

# 소프트웨어의 신뢰성 평가 방법

평택대학교 정혜정

기술표준원 박인수·양승배

## 1. 서 론

소프트웨어 신뢰성은 소프트웨어 공학과 신뢰성 분석 분야의 학문으로 현재 많은 관심을 가지고 연구가 진행 중에 있다. 1970년을 즈음하여 소프트웨어 신뢰성에 대한 연구가 진행되었다. 소프트웨어가 복잡화 되어지고 여러 분야에서 많이 활용되어지면서 소프트웨어 신뢰성에 대한 관심과 연구가 상당히 높아졌다. 하드웨어 신뢰성에 대한 관심은 상당히 많은 연구가 진행되어졌으나 소프트웨어 신뢰성에 대한 관심은 그리 큰 관심을 받지 못하였다. Weiss(1956)는 포아송 분포의 고장을 가지고 신뢰성 성장을 예측하는 연구를 발표하였고 Corcoran et al.(1964)은 소프트웨어 신뢰도를 예측하기 위한 모델을 연구하였다. Hudson(1967)은 Markov birth-death 과정으로 소프트웨어 고장 과정을 연구하였으며 Rubey & Harwick (1968)은 소프트웨어 품질에 대한 문제점을 연구하였다. 이와 같이 소프트웨어 신뢰성에 대한 연구는 1960년대에는 상당히 미비하였다. 본격적으로 소프트웨어 신뢰성에 대한 연구를 소프트웨어 신뢰성 성장 모델에 적용하여 시작한 것은 Jelinski & Moranda(1972)에 의해서 진행되었으며 지금도 여러 연구의 기초 모델로 이용되어 지고 있다. Jelinski & Moranda(1972)에 의해서 연구되어진 소프트웨어 신뢰성 성장 모델은 모델을 이용하여 소프트웨어 신뢰도를 예측하였다는 것과 소프트웨어 개발 당시에 소프트웨어에 남아있는 오류의 수를 예측하였다는 측면에서 상당히 의의가 있다고 보여진다. Jelinski & Moranda (1972)에 의해서 연구되어진 소프트웨어 신뢰성 성장 모델은 실패율의 예측값이 항상 실험단계에 관계없이 동일하다는 것과 실험단계에서 얻어진 오류는 모두 동일한 크기라는 것과 실패가 발생되었을 때 시간은 매번 독립이라는 강한 가정을 전제하고 시작하였다. 그러나 소프트웨어 신뢰도는 항상 증가하여야 한다는 기본 원칙을 만족하는 모델을 찾아야 하며 또한 실패시간에 따라서 모수를 예측하는데 오는 다소의 문제점으로 인하여 소프

트웨어 신뢰성 성장 모델에 대한 연구는 모델에서 예측되어져야 하는 모수추정에 대한 연구로 확장되었다. 현재 까지도 이러한 소프트웨어 신뢰성 성장 모델에 대한 연구는 모델과 모수 추정을 중심으로 하여 진행되어지고 있으며 이러한 연구 접근이 모두 확실적인 성질을 이용하여 추정과 예측을 하고 있다. 현재 소프트웨어 신뢰도에 대한 관심은 몇몇 기관에서 소프트웨어 인증 사업을 시작하면서 더욱 고조되었다. 특히 의료용 소프트웨어나 산업용 소프트웨어, 철도·국방 관련 소프트웨어의 경우는 신뢰성이 상당히 중요한 요소이다. 그러나 신뢰성에 대한 평가는 상당히 이론적인 접근이 쉽지 않고 소프트웨어 시험단계에서 얻어져야 측정 가능한 소프트웨어 고장에 관련된 데이터 수집이 쉽지 않아서 측정에 많은 어려움이 있다. IT 강국들은 우리 보다 먼저 소프트웨어 품질에 상당한 관심을 가지고 신뢰성 향상을 위해서 품질을 평가하는데 많은 관심을 가지고 연구하였다. 우리나라에서도 한국정보통신기술협회나 산업자원부 기술표준원 등에서 소프트웨어의 품질 인증을 시작하여 제품의 품질 향상을 위한 노력에 최대의 힘을 기울이고 있다. 신뢰성에 관련된 부분의 평가 방법에 대하여 필요한 내용을 살펴본다. 2장에서 간단히 소프트웨어 신뢰성 성장 모델을 확률적 성질에 따라서 분류하고 각 모델의 모수추정방법을 소개하고 3장에서는 양도시간 결정문제에 대하여 소개한다. 그리고 4장에서는 ISO/IEC 9126의 품질 평가를 위한 메트릭의 적용방안에 대하여 소개하고 앞으로의 연구과제에 대하여 제시한다.

