

소프트웨어 신뢰성 평가

정혜정*, 임성준**, 정영은**, 신석규**

*평택대학교 디지털 응용정보학과

**TTA 시험인증연구소, 소프트웨어 시험인증센터

e-mail : jihjung@ptu.ac.kr, joons@tta.or.kr, vejung@tta.or.kr, skshin@tta.or.kr

Evaluation of Software Reliability

Hye-Jung Jung*, Sung-Joon Lim**, Yung-Eun Jung** Seok-Kyoo Shin**

*Dept of Digital Applied Information and Statistics, Pyeong-Taek University

**Telecommunications Technology Association, Software Quality Evaluation Center

요 약

소프트웨어의 품질 특성은 크게 6가지로 나누어 평가되어지고 있다. 기능성, 신뢰성, 사용성, 효율성, 유지보수성, 이식성의 품질 특성중에서 신뢰성에 대한 평가가 상당히 중요하다는 인식을 받고 있다. 본 연구에서는 소프트웨어 신뢰성에 대한 평가 방법을 제시하고 소프트웨어를 시험하는 기관이나 소프트웨어 개발업체에서 적용할 수 있는 소프트웨어 신뢰성 평가 방법론에 대해서 연구하고 고장데이터를 적용하여 소프트웨어 신뢰성을 평가하는 평가 결과를 제시하였다.

1. 서론

소프트웨어 신뢰성에 대한 평가는 중요하다는 인식이 오래전부터 거론되어지고 있으며 특히 산업용 소프트웨어나 안전성을 요하는 임베디드 소프트웨어의 경우 신뢰성에 대한 평가 방안을 개발하기 위한 연구가 많이 진행되어지고 있다. 소프트웨어 품질 평가를 위해서 국제 표준 ISO/IEC 9126에서 제시하는 신뢰성에 대한 평가는 기능성(Functionality), 사용성(Usability), 유지보수성(Maintainability), 효율성(Efficiency), 이식성(Portability)과 같이 다른 품질 특성에 대한 평가 방안과 다소의 차이가 있으므로 이것을 평가하기 위한 평가 방안에 대한 연구는 필요하다고 보여진다.[1, 2] 국내의 시험 기관에서 적용하고 있는 국제 표준 ISO/IEC 9126의 소프트웨어 품질 평가를 좀더 정량적으로 평가할 수 있는 방안에 대한 연구를 하기 위하여 현재 ISO/IEC 25000이라는 이름으로 SQuaRE(Software Quality Requirement Evaluation) 프로젝트를 진행중에 있다. SQuaRE 프로젝트는 소프트웨어의 품질 평가에

있어서 좀더 정량적인 평가 방안을 제시하고 쉽게 적용할 수 있는 방안을 모색하기 위해서 연구를 진행하고 있으며 특히 신뢰성에 대한 평가에서는 MP(Measurement Primitive) 클래스를 구성하여 평가 방법을 제시하고 있다.[9, 11] 그러나 이러한 평가 방안이 제시된다 하여도 소프트웨어 품질에 대한 인식이 부족한 일반 개발 업체에서 적용하는 것이 그리 쉽지 않다. 또한 신뢰성을 평가하기 위해서는 국제 표준에서 제시하는 기본적인 평가 메트릭을 만족하는 형태의 평가가 진행되어야 하나 이러한 것을 평가할 수 있는 기반 지식이 부족하다고 보여진다. 본 연구에서는 국제 표준에서 제시하고 있는 소프트웨어 신뢰성평가를 위해서 소프트웨어 신뢰성 성장 모델에 적용하는 방법을 제시한다. 또한 제시된 모델을 통해서 추정된 모수를 이용하여 소프트웨어 신뢰성을 평가하는 방법에 대해서 제시한다. 제시된 방법을 업체에서 적용할 수 있도록 적용 방안에 대해서 제시한다. 본 연구는 2장에서 소프트웨어 신뢰성 성장 모델을 소개하고 3장에서는 국제 표준에서 제시하고 있는 신뢰성에 대한 평가 메트릭을

제시한다. 4장에서는 소프트웨어 시험을 통해서 얻어진 고장데이터를 이용하여 신뢰성을 평가하는 방법에 대해서 제시한다. 5장에서는 이와 같은 연구 결과를 비교하고 앞으로의 연구 과제에 대해서 제시한다.

