

# 중첩 NHPP를 이용한 소프트웨어 신뢰도 평가 모형 연구

김도훈\* · 남경현\*\*

\* 경기대학교 응용정보통계학과

## A Study on Software Reliability Assessment Model of Superposition NHPP

Do Hoon Kim\* · Kyung H. Nam\*\*

\* Department of Applied Information Statistics, Kyonggi University

Key Words : Software Reliability Growth Model, Superposition NHPP, Mean Value Function, Intensity Function

### Abstract

In this paper, we propose a software reliability growth model based on the superposition cause in the software system, which is isolated by the executed test cases in software testing. In particular, our model assumes an imperfect debugging environment in which new faults are introduced in the fault-correction process, and is formulated as a nonhomogeneous Poisson process(NHPP). Further, it is applied to fault-detection data, the results of software reliability assessment are shown, and comparison of goodness-of-fit with the existing software reliability growth model is performed.

### 1. 서 론

최근 사회가 산업화되면서 컴퓨터 시스템의 기능 및 중요성은 날이 급증하고 있으며, 컴퓨터 시스템을 제어하는 소프트웨어의 역할 또한 중요한 부분을 차지함에 따라 하드웨어뿐만 아니라 소프트웨어 제품의 고품질화가 요구되고 있으며, 이러한 소프트웨어 시스템의 고장은 인간의 생명, 환경 및 경제에 심각한 영향을 초래하기도 한다. 예를 들어, 항공기, 원자력발전소, 인공위성 및 병원시스템 등 현대의 첨단 산업분야에서 어떤 다른 품질특성보다 소프트웨어 신뢰성이 중요한 품질요소로 취급되고 있으며, 하드웨어와 소프트웨어가 결합된 임베디드 시스템(embedded

system)의 급격한 성장으로 인해 소프트웨어 신뢰성은 제품 개발자 또는 최종 사용자 모두의 기본적인 관심사항이 되고 있다.

소프트웨어 신뢰성이란, 소프트웨어 시스템의 중요한 품질 속성으로 소프트웨어가 의도하는 기능을 제대로 수행하는가를 평가하는 것이다. 이 속성을 정량적으로 측정하는 측도중의 하나가 소프트웨어 신뢰도이다. 따라서 컴퓨터의 고장 없는 지속적인 가동을 위하여 필수적으로 요구되는 소프트웨어 신뢰성을 향상시키기 위한 노력과 관심이 매우 높아지고 있으며, 이 분야에 관한 연구가 최근 들어 중요한 연구 과제로 등장하게 되었다. 특히 컴퓨터의 기능이 확대됨에 따라 이에 필요한 소프트웨어의 규모가 매우 커질뿐만 아니라 내재된 소프트웨어에 대한 결함발견도 점점 어려워지고 있고, 또한 결함 발생시 치명적인 손실을 가져오는 경우가 많아짐에 따라 소프트웨어 신뢰성에 관한 연구는 컴퓨터의 발달과 더불어 필

\* 교신저자 knam@kyonggi.ac.kr

※ 이 논문은 2007년도 경기대학교 교내과제 연구비 지원에 의한 논문임.

수적인 과제가 되고 있다. 최근에는 소프트웨어를 개발하는 비용뿐만 아니라 소프트웨어의 고장으로 인한 손실이나 penalty 비용등도 엄청나게 증가되어 소프트웨어의 신뢰성에 대한 중요성이 매우 강조되고 있다. 이러한 상황에서 소프트웨어 신뢰성은 극히 중요한 연구 영역으로 다루어지고 있다. 소프트웨어 신뢰성의 향상은 소프트웨어의 품질에 대한 관심으로 돌려졌으며, 소프트웨어의 품질을 향상시키기 위한 많은 연구가 진행되고 있다. 즉, 소프트웨어 구성 모듈을 포함한 제품 전체의 품질보증, 제품의 평균수명 측정, 그리고 소프트웨어 제품이 일정기간 고장 없이 기능을 발휘하게 하기 위한 품질관리활동 등 신뢰성과 관련된 품질요인들이 큰 비중을 차지하고 있다. 이에 따라 신뢰성 있는 소프트웨어를 개발하기 위한 연구방법들이 지난 20여년 동안 급속히 발전하여 왔으며, 이에 대한 부분은 통계적 방법에 기초한 다양한 이론적 정립과 확률과정(stochastic process)이론 적용으로 이루어지게 되었다.