## 2. 소프트웨어 신뢰성 성장 모델

### 2.1 Markov Models

Markov 모델은 소프트웨어 고장 과정을 Markov Process로 설명한 모델이다. Markov Process는 미래에 대한 것이 항상 현재의 상태에 의존한다는 가정에서 출발한 모델이다. 확률적 성질은 아래와 같다.

$$P[X(t) \geq x(t) | X(t_1) \geq x_1, \dots, X(t_n) \geq x_n] \\ = P[X(T) \geq x(t) | X(t_n) \geq x_n] \\ \text{for all } t_1 < t_2 < \dots < t_n < t.$$

Markov의 성질을 이용한 소프트웨어 신뢰성 성장 모델은 다수 있으며 가장 대표적인 모델로는 Jelinski & Moranda(1972)[7]에 의해서 연구되어진 모델이다. 이 모델에 대한 기본 가정은 아래와 같다

- ① 초기 소프트웨어 결함의 숫자는 알려져 있지 않은 고정된 상수임
- ② 발견된 결함은 즉시 제거되고 새로운 결함을 다시 만들지 않음
- ③ 고장 사이의 시간은 독립이고 지수분포를 따름
- ④ 남아있는 소프트웨어 결함은 소프트웨어 고장률에 같은 양 기여 함

위의 네 가지 가정은 현재의 시점에서 상당히 많은 연구가 진행되어지고 있으며 위의 네 가지 가정에서 좀 더 다양하게 연구되어진 특성들을 살펴보면 아래와 같다.

- ① 고장사이의 시간이 독립인 경우는 많지 않으므로 고장 사이의 시간이 앞에 일어나 고장시간에 종속적인 경우에 대한 연구가 진행되어 지고 있음
- ② 소프트웨어에 남아있는 고장은 실패률이 같다는 가정에서 현재 연구는 초기에 발견되어지는 오류의 크기와 시험 중간 단계쯤에서 발생되어지는 오류의 크기와 마지막 단계에서 발견된 오류의 크기가 다르다는 관점에서 소프트웨어 신뢰성 성장 모델에 대한 연구가 진행되어지고 있다. Markov Process를 이용한 대표적인 Jelinski & Moranda (1972)[7] 모델은 아래 식(2.1)과 같다

$$P(T_i < t_i) = \phi(N_0 - i + 1) \\ \exp[-\phi(N_0 - i + 1)t_i] \quad (2.1) \\ \text{for } i = 1, 2, \dots, N_0$$

식 (2.1)에 제시되어 있는 Jelinski & Moranda에 의해서 연구되어진 소프트웨어 신뢰성 성장 모델은 모수 추정에 있어서 최우추정법을 사용하였으며 최우추정법에 의해서 구한 해의 결과는 아래와 같다. 이 모델에서 해를 구하기 위해서는 시물레이션 과정에서 초기해에 대한 설정에 주의를 해야 한다

$$\phi = n \left[ \sum_{i=1}^n (N_0 - i + 1)t_i \right]^{-1}$$

$$\frac{1}{N_0} + \frac{1}{N_0 - 1}, \dots + \frac{1}{N_0 - n + 1} \\ = \frac{n \sum_{i=1}^n t_i}{\sum_{i=1}^n (N_0 - i + 1)t_i} \quad (2.2)$$

Jelinski & Moranda (1972)가 제안한 소프트웨어 신뢰성 성장 모델에서 모수추정을 최우추정법을 이용하여 모수를 추정할 경우 전체 오류의 수는 유한의 값을 가져야 하며 실패율은 양수이어야 한다는 기본 가정에 위배되는 경우가 발생되어지면서 소프트웨어 신뢰성 성장 모델에서 모수추정 방법에 대한 연구도 활발히 진행되었다. 가장 대표적인 모수추정법으로 최우추정법을 이용하고 있으며 그 외에도 베이지안 추정법이나 적률추정법, 깃스샘플링 방법에 의한 모수추정법등 다양한 모수 추정 방법을 소프트웨어 신뢰성 성장 모델에 적용하여 모수를 추정하고 있다. 이러한 (2.1)식의 소프트웨어 신뢰성 성장 모델은 실패율이 항상 같은 양 감소한다는 이론에 대한 모순성을 지적하고 실패율의 크기가 다르다는 가정으로 모델을 연구한 것으로 대표적인 DFI(Decreasing Failure Intensity) 모델은 (2.3)식과 같다.

