

소프트웨어의 운전중 결함 예측 기법

최규식, 건양대학교 IT 학부
 장원석, 건양대학교 IT 학부
 e-mail : che@konyang.ac.kr

Estimating Defects of Software During Operational Use

Che Gyu Shik

*Div. of Information Technology, Konyang University

요약

본 논문에서는 운전 단계중의 상용소프트웨어 성장활용을 설명할 수 있고 또 현장 고장 데이터로부터 활용성장을 예측하는데 관계되는 인자를 결합할 수 있는 새로운 모델을 개발한다. 이 모델은 상용 소프트웨어의 실제 활용이 시간의 멱수 함수로 나타난다는 가정으로부터 생기는 웨이블 분포에 근거한다. 선형신뢰도모델은 잔여결함의 평균크기와 작업량이 일정하고 결보기 결함밀도가 실제 결함밀도와 동일하다는 가정 하에 유도된다. 기하학적모델은 결함을 수정함에 따라 평균결함크기가 기하학적으로 감소한다는 가정에 있어서 차이가 있다. 한편, Rayleigh모델은 잔여 결함의 평균크기가 시간에 따라 선형적으로 감소한다는 가정에 있어서 차이가 있다.

본 논문에서는 소프트웨어의 신뢰도 요인의 거동을 가정하여 이러한 다양성을 수용하기 위한 모델링을 하였다.

1. 서론

1970년대 초반 이후 많은 SRGM이 개발되었다. 대부분의 소프트웨어 신뢰도 모델들은 시스템 테스트 단계에서 관찰되는 고장 패턴에 근거를 두어 이러한 고장패턴을 유효수명 기간동안의 소프트웨어 시스템 신뢰도에 투영하려 시도한 것들이다. 어떤 모델들은 소프트웨어의 운영신뢰도를 예측하는 외에 고객에게 발행될 시점에 코드 내에 잔여하고 있는 결함의 수를 추정하기도 한다. 이러한 잔여 결함들은 운전기간중에 발생하는 소프트웨어 고장의 주요 원인이 된다.

소프트웨어 신뢰도는 그것이 설계 범위 내에서 사용될 때 주어진 시스템이 탑재된 장비상에서 소프트웨어의 고장 없이 주어진 기간 동안 작동할 확률이다.

2. 모델 구축

상용 소프트웨어가 운영중일 때 잔여 결함을 발견하는 비율을 예측하는 모델을 개발하기 먼저 결함을

확인시키는 중요한 가정을 수립할 필요가 있다. 결함은 소프트웨어가 규정에 따라 작동될 때의 고객에 보고하는 결과로서 생기는 것으로 확인되었다.

트라젠베르그와 같은 연구자는 소프트웨어 운전중에 소프트웨어 결함을 접하게 될 때 소프트웨어 고장이 발생하는 것으로 가정하여 소프트웨어 신뢰도에 대한 일반적인 이론을 개발하였다.

1. wfd-결함은 원시코드상의 결함으로 정의한다.
2. 고장발생, 총 결함 발견, wfd-결함발견 과정은 시변(time-varying) 이항식(binomial)이다.

3. 결함크기/consequence는 일정하고 동일하다. 모든 잔여 결함은 동일한 가능성으로 고장을 발생시킨다.

$$du/de = s = k_1 \quad (1)$$

4. 실제 결함 밀도와 결보기 결함 밀도는 동일하다.

$$de/dx = \rho_c = k_2 \cdot r \quad (2)$$

5. 노동량은 역일의 멱함수(power function)이다. 이것이 기존 모델과 다른 핵심사항이며, 소프트웨어를 사용하는 고객의 수가 증가한다는 것에 근거를 두고 있다.

$$dx/dt = w_c = k_3 \cdot t^k \quad (3)$$

6. 결함 발견 즉시 이를 수정하며 이 수정은 완벽하다.(새로운 오류, 결함 또는 다른 장애사항이 도입되지 않는다.)

$$\lambda_{wfd} = -dr/dt \quad (4)$$

3. 모델개발

3.1 시간함수로서의 고장강도

(4)로부터 WFD모델을 유도할 수 있다.

$$du/dt = s \cdot \rho_c \cdot w_c = k_1 \cdot k_2 \cdot r \cdot (k_3 \cdot t^k) \quad (5)$$