## 2. 소프트웨어 신뢰도 성장모형

소프트웨어 시스템을 개발하는 과정은 매우 복잡하여 개발과정 동안 예측할 수 없는 결과가 발생한다. 이러한 소프트웨어 시스템의 품질을 측정하고 평가하는 일이 소프트웨어 개발에서 중요한 문제로 대두되고 있으며, 소프트웨어 품질보증 및 보전에 대한 관심도 점차 증대되고 있다. 일반적으로 소프트웨어 개발 단계는 명세서(specification), 설계(design), 코딩(coding), 시험(testing)의 네 단계로 구성되어 있다. 소프트웨어 결함은 소프트웨어 개발의 마지막 단계인 시험기간(testing phase)에서 탐지되고 수정되며 이러한 결함 발견현상은 소프트웨어 신뢰도 성장모형(software reliability growth model : SRGM)으로서 기술할 수 있다.

지난 20여년 동안 다양한 통계적 모형들이 소프트웨어 신뢰성을 평가하기 위한 목적으로 제안되어 왔다. 비동질적 포아송 과정(nonhomogeneous Poisson process : NHPP)에 기초한 소프트웨어 신뢰도 성장모형(SGRM)은 실제적인 소프트웨어 신뢰성공학 면에서 매우 성공적인 이론이라는 것이 증명되었다(Musa et al., 1987). NHPP 상에서 주요 논점은 특정 시점까지의 기대고장수를 나타내는 적절한 평균값 함수

(mean value function : mvf)를 결정하는 것이다. 지금까지의 다양한 NHPP SRGM들은 모형에 맞는 다양한 가정들로 이루어져 왔다. <표 1>은 다양한 가정들로 이루어진 SRGM들과 평균값 함수를 정리한 표이다.

Goel-Okumoto(1979)는 NHPP를 기초로 한 SRGM에 관하여 논의하였다. 평균값 함수(mean value function)와 강도함수(intensity function)를 이용하여 테스트 이후에 남아있는 결함수와 신뢰도를 추정하였다. Yamada, Ohba-Osaki(1983)는 실질적으로 누적된 오류의 수는 오류가 발생하는 시간 사이에 독립을 가정하기가 힘든 경우 소프트웨어 신뢰도 함수 형태가 S-자 형태를 이루어지는 지수적 형태를 갖는다고 가정하고, 평균값 함수를 이용하여 신뢰도 성장모형을 연구하였다.

현재까지의 신뢰도 성장모형에서 고려한 수정과정(debugging process)은 결함발생시 완전수정(perfect debugging) 모형과 불완전 수정(imperfect debugging) 모형으로 분류되는데, 기존의 성장모형들에서의 가정을 보다 현실에 가깝게 확대시키고자 모형개발에 힘쓰고 있다. Pham(1993), Yamada, Tokuno and Osaki(1992) 등은 결함수정과정에서 완전수정을 통한 결함의 제거도 가능하지만, 한편으로는 결함이 수정되지 않을 뿐만 아니라, 새로운 결함 발생의 가능성을 고려한 신뢰성 성장모형을 제안하였다.