$$\lambda(i) = \phi [N_0(i-1)]^{\alpha} \quad i = 1, 2, \dots, N_0 \\ \text{Power Type DFI Markov Model} \\ \lambda(i) = \phi [\exp^{-\beta(N_0-i+1)} - 1] \quad (2.3) \\ \text{Exponential Type DFI Markov Model}$$

## 2.2 비동질적 포아송과정(NHPP: Nonhomogeneous Poisson Process)

고장의 과정을 비동질적 포아송과정(NHPP)의 성질을 이용하여 모델을 제안한 것으로 Goel & Okumoto (1979)[6]가 가장 대표적인 연구를 진행시켰으며 이 연구를 기반으로 S-Shaped NHPP 모델에 대한 연구가 진행되어졌다. NHPP 모델을 소프트웨어 신뢰성 평가를 위해서 활용하기 위한 기본가정은 아래와 같다.

- ①  $N(0) = 0$
- ②  $[N(t), t > 0]$  독립임
- ③  $P\{N(t+h) - N(t) = 1\} = \lambda(t) + o(h)$
- ④  $P\{N(t+h) - N(t) > 2\} = o(h)$

위의 가정에서  $N(t)$  는  $t$ 시간 까지 소프트웨어를 시험하여 발견되어진 소프트웨어의 오류 수를 나타내고  $o(h)$ 는 작은  $h$ 에 대하여 0으로 수렴하는 값을 나타낸

다. 위의 기본 가정을 만족하는 가장 대표적인 NHPP 모델이 Goel & Okumoto(1979)[6]에 의해서 연구되어진 소프트웨어 신뢰성 성장 모델이다.

Goel & Okumoto(1979)에 의해서 연구되어진 소프트웨어 신뢰성 성장 모델의 기본 가정은 아래와 같다.

- ① t 시간까지 발견된 고장의 누적수는 포아송분포를 따름
- ② 모든 고장은 독립이며 발견될 가능성은 같음
- ③ 모든 발견된 고장은 즉시 제거되어지며 새로운 고장을 만들어 내지 않음

위의 기본 가정을 만족하는 Goel & Okumoto(1979)에 의해서 연구되어진 소프트웨어 신뢰도 성장 모델의 평균값 함수  $m(t)$ 는 아래 식(2.4)와 같다.

$$m(t) = a(1 - \exp(-bt)), \quad a > 0, b > 0 \quad (2.4)$$

Goel & Okumoto(1979) 모델에서 사용된 모수인  $a$ 는 소프트웨어 시험과정에서 발견되어질 수 있는 전체 고장의 수이고  $b$ 는 고장의 수에 대한 실패 발생 비율을 나타내며 여기서 제시한 모수는 최우추정법에 의해서 구하였고 이 모델에서 고장률과  $t$ 시간의 시험을 거치고 남아 있는 오류수에 대한 평균값 함수의 형태는 아래의 식(2.5)와 같다.

$$\lambda(t) = \frac{dm(t)}{dt} = abe^{-bt} \quad (2.5)$$

$$\begin{aligned} \overline{EN(T)} &= m(\infty) - m(t) \\ &= a - a(1 - e^{-bt}) = ae^{-bt} \end{aligned}$$

이러한 Goel & Okumoto(1979) 모델의 경우도 발생되어지는 오류에 대한 실패강도에는 차이가 없다는 가정에서 시작하였으며 이러한 가정에 모순점을 지적하고 오류의 강도에 따라서 소프트웨어 신뢰성 성장 곡선은 일반적으로 S 자 형태를 갖는다는 가정하에

S-Shaped NHPP 모델이 연구되어졌으며 S-Shaped NHPP 모델의 대표적인 경우는 아래와 같다.

$$\begin{aligned} m(t) &= a[1 - (1 + bt)\exp(-bt)], \quad b > 0 \\ m(t) &= a[1 - (1 + \lambda_1 t)\exp^{-\lambda_1 t}] \end{aligned} \quad (2.6)$$

### 2.3 그 외 소프트웨어 신뢰성 성장 모델

테스트 과정에서 성공의 확률을 구하는 모델에서 시작된 소프트웨어 신뢰성 성장 모델로써 표본공간의 구성이 사건의 성공과 실패의 두가지 경우만 발생한다는 가정에서 모델을 제안한 것이다

$$\begin{cases} 1 & \text{if the execution is successful} \\ 0 & \text{otherwise} \end{cases} \quad (2.7)$$

식 (2.7)에서 확률  $p=P[X=0]$ 로 하면 실패의 확률이 되어지며 이 경우 소프트웨어 신뢰도  $R=1-p=P[X=1]$ 로 계산되어진다.

