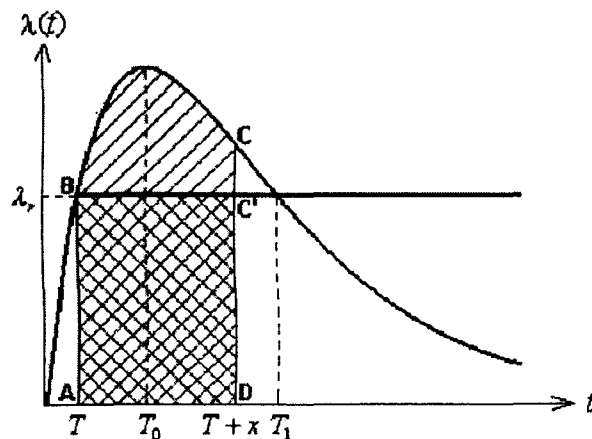
$M = 0$ 일 때 $R_{op}(xT) = R_{te}(xT)$

$M > 0$ 일 때 $R_{op}(xT) < R_{te}(xT)$

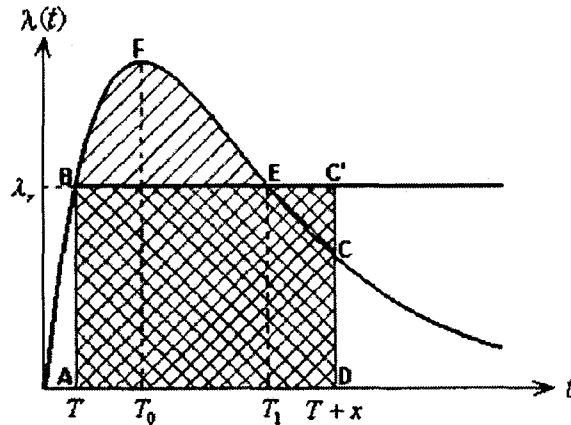
이때, 여기서, $M \equiv \lambda_{,x} - m(T+x) + m(T) = S_{ECC} - S_{BFE}$ 이다.



(a) $T_0 > T$, $T_1 < T+x$ 인 경우



(b) $T_0 > T, T_1 \geq T+x$ 인 경우



(c) $T_0 \leq T$ 인 경우

그림 3. 고장강도가 증가하다가 감소하는 경우

대부분의 실제적인 케이스는 정리 1.1의 경우나 정리 1.2의 경우 둘 중 하나에 관련된 것이다.

4. 최적소프트웨어 발행시각에 대한 영향

상기 고찰로부터 상이한 신뢰도 정의를 사용하게 되면 상이한 신뢰도값을 얻게 되어 상이한 신뢰도기준 결정법을 도출하게 된다. 본 항에서는 최적 소프트웨어 발행에 대한 상이한 신뢰도개념을 도입할 때의 영향을 고찰하고자 한다. 이는 소프트웨어 신뢰도모델 적용에 이것이 매우 중요하기 때문이다.

소프트웨어 개발에서 중요한 과제는 소프트웨어의 최적발행시각을 결정하는 일이다. 소프트웨어에서 성취해야 할 필수요건이 바로 최저 신뢰도 레벨이므로, 상이한 소프트웨어 신뢰도 정의를 사용하게 되면 산출된 최적 발행시각에 영향을 미칠 것이다. 그러므로, 그 상이점을 조사하여 좀더 적절한 신뢰도 개념을 적용하는 것이 필요하다.

최적 발행시각 문제는 일반적으로 다음과 같이 공식화한다.

$$R(x|T) \geq R_0 \quad (8)$$

인 조건하에서

$$C(T) \quad (9)$$

를 최소화한다.

식(8)에 적용하는 소프트웨어 신뢰도가 테스트신뢰도이다. 즉,

$$\exp\{-[m(T+x) - m(T)]\} \geq R_0 \quad (10)$$

사실상, 소프트웨어가 T 의 시간 단위동안 테스트되어 고객에게 발행되므로 운영기간중에는 신뢰도 성장이 없다. 그러므로, 신뢰도를 운영신뢰도 요건으로 언급하는 것이 더욱 더 적

절하다. 즉,

$$\exp[-\lambda(T)x] \geq R_o \quad (11)$$

식(9)와 (10)에 있는 최적 발행시각 문제는 아래와 같이 이후 P1으로 언급하고 식(9)와 (11)에 있는 최적 발행시각 문제는 P2로 언급하기로 한다.