2. 소프트웨어 신뢰성 성장 모델

소프트웨어 신뢰성 성장 모델에 대한 연구는 1970년 이후로 본격적으로 시작되었으며 소프트웨어 신뢰성 평가를 위해서 계속적으로 진행되어지고 있다. [4, 5, 6, 10] Jelinski-Moranda(1972)에 의해서 연구되어진 소프트웨어 신뢰성 성장 모델은 지수함수의 성질을 이용하여 신뢰도를 평가하였다. 본 모델의 기본형은 아래의 식 (1)과 같다.

$$f(t_i) = \phi(N-i+1)exp^{-\phi(N-i+1)t_i}$$

$$L(t_1, \dots, t_n) = \phi^n \prod_{i=1}^n (N-i+1) exp^{-\phi \sum_{i=1}^n (N-i+1)t_i} \dots \dots \dots (1)$$

위의 (1)식을 추정하고자 하는 모수에 대해서 각각 편미분하여 추정된 결과는 아래의 식 (2)와 같다

$$\frac{1}{N} + \frac{1}{N-1} + \dots + \frac{1}{N-n+1} = \frac{n \sum_{i=1}^n t_i}{\sum_{i=1}^n (N-i+1)t_i} \quad (2)$$

$$\hat{\phi} = \frac{n}{\sum_{i=1}^n (N-i+1)t_i}$$

여기에서 N은 소프트웨어 개발 당시 포함하고 있는 전체 오류 수이며 $\hat{\phi}$ 는 실패율 이다.

위의 (2)식에서 구한모수의 추정 값은 소프트웨어에 포함되어 있는 전체 오류의 수 N과 실패율 $\hat{\phi}$ 구하며 이것을 통해서 소프트웨어 신뢰성을 평가하고 신뢰성 품질 특성 값을 산출할 수 있다.

본 모델에 대한 신뢰성 R(t)와 고장간 평균 시간 (Mean Time To Failure)는 아래의 식 (3)과 같다.

$$R(t) = exp^{-(N-i+1)\phi t}$$

$$MTTF = \frac{1}{(N-i+1)\phi} \dots \dots \dots (3)$$

Jelinski-Moranda 모델은 실패율이 항상 고정된 상수값이므로 고장이 발생되었을 경우 각 고장에 대한 실패가 발생되어질 비율이 항상 같다고 볼 수 있다. 그러나 고장 사이의 실패율은 변화되어진다는 이론 하에 연구되어진 모델이 DFI (Decreasing Failure Intensity)모델이다. 본 모델에 대한 분포함수와 최우추정법(Maximum Likelihood)에 의해서 구해진 모수의 추정식은 (4)와 같다.

$$f(t_i) = \phi(N-i+1)^\alpha exp^{-\phi(N-i+1)^\alpha t_i}$$

$$\hat{\phi} = \frac{n}{\sum_{i=1}^n (N-i+1)^\alpha t_i} \quad (4)$$

$$\alpha \left(\frac{1}{N} + \frac{1}{N-1} + \dots + \frac{1}{N-n+1} \right) = \frac{n \alpha \sum_{i=1}^n (N-i+1)^{\alpha-1} t_i}{\sum_{i=1}^n (N-i+1)^\alpha t_i}$$

여기에서 N은 소프트웨어 개발 당시 포함하고 있는 전체 오류의 수이며 $\hat{\phi}$ 는 실패율이고 α 는 척도 모수 이다.