그 외에도 사전정보를 주고 사후확률을 구하는 방법으로 소프트웨어 신뢰성 성장 모델을 연구한 베이지안 입장에서 진행된 소프트웨어 신뢰성 모델이 있다. Littlewood & Verrall(1974)[10]에 의해서 베이지안 방법에 대한 소프트웨어 신뢰도 성장 모형에 대한 연구가 진행되어졌다. Littlewood & Verrall[10]은 소프트웨어 오류 발생시간에 대한 함수로 지수함수를 제안하였다

$$f(t_i) = \lambda_i \exp^{-\lambda_i t_i} \quad (2.8)$$

위의 모델은  $\lambda_i$ 에 대한 사전분포를 주고 사후분포를 구하여 모수를 추정하는 방법이다. 위의 모델에서 실패율에 해당되는  $\lambda_i$ 의 특성에 따라서 사전분포로는 감마 분포를 제안하였다.

$$f(\lambda_i | \alpha, \phi(i)) = \frac{\phi(i)^\alpha \lambda_i^{\alpha-1} \exp(-\phi(i)\lambda_i)}{\Gamma(\alpha)} \quad (2.9)$$

$\lambda_i = (N - i + 1)\phi$  이다. 그리고  $N$ 은 소프트웨어 개발 초기에 담고 있는 총 오류의 수이고  $\phi$ 는 실패강도이다. 위의 제안된 모델에 대한 사전분포를 이용하여 실패 시간에 대한 함수를 구하면 식 (2.10)과 같다.

$$f(t_i; \alpha, \phi(i)) = \frac{\phi(i)^\alpha}{\Gamma(\alpha)} \times \frac{\Gamma(\alpha+1)}{t_i + \phi(i)^{\alpha+1}} \quad (2.10)$$

위의 실패시간 함수에 대하여 우도함수를 구하면 (2.11)과 같다.

$$f(t_1, \dots, t_n | \alpha, \phi(i)) = \frac{\alpha^n \prod_{i=1}^n \phi(i)^\alpha}{(t_i + \phi(i))^{\alpha+1}} \quad (2.11)$$

위의 모델에서  $\phi(i) = \beta_0 + \beta_1 i$  라고 가정하고 미지의 모수를 추정하면 (2.12)식과 같이 정리된다.

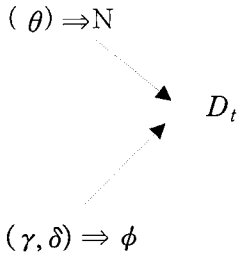
$$\sum_{i=1}^n \left( \frac{\alpha}{\beta_0 + \beta_1 i} - \frac{\alpha+1}{t_i + \beta_0 + \beta_1 i} \right) = 0 \quad (2.12)$$

$$\sum_{i=1}^n \left( \frac{\alpha i}{\beta_0 + \beta_1 i} - \frac{(\alpha+1)i}{t_i + \beta_0 + \beta_1 i} \right) = 0$$

최대우도추정치  $\widehat{\beta}_0, \widehat{\beta}_1$ 를 구해서 구하고자 하는 소프트웨어 신뢰도 성장 모형의 결과를 얻어낼 수 있다.

다음은 깃스샘플링 방법을 이용한 모수 추정방법이다. Jelinski & Moranda(1972)[7]가 제안한 소프트웨어 신뢰성 모델에서 깃스샘플링 방법을 이용한 모수 추정 방법을 간단히 제시하여 보면 아래와 같다.

$$N \sim P(\theta), \quad \phi \sim I(Y, \delta); \quad N \perp \phi.$$



그리고  $N' = N - n$ 으로 남아있는 오류의 수로 정의하고 깃스샘플링 방법에 의해서 구해보면 아래와 같다.

$$N | \phi, D_i \sim P(\theta e^{-\phi t_i})$$

$$\phi | N, D_i \sim \Gamma(\gamma + n, \delta + N t + \sum_{i=1}^n t_i)$$

소프트웨어 신뢰도 성장 모형은 실패시간에 대한 함수로 제안되어져 있으며 각각의 실패시간에 대한 성질에 따라서 여러 가지 확률모형에 대한 성질을 이용하여 제안되어져 있다.