(용도 성장율(usage growth rate))

$$\lambda_{wfd} = -dr/dt = k_4 \cdot r \cdot t^k = -dr/dt \quad (6)$$

코드는 N개의 결함을 포함한상태에서 현장에서 수명을 시작한다. 결함을 사용 전 시간에 걸쳐서 제거하여 어떤 시각 t에서는 r개의 잔여결함만 남아있게 된다. (6)을 시간으로 적분하면 시간에 대한 결함의 잔여평균수가 된다.

$$\frac{dr}{r} = -k_4 t^k dt, \int_N^r \frac{dr}{r} = -k_4 \int_0^t t^k dt \quad (7)$$

$$r = N \cdot \exp\left(-\frac{k_4 t^{k+1}}{k+1}\right) \quad (8)$$

여기서, 고장강도를 얻는다.

$$\lambda_{wfd} = k_4 N t^k \cdot \exp\left(-\frac{k_4 t^{k+1}}{k+1}\right) \quad (9)$$

이것이 코드에서 웨이블 pdf에 원래의 결함수 N을 곱한 형태이다. 웨이블 pdf는 다음과 같다.

$$(\beta/a) \cdot (t/a)^{\beta-1} \cdot \exp[-(t/a)^\beta] \quad (10)$$

3.2 파라미터 설명

(9)의 k는 (5)의 용도성장율로부터 왔으며, $\beta - 1$ 로 설정한다. 그러므로, 우리는 β 를 용도성장율에 1을 더한 것으로 설명할 수 있다. α 는 단순히 누적 척도파라미터이다. 우리는 이 모델을 두 개의 파트로 구성된 것을 보일 수 있다.

- 다항식적 성장 - 용도성장
- 지수적 감쇠 - 시스템당 wfd-고장강도 감소

이 두부분의 곱이 복합적 wfd-결함검출비를 만든다. 그림 1은 이 두 부분을 구분하여 그린 그래프의 예이며, 시스템의 수에서 가설적 용도 성장을 그린 것과 시스템당 고장강도 감쇠를 그린 것이다. 예제 파라미터는 $\alpha=1$, $\beta=1.5$ 이다. $\beta=1.5$ 는 역일시간 $(t/a)^{0.5}$ 의 함수로서 반 선형(sub-linear)인 용도성장율과 부합된다. 실제로, WFD모델을 소프트웨어고객이 보고한 현장결함검출시간들에 맞추어 α & β 를 평가

한다.

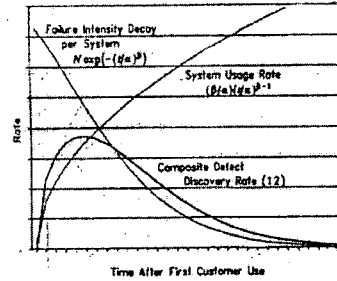


그림 1. 용도 성장, 고장강도 감쇠, 복합 결함검출비

3.3 결함총수 평가

결함검출시간들을 이용하여 조건 likelihood를 이용하여 수치적으로 최대화하여 α & β 를 평가한다.

$$L_c(t_i; \alpha, \beta) = \frac{\prod_{i=1}^{n_c} (\beta/a)(t_i/a)^{\beta-1} \cdot \exp[-(t_i/a)^\beta]}{\{(1 - \exp[-(t_c/a)^\beta]\}^{n_c}}, \quad 0 < t_i < t_c \quad (11)$$

(11)을 시간으로 적분하여 모델 n_c 를 얻는다.

$$n_c = N \cdot \text{weib}(t_c/a, \beta) = N[1 - e^{-(t_c/a)^\beta}] \quad (12)$$

모든 가용정보 즉, t_c 인 순간까지 검출된 모든 결함을 이용하여 N을 구한다.

$$N = \frac{n_c}{\text{Cdf}(t_c)} = \frac{n_c}{\text{weib}(t_c/a, \beta)} \quad (13)$$

3.4 예제

$\alpha=11.7$ 개월, $\beta=1.5$, $t_c=14$ 개월, $n_c=74$, $N=101.4$ 이다. 그림 2는 WFD모델로서 가설적현장결함발견데이터에적합시키려한것이다. 결함검출시간들은 수직 막대기로서 나타난 14개월에 걸친 것으로 나타나 있다. 원활한 곡선은 적합시킨 모델로부터 구한 평균 결함검출비이다. 실행행선은 검출해야할 잔여결함을 나타낸다.