(4)식에서 구해진 추정 모수를 이용하여 소프트웨어 신뢰성과 실패간 평균 시간을 추정하면 아래의 식 (5)와 같다.

$$R(t) = exp^{-(N-i+1)^\alpha \phi t}$$

$$MTTF = \frac{1}{(N-i+1)^\alpha \phi} \quad (5)$$

DFI 모델에서 α 값이 1인 경우는 Jelinski-Moranda 모델과 같으며 α 값에 따라서 분포의 경향을 파악할 수 있다. 다음은 비동질적 포아송 과정 (Nonhomogeneous Poisson Process, NHPP)의 성질을 이용하여 소프트웨어 신뢰성을 평가한 기본 모델을 연구한 것은 Goel-Okumoto로써 모델에 대한 기본 분포는 아래 식 (6)과 같다.

$$m(t) = a(1 - e^{-bt}) \dots \dots \dots (6)$$

$$\hat{a} = \frac{n}{1 - e^{-bt_k}}$$

$$\sum_{i=1}^k \frac{n_i(t_i e^{-bt_{i-1}} - t_{i-1} e^{-bt_i})}{(e^{-bt_{i-1}} - e^{-bt_i})} - \frac{nt_k e^{-bt_k}}{1 - e^{-bt_k}} = 0$$

(6)번 식에서 추정하여야 할 모수 a는 소프트웨어 개발 당시 포함하고 있는 전체 오류의 수이며 b는 실패강도이다. 추정하여야 할 모수의 추정 방법은 최우추정법을 사용하여 추정하였다. 추정된 모수를 이용하여 신뢰도를 측정하면 아래의 식 (7)과 같다.

$$R(t;s) = e^{-a(e^{-bs} - e^{-b(s+t)})} \dots\dots\dots(7)$$

본 연구에서 제시된 소프트웨어 신뢰성 성장 모델은 가장 기본이 되는 모델을 소개하였으며 본 모델에 모수를 추정하여 신뢰성을 평가할 수 있는 방안이 연구되어져야 한다.[4, 5, 8, 9]

3. 국제 표준에서 제시한 신뢰성 평가메트릭

국내의 시험기관에서 소프트웨어 품질 평가를 위한 국제 표준모델로 ISO/IEC 9126을 참조하고 있다. ISO/IEC 9126에서 ISO/IEC 9126-2가 소프트웨어 외부 품질에 대한 평가를 제시하고 있다. 본 표준 문건에 제시되어있는 소프트웨어 품질은 총 6개의 품질 특성에 대해서 제시하고 있으며 6개 품질 특성으로는 기능성, 신뢰성, 사용성, 유지보수성, 효율성, 이식성 등으로 구성되어져 있다. 6가지 품질 특성 중에서도 기능성이나 사용성에서 제시된 평가 메트릭은 쉽게 적용할 수 있으나 신뢰성이나 유지보수성의 경우 소프트웨어 품질을 평가하기 위해서 평가 메트릭에 적용하는데에는 다소의 어려움이 있다.

특히 ISO/IEC 9126에서 제시하고 있는 신뢰성의 부 특성인 성숙성은 기존의 시험자료를 이용하여 앞으로 시험을 통해서 얻게 되어질 소프트웨어의 고장파 결함에 대한 자료를 얻어야 하는 어려움이 있다. 시험기간 동안 얻어진 고장파 결함의 자료를 이용하여 소프트웨어 신뢰성 성장 모델(Software Reliability Growth Model)에 적용하여 예측을 하여야 하는 문제가 있으므로 이론적으로 해결되어져야 할 어려움

이 있다. <표 1>은 ISO/IEC 9126에서 소프트웨어 신뢰성 중 성숙성에 관련된 평가메트릭이다.