또다른 소프트웨어 신뢰도 성장모형의 기본 가정 중의 하나는 결함발견율과 관련된 가정이다. 많은 NHPP SRGM들에서 각각의 고장은 서로 독립적이고 확률적으로 발생하는 것으로, 결함탐지과정동안 동일한 분포에 따라 발생한다(Musa et al., 1987). 그러나 더 현실적인 상황하에서 고장분포는 여러 요인 즉, 작업환경, 시험전략 및 작업할당과 같은 요인들에 의해 영향을 받을 수 있다. 이러한 요인들은 소프트웨어 시험단계 동안 변화하며, 이들은 소프트웨어 고장강도함수(failure intensity function)가 비단조적으로 증가하거나 감소하는 결과를 나타낼 수 있다. 이러한 현상은 결함 발견에 대한 변화점 문제(change-point problem)로 동일시되며, Zhao(1993), Chang(1997), Shyur(2003) 등에 의해 연구되었다.

이처럼 많은 연구자들이 소프트웨어 신뢰성을 평가할 수 있는 SRGM을 연구해 왔음에도 불구하고 아직 이들을 완전하고 과학적으로 평가할 수 있는 정량

적인 측도가 미흡한 실정이다. 다시 말해, 현재 개발된 소프트웨어 수만도 엄청나며, 그 고장의 형태도 매우 다양하게 나타날 수 있다. 즉, 기존의 모형 하나로 모든 소프트웨어 신뢰도를 측정한다면 잘못된 결과를 초래할 수 있을 것이다. Brocklehurst, Lu and Littlewood(1992), Lyu and Nikora(1992), Malaiya

and Srimani(1992) 등은 특정 SRGM들이 모든 경우에 최적 모형이 될 수 없다는 연구결과를 발표하였다. 그러므로 여러 가능한 SRGM들 중에서 주어진 환경에 대한 최적의 모형을 선택하는 것이 중요한 일이다.

<표 1> 소프트웨어 신뢰도 모형과 평균값 함수

| 소프트웨어 신뢰성모형                              | 모형 형태                | 평균값함수 $m(t)$   | 비 고                         |
|--|----------------------|--|-----------------------------|
| Goel-Okumoto(G-O)<br>(1979)              | Concave              | $m(t) = a(1 - e^{-bt})$<br>$a(t) = a$<br>$b(t) = b$  | $a$ : 최초 결함수<br>$b$ : 결함발견율 |
| Delayed S-shaped SRGM<br>(1983)          | S-shaped             | $m(t) = a(1 - (1 + bt)e^{-bt})$<br>$a(t) = a$<br>$b(t) = \frac{b^2 t}{1 + bt}$   | $\alpha$ : 결함도입율            |
| Yamada imperfect debugging model 1(1992) | Concave              | $m(t) = \frac{ab}{\alpha + b}(e^{\alpha t} - e^{-bt})$<br>$a(t) = ae^{\alpha t}$<br>$b(t) = b$   | $\alpha$ : 결함도입율            |
| Yamada imperfect debugging model 2(1992) | Concave              | $m(t) = a[1 - e^{-bt}](1 - \frac{\alpha}{b}) + a\alpha t$<br>$a(t) = a(1 + \alpha t)$<br>$b(t) = b$  | $\tau$ : 변화점                |
| Chang change-point model(1997)           | Concave              | $m(t) = \begin{cases} a(1 - e^{-bt}), & 0 \leq t \leq \tau, \\ a(1 - e^{-b_1\tau - b_2(t-\tau)}), & t > \tau. \end{cases}$<br>$a(t) = a$<br>$b(t) = \begin{cases} b_1, & 0 \leq t \leq \tau, \\ b_2, & t > \tau. \end{cases}$  | $\alpha$ : 결함도입율            |
| Pham-Zhang coverage model (2003)         | S-shaped and concave | $m(t) = a \left( 1 + \alpha t - \frac{bt+1}{e^{bt}} \right) - \frac{a\alpha(1+bt)}{be^{bt+1}}$<br>$\left[ \ln(bt+1) + \sum_{i=1}^{\infty} \frac{(1+bt)^{i+1} - 1}{(i+1)!(i+1)} \right]$<br>$a(t) = a(1 + \alpha t)$<br>$b(t) = 1 - (1 + bt)e^{-bt}$  | $\tau$ : 변화점                |
| Shyur imperfect change-point model(2003) | Concave              | $m(t) = \begin{cases} \frac{a}{1-\alpha_1} [1 - e^{-(1-\alpha_1)bt}], & 0 < t < \tau \\ \frac{a}{1-\alpha_2} [1 - e^{-(1-\alpha_1)b_1\tau - (1-\alpha_2)b_2(t-\tau)}] \\ + \frac{m(\tau)(\alpha_1 - \alpha_2)}{1-\alpha_2}, & t > \tau \end{cases}$<br>$a(t) = a$<br>$b(t) = \begin{cases} b_1, & 0 \leq t \leq \tau, \\ b_2, & t > \tau. \end{cases}$ | $\tau$ : 변화점                |