정리 2.1 $t \geq 0$ 에서 $\lambda(t)$ 가 단조감소하면,

케이스 1 : If $R_{op}(x|0) \geq R_o$, then $T_{P1}^* = T_{P2}^* = T_C$

케이스 2 : If $R_{te}(x|0) \geq R_o > R_{op}(x|0)$, then
 $T_{P1}^* = T_C$ and $T_{P2}^* = \max\{T_C, T_R^2\}$

케이스 3 : If $R_{te}(x|0) < R_o$, then $T_{P1}^* = \max\{T_C, T_R^1\}$
 and $T_{P2}^* = \max\{T_C, T_R^2\}$

정리 2.2 $\lambda(t)$ 가 처음에 증가하다가 감소하며, $T_o \leq T$ 이면

케이스 1 : $R_{op}(x|T_o) \geq R_o$ 이면 $T_{P1}^* = T_{P2}^* = \max\{T_C, T_o\}$ 이다.

케이스 2 : If $R_{te}(x|T_o) \geq R_o > R_{op}(x|T_o)$, then
 $T_{P1}^* = \max\{T_C, T_o\}$, $T_{P2}^* = \max\{T_C, T_R^2\}$

케이스 3 : If $R_{te}(x|T_o) < R_o$, then $T_{P1}^* = \max\{T_C, T_R^1\}$
 and $T_{P2}^* = \max\{T_C, T_R^2\}$

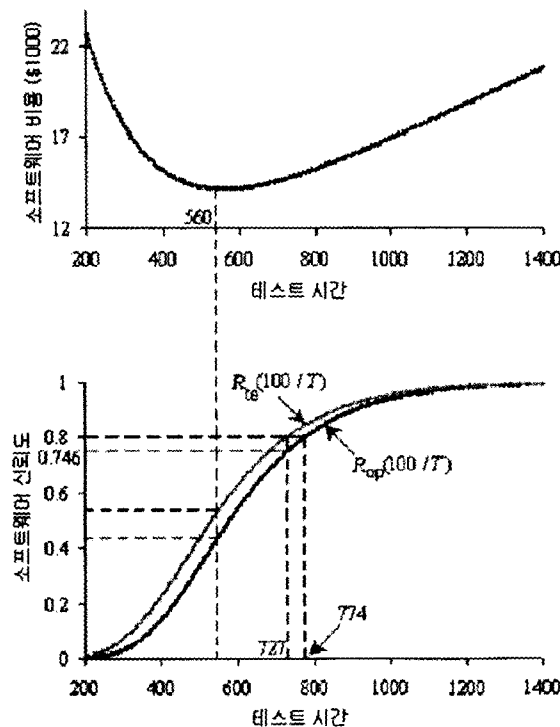


그림 4. 발행시각과 비용과의 관계

정리 2로부터 언제나 $T_{P1}^* \leq T_{P2}^*$ 의 관계가 성립하는 것이 분명해졌다. 이는, 최적발행시간에 있어서 테스트 신뢰도를 택하면 좀더 짧은 T^* 를 얻을 수 있다는 것을 말해준다. 이는 요구되는 테스트시간에 대해서 과도한 최적 추정에 이르게 하는 것이 되어 소프트웨어가 요구되는 신뢰도 레벨에 이르기 전에 소프트웨어를 부적절하게 발행하는 것이 된다. 사실상, 고객에게 의미가 있는 것은 운영신뢰도이므로, 최적발행문제는 P1보다 P2로 공식화해야 한다.

5. 결론

소프트웨어시스템은 일단 발행되면 신뢰도가 더 이상 개선되지 않으므로 고장 패턴과 행위에 변화가 있게 된다. 본 논문에서는 두 가지 소프트웨어 신뢰도 개념 즉, 상이한 단계에서 유용한 테스트신뢰도와 운영신뢰도를 구분하여 연구하였다. 그 두 개의 신뢰도 개념은 서로 다르므로 주의하여 사용해야 한다.

최적 소프트웨어 발행 시각에 상이한 개념을 적용한 영향을 조사하였다. 테스트신뢰도개념을 사용하게 되면 부정확한 결과가 초래되어 결정방법론을 그르치게 된다. 고객에게 중요한 것은 운용신뢰도이므로, 소프트웨어 발행시각 문제와 기타 다른 관련 결정론법적인 공정에 운용신뢰도개념을 채택해야 한다.

감사의 글
본 연구는 한국과학재단 목적기초연구(2000-2-30300-001-2) 지원으로 수행되었음