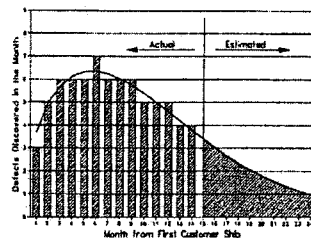


그림 2. 잔여소프트웨어결함 산정

4. 모델을 관찰데이터에 적합시키기

두 개의 현장보고결함검출시간집합이 N을 예측하기 위한 WFD모델과 그 용법을 보여준다. 코드의 라인수와 제거기의 수를 근사화한다.

데이터집합은 개발된 IBM 소프트웨어에서 구했다. 결함은 특별한소프트웨어를 위해개발된 코드에 나타나는 것들이고, 그러한소프트웨어발행의사용자에의해서보고된 것이거나 그 제품의 후속 발행의 사용자에의해서 보고된 고장결과로 검출되었다.

4.1 데이터집합 #1 (제어기 소프트웨어) 결함예측

$$\hat{\alpha} = 11.14m, [10.89, 11.53]m$$

$$\hat{\beta} = 1.514, [1.23, 1.81]$$

$$T_c = 36M, N_c = 111 \text{ 결함}$$

이 산정치를 이용하면

$$\hat{N} = \frac{n_c}{1 - \exp[-(36/11.14)^{1.514}]} = 111.3 \quad (14)$$

우리가 36개월간을통하여 11개의결함을 관찰해왔으므로, 이모델로 보면우리가 소프트웨어 내에있는 모든 결함을 발견한 것이된다.

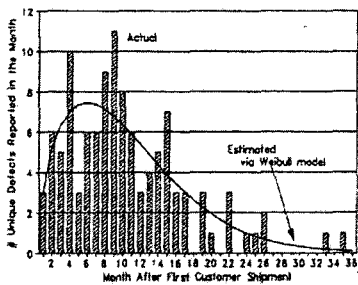


그림 3. 빈도분포 : 데이터집합#1(제어기소프트웨어)

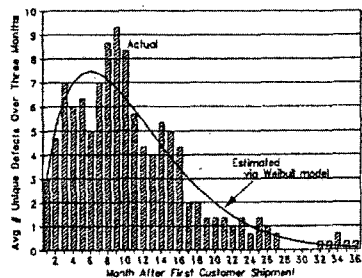


그림 4. 원활하게 만든 빈도분포 : 데이터집합 #1(제어기소프트웨어)

그림 3~5에서는 실제데이터 및 적합시킨 WFD모델의 그래프를 보여주고 있다. 표시에서 흠어진 일부분의 데이터를 제거하기위해 월간 빈도그래프를 추가하는 외에 원활하게작성한 그래프(그림 4)를 보았다. 문제의 월을 중심으로 3개월간의 평균치를 그려서 데이터를 원활하게 그렸다. 예를들면 5번째월은 4~6월에 검출된 결함수의 평균치이다. 이것과 Cdf(그림 5)는 매우 잘 적합된 전체경향을 보이고 있다.

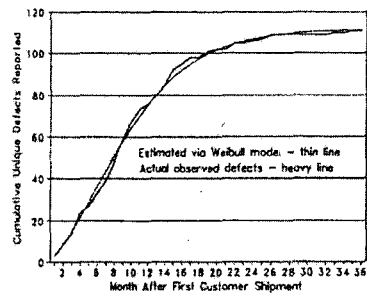


그림 5. Cdf : 데이터집합#1(제어기소프트웨어)

4.2 데이터집합 #2 (소프트웨어제품) 결함예측

$$\hat{\alpha} = 18.7\text{월}, [16.2, 28.4]\text{월}$$

$$\hat{\beta} = 1.53, [1.26, 1.80]$$

$$t_c = 24\text{월}, n_c = 357 \text{ 결함}$$

이 산출치와 (16)의 데이터를 이용하면 결과가

$$\hat{N} = \frac{n_c}{1 - \exp[-(24/18.7)^{1.53}]} = \frac{357}{0.769} = 464.2 \quad (15)$$

이다. 광역 s-신뢰구간에서는 분포파라미터와 N에서 상당한 불확실성이 있음을 보여준다. 24개월에 걸쳐서 357개의 결함을 검출했으므로 더 보고되어야 할 결함은 107개인 것으로 산출/예측할 수 있다. 그림 6-8에서는 WFD모델 산출을 따른 소프트웨어 제품 데이터의 그래프를 보이고 있다.