### 3. 중첩 NHPP 모형

소프트웨어 시스템이 복잡해지면 고장의 원인이 하나의 강도함수에 의해서만 일어나지 않고 여러 원인이 중첩되어 발생할 수 있다. 이런 경우에 강도함수와 평균값 함수를 알고 있으면 모형화가 가능하다.

$\{N(t), t \geq 0\}$ 를 시점  $t$ 까지 탐지된 소프트웨어 누적결함수를 나타내는 계수과정(counting process)이라 하자. 여기서  $x_i$ 를 결함발생시간 간격이라 하고,  $S_n$ 은  $n$ 번째까지 발견된 결함시점이라 하면,

$$S_n = \sum_{i=1}^n x_i \text{이 되며 데이터 집합 } D_{S_n} \text{은 } \{S_1, S_2, \dots, S_n\}$$

으로 구성된다.  $m_j(t)$ 를 시점  $(0, t]$  사이에서 강도함수가  $\lambda_j(t)$ 를 갖는  $j$ 번째 요소로부터 결함이 발생되

는 NHPP라고 표현하고  $m(t) = \sum_{j=1}^J m_j(t)$  (즉,  $j$ 번째

요소에 의해 발생한 고장의 수  $m_j(t)$  ( $j=1, 2, \dots, J$ )은 서로 독립이라고 가정하면)이면,  $(0, t]$ 의 구간에서 발생한 결함수는 다음과 같은 강도함수와 평균값 함수를 갖는 중첩 NHPP가 된다.

$$m(t) = m_1(t) + m_2(t) + \dots + m_j(t), \quad (1)$$

$$\lambda(t) = \lambda_1(t) + \lambda_2(t) + \dots + \lambda_j(t).$$

본 단계에서는 앞의 연구단계에서 정의된 중첩 소프트웨어 신뢰도 모형에 포함된 모수들에 관한 추정을 수행한다. 모수추정에 사용되는 고장 데이터는 개발된 소프트웨어 시스템이 시험기간 동안에 발생한 결함의 수와 시점에 관하여 얻어진 결과로서 이러한 자료를 이용하여 신뢰성 모형을 특징지어주는 모수들의 값을 추정한다. 기존의 연구결과에서 사용되어진 추정 방법은 대부분 우도함수를 이용한 방법으로 본 연구에서도 최우추정법(maximum likelihood method)을 사용한다. 따라서  $S_1, S_2, \dots, S_n$ 을 소프트웨어 고장발생시점을 순서대로 나타내는 순서통계량 확률변수라 가정하면,  $S_1, S_2, \dots, S_n$ 의 우도함수(likelihood function)는

$$L(D_{S_n}) = \prod_{k=1}^n \lambda(S_k) \exp\{-m(S_n)\}, \quad (2)$$

이며,  $j$ 번째 요소로부터 고장이 발생하는 중첩 NHPP에 대한 우도함수는 다음과 같다.