그리고 제안된 모델에 따라서 각기 다른 방법으로 모수를 추정하는 방법이 제안되어져 있다. 각각의 소프트웨어 신뢰성 성장 모형에 대하여 모수 추정을 하고 예측을 한다. 그러나 추정된 모수에 대하여 정확성을 평가하는 방법에 대한 연구는 아직까지 진행되어지고 있지 않으므로 앞으로 연구가 진행되어져야 할 부분이다.

과거의 소프트웨어 신뢰도 성장 모형에 대한 연구에서 모수추정방법으로는 최우추정법이나 최소자승추정법에 의한 연구였으나 최근에 연구되어지는 많은 연구에서는 소프트웨어 신뢰성 모델에 대한 모수추정방법으로 깃스샘플링 방법을 이용하고 있다.

그 외의 모델에 대한 연구는 초기하분포나 이항분포 등의 확률적인 성질을 이용하여 소프트웨어 신뢰도 성장 모델을 연구하고 있다. 이러한 모델에 대한 연구는 현재에도 계속적으로 진행되어지고 있으며 좀더 확장되어 비모수분포에서 적용할 수 있는 모델로 확장 되어 연구가 진행되어지고 있다.

### 3. 양도시간 결정

소프트웨어 신뢰성 성장 모델을 이용하여 소프트웨어 신뢰도가 측정되어지면 이러한 신뢰성의 측정값을 이용하여 양도시간을 결정할 수 있다. 소프트웨어 신뢰도를

이용하여 소프트웨어의 양도시간을 결정할 수 있는데 소프트웨어를 양도하기 위해서 테스트를 계속할 것인지 멈출 것인지에 대한 문제를 Stopping Rules라고 한다 이와 같이 Stopping Rule에 의한 값이 결정되어지면 소프트웨어를 양도할 수 있는 양도시간이 결정되어지고 이것을 Release Problem이라 한다. 양도시간에 대한 문제로 테스트를 많이 하면 할수록 더욱더 신뢰도가 높은 S/W를 사용자에게 양도할 수 있으나 기업에서는 최대의 이윤이라는 경제적인 원리에서 가능한 한 빠르게 사용자에게 양도하기를 희망하고 있으므로 양도시간에 대한 결정은 소프트웨어 신뢰도를 요구조건에 만족할 때 결정하는 것으로 한다.

$$c(T) = c_1 m(T) + c_2 [m(\infty) - m(T)] + c_3 T \quad (3.1)$$

위의 식 (3.1)은 소프트웨어의 총 비용에 대한 식이며 여기에서  $C_1$ 은 테스트 기간 동안 결함을 제거하는데 드는 비용이고  $C_2$ 는 소프트웨어가 작동하는 동안에 결함을 제거하는데 드는 비용이며  $C_3$ 는 소프트웨어가 양도가 늦어지면서 드는 비용으로 소프트웨어의 단위시간당 테스트 비용이다. 또한  $m(t)$ 는 NHPP 모델에서의 평균값 함수를 나타내고  $C(T)$ 는 T 시간에 양도 되어질 소프트웨어 제품에 대한 기대비용이라 할 수 있다.

기업에서 개발된 제품에 대하여 최선의 양도 정책을 세워야 하며 이러한 양도정책은 전체 소프트웨어의 개발 코스트를 최소화 할 수 있는 양도시간 T를 찾는 것이라 할 수 있다.