<표 1>ISO/IEC9126의 신뢰성 중 성숙성 평가

메트릭명	세부항목	계산식	값범위
예상잠재 고장밀도	NPFI	예상된 잠재고장의 수	X=ABS(NPFI-NAFI)/SIZE 0<=X
	NAFI	실제 검출된 고장의 수	
	SIZE	제품크기	
예상잠재 결함밀도	FPFU	예상된 잠재결함의 수	X=ABS(NPFU-NAFU)/SIZE 0<=X
	NAFU	실제 검출된 결함의 수	
	SIZE	제품크기	
고장밀도 (결함밀도)	NFAI	검출된 고장의 수	X=NFAI/SIZE 0<=X
	NAFU	검출된 결함의 수	
	SIZE	제품크기	
고장예결	NRFI	해결된 고장의 수	X=NRFI/NAFI 0<X<1
	NAFI	실제로 검출된 고장의 수	
	NPFI	예견된 잠재적 고장의 수	
결함예결	NCFU	해결된 결함의 수	X=NCFU/NAFU 0<X<1
	NAFU	실제로 검출된 결함의 수	
	NPFU	예견된 잠재적 결함의 수	
평균고장 발생시간	TOPT	프로그램 작동시간	X=TOPT/NAFI 0<X 0<Y
	TISB	고장발생사이의 시간간격	
	NAFI	실제로 발견된 고장수	
테스트 적용범위	A	실제로 수행된 시험수	A/B
	B	요구사항충족 수행될 시험수	
테스트 성숙성	A	테스트 합격수	A/B
	B	시험절차의 수	

<표 1>에서 제시되어져 있는 예상 잠재 고장밀도나 예상 잠재 결함밀도 등의 평가 항목을 참조하여 보면 예상되는 결함과 고장 수를 예측하여야만 평가되어질 수 있도록 구성되어져 있다. 본 연구를 위해서 제시한 2장의 소프트웨어 신뢰성 성장 모델에 적용하여 예상잠재 고장수와 결함수를 예측하여야 한다. 4장에서는 소프트웨어 품질 평가를 담당하고 있는 기관에서 시험을 통해서 얻어진 고장데이터를 적용하여 신뢰성을 평가한 사례에 대해서 소개한다.

4. 신뢰성 평가 시뮬레이션

소프트웨어 품질 평가를 위해서 소프트웨어의 시험과정에서 얻어진 고장데이터를 소프트웨어 신뢰성 성장 모델에 적용하여 모수를 추정한 예이다.

<표 2>의 데이터는 소프트웨어 시험과정에서 고장이 발생하였을 경우 고장 시간에 대한 자료이다

<표 2>고장 시간에 대한 데이터 단위:분

고장시간에 대한 자료
115, 0, 83, 178, 194, 136, 1077, 15, 15, 92, 50, 71
606, 1189, 40, 788, 222, 72, 615, 589, 15, 390,
1863, 1337, 4508, 834

<표 2>의 고장 발생 시간에 대한 26개 자료에 대하여 고장에 대한 예측값은 26.35정도로 예측이 되어지고 실패율은 0.0002225로 예측되어진다.

<표 3>고장 시간에 대한 데이터 단위:분

고장시간에 대한 자료
28, 34, 19, 7, 1, 2, 3, 54, 55, 5, 60, 59, 100, 30
30, 60, 56, 32, 32, 30, 93, 117, 50, 70, 120, 32
28, 194, 46, 151, 89, 19, 60, 281, 355, 5, 120, 60
21, 274, 661, 492, 480, 160, 140, 74, 526, 960, 180
60, 240, 540

위의 <표 3>에 있는 고장 시간에 대한 자료는 시험기관의 의뢰를 받아서 소프트웨어 시험 동안 발생한 고장 관련 자료를 가지고 총 남아 있는 오류수에 대한 예측을 하였다. 본 시험 결과에서 총 52개의 고장 발생데이터에 대하여 53개의 오류가 있을 것으로 예측되었으며 실패율은 0.000443으로 예측되었다. 이와 같이 고장 발생 시간에 관련된 데이터에 대하여 소프트웨어 신뢰성 성장 모델에 적용하게 되면 앞으로 시험 동안 남아 있을 고장 수를 예측할 수 있게 되며 이것을 통해서 신뢰성을 예측할 수 있다. [10, 11]