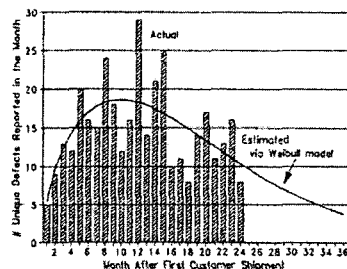


그림6. 빈도 분포 : 데이터집합#2(소프트웨어 제품)

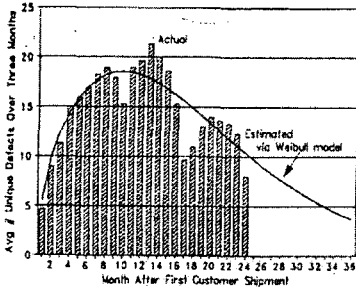


그림7. 원활화 빈도 분포 : 데이터집합#2(소프트웨어 제품)

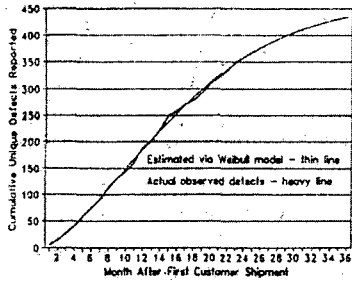


그림 8. Cdf : 데이터집합#2(제어기소프트웨어)

실을 인식해야 한다.

5.2 월간 결함 산출 - 데이터집합 #2(소프트웨어 제품)

소프트웨어 제품에 대한 N 은 18개월에서 95% UCL이 571 결함으로 낮아질 때 우선 어느 정도의 자신감을 가지고 산출할 수 있다. 그러나, 그 시점에서의 $\hat{N}=360$ 결함은 그 다음 6개월간 464개로 증가한다. 이는 18개월에서 낮은 값에서 29% 상승한 것이다. WFD 모델이 너무 난관적인 것으로 나타났다. 즉, 결함 검출율이 실제 현상보다 더 빠르게 감소되는 것을 예측한다. 소프트웨어 제품에 대한 원활한 빈도 그래프(그림7)에서는 후속 제품 발행 후에 상응하는 20개월을 중심으로 하는 낮은 피크치가 이러한 저평가의 원인이 된다는 것을 보여준다. 이 모델이 제품의 후속 발행에 의해서 발생하는 결함 검출율의 명백한 자국이 된다고 예측할 수는 없다.

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

5. 전체 결함의 월간 산출

WFD 모델에 관한 관심항목은 N 을 얼마나 빨리 합리적으로 산출해내느냐 하는 것이고 또, 오랜 시간에 걸쳐서 그 산출치가 안정되느냐 하는 것이다. 어떤 특수한 시간에 두 가지 예제 데이터에 대한 \hat{N} 을 만들었다. 데이터집합 #1(제어기 소프트웨어)에 대한 모델파라미터는 36개월간의 현장 검출시간 단위에 걸쳐서 산출되었으며, 데이터 집합 #2(소프트웨어 제품)는 24개월이 걸렸다.

5.1 월간 결함 산출 - 데이터 집합 #1(제어기 소프트웨어)

소프트웨어가 고객이 사용할 수 있게 된 후 제어기 소프트웨어에 대한 N 을 합리적으로 산출할 수 있게 되는 최초의 시간은 16개월이다. 이것은 우리가 95% UCL(655개의 총 결함)을 얻을 수 있는 첫 번째 점이다. 16개월에서의 $\hat{N}=118$ 결함이 이점을 넘어서 안정적으로 남아있다. 18개월 후 높은 $\hat{N}=118$ 결함, 낮은 $\hat{N}=106$ 결함, 최종값 $\hat{N}=111$ 주위의 +1% 및 -5%의 변량이 상당히 양호한 것이다. 18개월에서 우리가 이미 88%의 N 을 검출했다는 사