### 4. 신뢰도 평가 방법

산업자원부 기술표준원이나 한국정보통신 기술협회에서는 국제표준 ISO/IEC 9126의 품질 평가 메트릭을 기본으로 하여 소프트웨어의 적합성 시험을 실시하고 있다. ISO/IEC 9126(3,4,5)의 평가 메트릭에서 간단히 신뢰성의 성숙성에 관련된 몇 개의 메트릭에 대한 값은 아래의 표와 같다

표 1에서 제시한 ISO/IEC 9126의 신뢰성에 대한 성숙성 평가 메트릭에서 예상잠재고장밀도와 예상잠재결함밀도의 경우에서 NPMI(예상잠재 고장의 수), NPFU(예상잠재 결함의 수)는 소프트웨어 신뢰성 성장 모델에 적용하여 예측되어져야 하는 값이고 이러한 값의 예측을 위해서는 개발 회사에서 개발 제품에 대하여 일정시간 시험을 하면서 수집된 고장이나 결함과 관련된 시간 자료가 있어야 한다. 그러나 국내의 실정으로 개발 제품에 대하여 시험과정을 거쳐서 고장에 관련된 시간 자료를 갖추고 있는 업체가 거의 없으므로 신뢰성에 대

표 1 ISO/IEC 9126 신뢰성 중 성숙성 평가 메트릭

| 메트릭명                   | 세부사항                       |                                   | 계산식                        | 범위  |
|------------------------|----------------------------|-----------------------------------|----------------------------|-----|
| 1. 성숙성<br>(예상잠재고장밀도)   | NPFI                       | 예상된 잠재고장의 수                       | X=ABS(NPFI-N<br>AFI)/SIZE  | 0~1 |
|                        | NAFI                       | 실제 검출된 고장의 수                      |                            |     |
|                        | SIZE                       | 제품크기                              |                            |     |
|                        | 신뢰성 성장 모델을 이용하여 NPFI를 추정한다 |                                   |                            |     |
| 2. 성숙성<br>(예상잠재결함밀도)   | NPFU                       | 예상된 잠재 결함의 수                      | X=ABS(NPFU-<br>NAFU)/SIZE  |     |
|                        | NAFU                       | 실제 검출된 결함의 수                      |                            |     |
|                        | SIZE                       | 제품크기                              |                            |     |
|                        | 신뢰성 성장 모델을 이용하여 NPFU를 추정한다 |                                   |                            |     |
| 3. 성숙성<br>(고장밀도(결함밀도)) | NFAI                       | 검출된 고장의 수                         | X=NFAI/SIZE<br>Y=NFAU/SIZE |     |
|                        | NFAU                       | 검출된 결함의 수                         |                            |     |
|                        | SIZE                       | 제품 크기                             |                            |     |
| 4. 성숙성<br>(고장해결 )      | NRFI                       | 해결된 고장의 수                         | X=NRFI/NAFI<br>Y=NRFI/NPFI |     |
|                        | NAFI                       | 실제로 검출된 고장의 수                     |                            |     |
|                        | NPFI                       | 예견된 잠재적 고장의 수                     |                            |     |
| 5. 성숙성<br>(결함해결)       | NCFU                       | 해결된 결함의 수                         | X=NCFU/NAFU<br>Y=NCFU/NPFU |     |
|                        | NAFU                       | 실제로 검출된 결함의 총수                    |                            |     |
|                        | NPFU                       | 예견된 잠재적 결함의 총수                    |                            |     |
| 6. 평균고장발생시간<br>(MTBF)  | TOPT                       | 프로그램 작동시간                         | X=TPOP/NAFI<br>Y=TSIB/NAFI |     |
|                        | TSIB                       | 고장 발생 사이의 시간 간격들의 합               |                            |     |
|                        | NFAI                       | 실제로 발견된 고장의 총수(관찰된 운전시간동안 발생한 고장) |                            |     |

한 평가가 상당히 어려운 상태이다. 이러한 문제점을 해결하기 위해서 산업자원부 기술표준원에서는 산업용 소프트웨어 개발업체를 대상으로 신뢰성에 관련된 문제를 설문조사하였으며 또한 계속적인 측정 방법에 대한 연구를 진행하고 있다. 소프트웨어 개발업체에서 개발한 제품에 대한 시험을 통해서 개발된 제품에 대하여 고장 시간에 대한 데이터를 얻을 수 없으므로 현재 인증을 담당하고 있는 인증 기관에서 시험 과정을 거쳐서 얻게 되는 고장 관련 자료를 이용하여 신뢰성을 예측하는 방법에 대하여 연구를 진행중에 있다. 몇 개의 시험데이터를 이용하여 측정하여 본 결과 상당히 정확히 예측됨을 확인할 수 있었으며 앞으로 모델에 적용하여 신뢰성을 측정하는 방법을 통한 ISO/IEC 9126의 신뢰성 관련 평가 메트릭의 측정 방법에 대한 연구는 계속적으로 진행되어질 것이다.