5. 신뢰성 수행 절차 및 앞으로의 연구과제

본 연구에서 발표한 것과 같이 소프트웨어 신뢰성을 측정하기 위해서는 먼저 소프트웨어 시험기간 동안 발생한 고장 발생 시간에 대한 데이터가 필요하다. 일정시간 시험을 완료하기 전까지의 고장이 발생했을 때 시간을 기록하고 고장 간 시간을 측정하여 소프트웨어 신뢰성 성장 모델에 적용한다. 신뢰성 성장 모델에 적용하게 되면 시험 종료 시점에서 소프트웨어에 남아 있는 고장 수를 예측하게 되며 이 예측 값을 <표 1>에 제시한 메트릭에 적용하면 신뢰성에 관련된 값을 정량적으로 평가할 수 있다. 본 연구는 소프트웨어 품질 향상이란 측면에서 소프트웨어 신뢰성 평가를 위한 연구이다. 신뢰성 평가는 소프트웨어 신뢰성 성장 모델에 적용하여

신뢰성을 평가하여야 만 정확한 평가 결과를 산출할 수 있다. 현재의 연구는 소프트웨어 신뢰성 성장 모델에 대한 연구를 통해서 소프트웨어의 신뢰성 $R(t)$ 를 평가하는 연구가 진행되어지고 있으며 국제 표준에서 제시하고 있는 신뢰성에 대한 평가 메트릭과 결합하여 연구가 진행되지 않고 있다. 본 연구는 이러한 측면에서 소프트웨어 품질 향상을 위한 방법으로 신뢰성 평가 결과를 품질 특성 평가 메트릭에 적용하는 방안에 대해 제시하였다. 앞으로 연구과제는 이러한 신뢰성에 관련된 평가를 업체에서 쉽게 적용할 수 있도록 하는 방안을 연구하는 것이며 신뢰성 성장 모델에 적용하여 가장 적은 오차 범위 내에서 신뢰성을 측정할 수 있도록 연구되어야 할 것이다. 신뢰성 향상이란 측면에서 많이 활용되어질 수 있도록 자동화 틀을 만들어 구현하는 방법도 계속적으로 연구되어야 할 것이다.

참고문헌

- [1] ISO/IEC 9126, "Information Technology - Software Quality Characteristics and metrics - Part 1, 2, 3.
- [2] ISO/IEC 12119, "Information Technology - Software Package - Quality requirement and testing".
- [3] 정혜정, "의료용 소프트웨어의 평가기준 개발", 식품의약품안전청, 최종보고서, 2002. 12.
- [4] 정혜정, 정원태 "S/W 신뢰도 평가기술 및 품질관리 적용방안", 산업자원부기술표준원, 2003. 10.
- [5] 정혜정, 정원태, "S/W 신뢰성 성장 모델을 적용한 신뢰성평가기술", 산업자원부기술표준원, 2004. 9.
- [6] 정혜정, 정원태, "게임 소프트웨어 평가 모델 개발", 한국정보통신기술협회, 최종보고서, 2004, 11
- [7] 정혜정, 정원태, 조유덕, 정영은, 신석규 "게임 소프트웨어 평가 모델 개발", 한국정보처리학회 추계학술발표논문집, 2005, 5.
- [8] 정혜정, "소프트웨어 품질 평가 표준에 관한 연구", 사단법인 한국정보처리학회 소프트웨어 공학논문집, 2006.01.
- [9] Hye-Jung Jung, "A Study on the Standard of Software Quality Testing", ACIS, 2006,5.
- [10] 정혜정, "소프트웨어 신뢰도 품질 평가 메트릭에 대한 연구", 2006, 4.
- [11] 정혜정, "소프트웨어 품질 평가 국제 표준 적용방안에 대한 연구", 2006. 8.