$$L_{NHPP}(D_{S_n}) = \left[ \prod_{k=1}^n \left\{ \sum_{i=1}^J \lambda_i(S_k) \right\} \right] \cdot \exp\left\{ - \sum_{i=1}^J m_i(S_n) \right\}. \quad (3)$$

따라서 기존의 모형인 Yamada imperfect debugging(1992) 모형과, Yamada, Ohba and Osaki(1983) 모형을 중첩한다면 다음과 같은 우도함수를 유도할 수 있다.

$$L_{NHPP}(D_{S_n}) = \left[ \prod_{k=1}^n \lambda_1(S_k) + \lambda_2(S_k) \right] \exp\{-(m_1(S_n) + m_2(S_n))\}, \quad (4)$$

여기서,

$$m_1(t) = \frac{ab_1}{\alpha + b_1} [\exp(\alpha t) - \exp(-b_1 t)],$$

$$\lambda_1(t) = \frac{ab_1}{\alpha + b_1} [\alpha \exp(\alpha t) + b_1 \exp(-b_1 t)],$$

$$m_2(t) = a[1 - (1 + b_2 t) \exp(-b_2 t)],$$

$$\lambda_2(t) = ab_2^2 t \exp(b_2 t)$$

이다.

소프트웨어 신뢰도 성장모형의 모수에 대한 추정값은 식 (4)에서 양변에 자연로그를 취한 후, 로그우도함수를 모수 각각에 대하여 편미분하여 0을 대입하여 풀면 얻을 수 있다.

$$\frac{\partial \ln L_{NHPP}}{\partial a} = \frac{\partial \ln L_{NHPP}}{\partial \alpha} = \frac{\partial \ln L_{NHPP}}{\partial b_1} = \frac{\partial \ln L_{NHPP}}{\partial b_2} = 0 \quad (5)$$

따라서 시험단계에서 얻어지는 결함 데이터를 이용하여 여러 고장 유형을 찾아내고 <표 1>에서 주어진 기존 모형들을 이용하여 중첩 NHPP에 의해 모형화 한다.

평균값함수의 추정을 위해서는 위에서 얻어진 우도함수로부터 모수들의 최우추정량을 계산하는 것이 중요한 과제이며 이러한 추정량들이 얻어진다면 평균값함수  $m(t)$ 의 추정량도 얻을 수 있다. 만일  $m(t)$ 의 추정량을 알게 된다면 시점  $t$ 에 소프트웨어 시스템에 남아있는 소프트웨어 결함의 개수에 관한 추정

값도 얻을 수 있어 소프트웨어 개발자에게는 매우 중요한 정보를 제공하게 될 것이다.

### 4. 수치예제

#### 4.1 모수추정

본 논문에서 제안된 모형의 모수를 추정하기 위해 Misra(1983)에 의해 제시된 소프트웨어 결함 데이터를 이용하였으며, 이는 <표 2>와 같다. 데이터의 속성은 시험시간에 따라 탐지된 결함의 수를 측정한다.

<표 2> 소프트웨어 결함 데이터(Misra, 1983)

| 시험시간 |      |     | 시험시간 |       |     |
|------|------|-----|------|-------|-----|
| 주    | 시간   | 결함수 | 주    | 시간    | 결함수 |
| 1    | 62.5 | 15  | 20   | 25    | 5   |
| 2    | 44   | 6   | 21   | 12    | 2   |
| 3    | 40   | 8   | 22   | 55    | 5   |
| 4    | 68   | 8   | 23   | 49    | 6   |
| 5    | 62   | 8   | 24   | 64    | 9   |
| 6    | 66   | 4   | 25   | 26    | 1   |
| 7    | 73   | 4   | 26   | 66    | 4   |
| 8    | 73.5 | 8   | 27   | 49    | 2   |
| 9    | 92   | 6   | 28   | 52    | 4   |
| 10   | 71.4 | 2   | 29   | 70    | 4   |
| 11   | 64.5 | 7   | 30   | 84.5  | 9   |
| 12   | 64.7 | 8   | 31   | 83    | 6   |
| 13   | 36   | 3   | 32   | 60    | 1   |
| 14   | 54   | 5   | 33   | 72.5  | 3   |
| 15   | 39.5 | 5   | 34   | 90    | 6   |
| 16   | 68   | 8   | 35   | 58    | 6   |
| 17   | 61   | 8   | 36   | 60    | 3   |
| 18   | 62.6 | 6   | 37   | 168   | 14  |
| 19   | 98.7 | 12  | 38   | 111.5 | 10  |