**참고문헌**

[1] Azuma, M., "Software Quality Evaluation System:Quality Models, Metrics and Processes - International Standards and Japanese Practice," Information and Software Technology, 1996.  
 [2] Cook, M.L., "Software Metrics : An Introduction and Annotated Bibliography,"

ACM SIGSOFT Software Engineering Notes, pp.41-60, Vol.7, No.2, April 1982.  
 [3] ISO/IEC 9126, "Information Technology - Software Quality Characteristics and metrics - Part 1, 2, 3."  
 [4] ISO/IEC 14598, "Information Technology - Software product evaluation - Part 1, 2, 3, 4, 5, 6."  
 [5] ISO/IEC 12119, "Information Technology - Software Package - Quality requirement and testing."  
 [6] Goel, A. L.& Okumoto, K., "Time Dependent Error Detection Rate Model for Software Reliability and Other Performance Measures," IEEE Trans. Reliability, R-28, 1979, pp.206-211.  
 [7] Jelinski,Z. & Moranda,P.B., 'Software Reliability Research, In Statistical Computer Performance Evaluation', New York, Academic Press, 1972, pp.465-484.  
 [8] Langberg, N. & Singpurwala, N.D, "A Unification of some Software Reliability Model," SIGM Journal on Scientific and Statistical Computation, 1985, pp.781-790.

- [9] Littlewood, B & Sofer, A., "A Bayesian Modification to the Jelinski-Moranda software reliability growth model," IEE/BCS Software Engineering Journal, pp.31-41.
- [10] Littlewood, B & Verral, J. L., "A Bayesian reliability model with a stochastically monotone failure rate," IEEE Trans. Reliability, 1974, pp.108-114.
- [11] Morris, C. N., "Parametric empirical Bayes inference: Theory and application," J. American Statistical Association, vol 78, 1983, pp.47-65.
- [12] 양해술, "임베디드 소프트웨어의 품질평가 모델 개발 연구", 한국정보통신기술협회 위탁과제, 최종보고서, 2002. 11.
- [13] 정혜정, "의료용 소프트웨어의 평가기준 개발", 식품의약품안전청, 최종보고서, 2002. 12.
- [14] 정혜정, "S/W 신뢰도 평가 기술 및 품질관리 적용 방안", 산업자원부 기술표준원, 2003. 10.
- [15] 정혜정, "S/W 신뢰성 성장 모델을 적용한 신뢰성 평가 기술", 산업자원부 기술표준원, 2004. 9. 발표논문집, 2000.5.

---

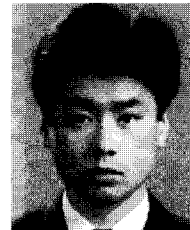
### 박 인 수



1988 원광대학교 전자과(공학사)  
 1992 원광대학교 전자계산기공학과(석사)  
 1978~1992 전북지방공산품검사소 연구원  
 1992~1994 전남지방공업기술원 기술지원과장  
 1994년~현재 기술표준원 공업연구관  
 2000년~현재 한국산업기술대학교 컴퓨터공학과 겸임교수

관심분야 : 소프트웨어공학(S/W품질평가), 임베디드 S/W, 계측·제어, 프로그래밍  
 E-mail : ispark@ats.go.kr

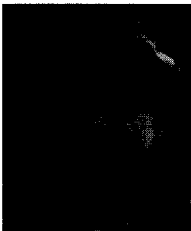
### 양 승 배



1989 숭실대학교 기계공학과(공학사)  
 1994 숭실대학교 기계공학과(석사)  
 1994~1995 대우중공업 철차연구소 연구원  
 1995~현재 기술표준원 공업연구사  
 관심분야 : 소프트웨어공학, 소프트웨어 품질평가  
 E-mail : yangsb@ats.go.kr

---

### 정 혜 정



1988 경북대학교 통계학과 조기졸업(이학사)  
 1991 경북대학교 통계학과 대학원(이학석사)  
 1994 경북대학교 통계학과 대학원(이학박사)  
 1995~현재 평택대학교 컴퓨터정보학부 부교수

관심분야 : 소프트웨어공학, 소프트웨어 신뢰성 성장 모형결정, 소프트웨어 신뢰도 측정, 소프트웨어 품질평가, 영상처리  
 E-mail : jhjung@ptuniv.ac.kr

---