식 (5)로부터 Misra(1983) 데이터에 대한 평균값 함수  $m(t)$ 의 모수인  $a, \alpha, b_1, b_2$ 를 수치적인 방법에 의해 구하였으며, 이에 대한 결과는 <표 3>와 같다.

<표 3> Misra(1983)데이터에 대한 모수 추정값

| 구 분    | $\hat{a}$ | $\hat{\alpha}$ | $\hat{b}_1$ | $\hat{b}_2$ |
|--------|-----------|----------------|-------------|-------------|
| 모수 추정값 | 363       | 0.00869        | 0.000311    | 0.000308    |

#### 4.2 적합도 비교 평가

본 절에서는 선행모형과 본 제안모형과의 적합도를 평가한다. 이를 위하여 평균제곱오차(mean square error : MSE)를 모형의 적합도를 평가하기 위한 기준으로 사용하고자 한다. 우선 평균제곱오차는

$$SSE = \sum_{k=1}^n [y_k - \hat{m}(t_k)]^2 \tag{6}$$

으로 계산되며, 이는 제안된 신뢰도 성장모형이 실제 결함 데이터와 일치정도를 평가할 수 있는 척도로 사용된다. <표 4>은 기존의 완전수정을 고려한 시험영역 SRGM과 새로운 모형의 MSE를 비교한 결과를 나타낸다.

<표 4> SSE에 기초한 모형 비교

| 구 분                                 | SSE     |
|-------------------------------------|---------|
| 제안모형                                | 159.296 |
| Goel-Okumoto(1979) 모형               | 185.288 |
| Yamada, Ohba, Osaki(1983) 모형        | 171.836 |
| Yamada imperfect debugging(1992) 모형 | 166.287 |

결과를 살펴보면 불완전 수정을 고려한 모형에서의 SSE 값이 기존 모형인 Goel-Okumoto 모형과 Yamada, Ohba, Osaki 모형 및 Yamada imperfect debugging 모형에서의 SSE 값보다 작은 것을 알 수 있다. 따라서 중첩된 제안모형이 실제 결함 데이터를 잘 적합시키는 것을 확인할 수 있다.

### 5. 결 론

본 논문에서는 소프트웨어 시험단계에서 고장의 원인이 여러 원인에 의해 중첩되어 발생 할 수 있는 경우를 고려하여 NHPT에 기초한 소프트웨어 신뢰도 성장모형을 제안하였다. 특히, 결함수정 활동에서 새로운 결함이 추가적으로 도입되는 불완전수정 환경을 고려한 평균값함수와 결함발견율이 시험시간에 따라 변화하는 경우를 고려한 평균값함수를 중첩하여 모형을 유도하였으며, 이를 통해 중첩 NHPT모형

의 모수를 추정하였다.

실제 결함 데이터를 이용하여 본 모형의 적합성을 평가한 결과 기존의 Goel-Okumoto 모형 및 Yamada, Ohba, Osaki 모형 및 Yamada imperfect debugging 모형보다 실제 데이터를 잘 적합시키고 있음을 알 수 있었다.

향후 기존의 여러 SRGM들을 이용하여 중첩 NHPP 모형을 생성한다면 보다 더 현실상황을 반영한 모형개발이 이루어질 것이다.

## 참고문헌

- [1] Brocklehurst, S., Lu, M., and Littlewood, B.(1992), Combination of Predictions Obtained from Different Software Reliability Growth Models. Proceedings of the 10th annual Software Reliability Symposium, Denver, Colorado.
- [2] Chang, I. P.(1997), An analysis of software reliability with change-point models. NSC 85-2121-M031-003, National Science Council, Taiwan.
- [3] Fenton, N. E., Pfleeger, S. L.(1997), Software Metrics: A Rigorous and Practical Approach. PWS Publishing Company, Boston.
- [4] Fujiwara, T., Yamada, S.(2003), A testing-domain-dependent software reliability growth model for imperfect debugging environment and its evaluation of goodness-of-fit. Electronics and Communications in Japan, Part 3, Vol. 86, No. 1, pp. 11-18.
- [5] Goel, A. L. and Okumoto, K(1979), Time-dependent error-detection rate model for software reliability and other performance measures. IEEE Trans. on Reliability, Vol. R-28, No. 3, pp. 206-211.
- [6] Hinkley, D. V.(1970), Inference about the change-point in a sequence of random variables. Biometrika, Vol. 57, 206-211.
- [7] Lyu, M. R., and Nikora, A.(1992), Applying Reliability Models More Effectively, IEEE Software, 9(4).
- [8] Malaiya, Y. K., and Srimani, P. K.(1992), Software Reliability Models : Theoretical Developments, Evaluation and Applications, IEEE Computer Society Press, CA..
- [9] Musa, J. D., Iannino, A., Okumoto, K.(1987), Software reliability measurement prediction application. McGraw-Hill, New York
- [10] Osaki, S.(1982), Nonhomogeneous error detection rate models for software reliability growth, in Stochastic Models in Reliability Theory, Osaki, S., Hatoyama, Y.(eds.), pp. 120-143.
- [11] Ohtera, H, Yamada, S., Narihisa, H(1990), Software reliability growth model for testing domain. Trans. IEICE, J73-D-I, 170-174.
- [12] Ohtera, H, Yamada, S., Ohba, M.(1990), Software reliability growth model with testing-domain and comparison of goodness-of-fit. Int. Symp. Reliability and Maintainability, 289-294.
- [13] Pham, H.(1993), Software reliability assessment: Imperfect debugging and multiple failure types in software development. EG&G-RAMM-10737, Idaho National Engineering Laboratory.
- [14] Pham, H. and Nordmann, L. and Zhang, X.(1999), "A General Imperfect-Software-Debugging Model with S-Shaped Fault-Detection Rate", IEEE Trans. Rel., Vol. 48, No. 2, pp. 169-75.
- [15] Ross, S. M.(1997), Stochastic Processes(6nd ed.) John wiley & Sons, New York.
- [16] Shyur, H. J.(2003), A stochastic software reliability model with imperfect debugging and change-point, The Journal of System and Software, Vol. 66, pp. 135-141.
- [17] Software reliability growth models incorporating imperfect debugging with introduced faults. Electronics and Communications in Japan, Part 3, Vol. 81, No. 4, pp. 33-41.
- [18] Software reliability models : Theoretical developments, Evaluation and Applications, IEEE Computer Society Press, CA.
- [19] Yamada, S., Ohba, M. and Osaki, S.(1983), S-shaped reliability growth modeling for software error detection. IEEE Trans. on Reliability, Vol. R-32, No. 5, pp. 475-478, 484.
- [20] Yamada, S., Osaki, S.(1983), A reliability assessment method for software products in operational phase - Proposal of an accelerated life testing model, Electronics and Communications in Japan, Part 3, Vol. 84, No. 8, pp. 294-301.
- [21] Yamada, S., Tokuno, K, and Osaki, S.(1992), Imperfect debugging models with fault introduction rate for software reliability assessment. Interna-

tional Journal of System Science, Vol. 23. pp. 2241-2252.

- [22] Zhao, M.(1993), Change-point problems in software and hardware reliability, Commun. Statistical-Theory Math. Vol. 22, pp. 757-768.