

## 오류데이터를 이용한 소프트웨어 품질평가

문 외 식

진주교육대학교 컴퓨터교육과

## A Study of Software Quality Evaluation Using Error-Data

Moon, Wae-Sik

Chinju National University of Education, Dept. of Computer Education

### Abstract

Software reliability growth model is one of the evaluation methods, software quality which quantitatively calculates the software reliability based on the number of errors detected. For correct and precise evaluation of reliability of certain software, the reliability model, which is considered to fit close to real data should be selected as well. In this paper, the optimal model for specific test data was selected one of among five software reliability growth models based on NHPP(Non Homogeneous Poission Process), and in result reliability estimating scales(total expected number of errors, error detection rate, expected number of errors remaining in the software, reliability etc) could obtained. According to reliability estimating scales obtained, Software development and predicting optimal release point and finally in conducting systematic project management.

## I. 서론

최근 컴퓨터 시스템에서 소프트웨어가 차지하는 비중이 점점 증가되어 컴퓨터 시스템의 신뢰성을 좌우하는 핵심 요소가 소프트웨어 부분이라는 것이 인식됨에 따라 소프트웨어 신뢰도에 관한 연구가 활발히 진행되고 있다[1]. 일본 규격협회에 의하면 컴퓨터 시스템의 중요 장애요인의 34% 이상이 소프트웨어 결함에 기인하는 것으로 분석하고 있다.[2] 이러한 문제를 해결하기 위하여 투자되는 개발비용이 소프트웨어 개발 전체 비용의 40% 이상이 된다. 1970년대부터 시작된 소프트웨어 신뢰도 연구는 하드웨어 신뢰도 이론에 의해서 시작하였으나 하드웨어와 소프트웨어는 근본적으로 고장 발생 유형이 다르기 때문에 신뢰도 개념 역시 다르다. 즉, 하드웨어의 고장은 어떤 우연한 사건에 의해서 일어나지만 소프트웨어의 고장은 소프트웨어 내부에 잔존하고 있는 오류가 어떠한 입력 데이터에 의해서 현실화될 때 고장이 발생한다. 이는 오류가 소프트웨어 내에 잔존한다고 해도 잔존하는 모든 오류가 고장을 일으키는 것은 아니며, 특정 데이터가 입력되지 않으면 오류와 입력된 데이터가 서로 매핑(mapping)되지 않기 때문에 고장이 발생되지 않을 수도 있다. 결과적으로 소프트웨어 개발 과정의 마지막 단계인 테스트를 완벽하게 하기 위해서는 가능한 모

든 입력 데이터에 대해서 테스트해야 하지만 이는 실제로 불가능하다. 따라서 현재의 신뢰도 기술로서는 현존하는 오류 개수를 그때까지 테스트한 결과를 기초로 예측하고 있다. 현재 대부분의 소프트웨어 신뢰도 모델들은 일정한 시각까지 테스트한 결과에서 얻어지는 오류 수를 기초로 소프트웨어 내에 잔존하는 오류 수를 예측하고 이 결과에 따라 사용자가 만족할 만한 최적 수준의 오류가 잔존하는 상태로 소프트웨어를 릴리즈(Release)시키고 있는 실정이다. 이와 같은 특성으로 인해서 소프트웨어 신뢰성에 관련된 연구들이 오류 예측의 수리적 모델을 많이 제시하고 있다. 본 논문에서는 소프트웨어 개발시 발생하는 오류데이터를 이용하여 유용한 품질평가 데이터를 도출하기 위해서 다음과 같은 수리적 모델을 사용하였다. 첫째로 소프트웨어 개발 과정에서 프로그램 내에 삽입된 오류를 테스트에 의해서 수집하여 이를 소프트웨어 품질 및 신뢰성을 정량적으로 평가하는 소프트웨어 신뢰도 성장모델(SRGM : Software Reliability Growth Model)과 둘째로 수요 예측, 경제성장등에서 사용되고 있는 Logistic곡선 및 Gompertz곡선을 이용한 통계적 해석방법을 사용하였다.[3] 본 논문에서 소프트웨어 품질을 평가하는 데이터를 도출해 내기 위하여 2장에서는 NHPP(Non Homogeneous Poisson Process)를 기본 가정으로 하는 지수형 모델, 지연S자형 모델, 습속 S자형 모델과 Logistic 및 Gompertz 곡선을 이용한

수리적 모델을 검토·분석하였으며 3장에서는 체계적인 신뢰성 데이터 분석 과정을 정리하였으며 4장에서는 기존 연구에서 사용되었던 검증된 테스트 오류데이터를 위의 5가지의 모델에 적용하여 신뢰성 평가척도를 도출하고 비교 및 분석하였다. 5장에서는 결론 및 향후 연구과제를 나타내었다.

## II. 모델 관련연구

### 1. 소프트웨어 오류

소프트웨어 오류[4]는 소프트웨어를 개발하는 전체 단계에서 발견된 결함(default)과 소프트웨어가 개발된 후 실제 환경에서 본래 주어진 요구명세서대로 작동되지 않는 모든 현상을 말하며 본 논문에서의 오류는 일반적으로 소프트웨어 공학에서 사용되는 결함(fault), 오류(error), 고장(failure)의 유형들을 구별하지 않고 총괄하여 오류라 정의하여 사용한다.

### 2. 소프트웨어 테스트 및 오류수집

소프트웨어 테스트[5]란 오류를 발견하려는 의도를 가지고 프로그램을 실행시키는 과정을 의미한다. 소프트웨어 품질보증의 척도인 테스트는 전체 개발비용중 약 절반에 가까운 비용이 든다. 따라서 소프트웨어 프로젝트 관리자는 일정 수준

의 신뢰도를 유지하기 위한 효율적인 테스트 관리가 매우 중요하다. 이와 같은 목적을 달성하기 위해서는 다음과 같은 적절한 테스트 단계와 종료가 요구되고 있다.[6]

#### (1) 테스트 단계

소프트웨어 테스트는 개발팀과 독립된 테스트 요원들에 의해 수행되어야 하며 대체적으로 단위테스트(unit test), 통합테스트(integration test), 확인테스트(validation test), 시스템테스트(system test), 인수테스트(acceptance test)의 순서적인 단계에 따라 적절한 기법이 적용된다.

#### (2) 테스트 종료 시기

테스트 전략을 활용해 테스트시 일정시간, 자원의 제한으로 인해 소프트웨어 내에 잔존하는 모든 오류를 전부 제거하기 위한 테스트과정은 불가능하다. 이러한 이유로 인해서 소프트웨어 내에 일정한 수준의 오류가 포함되어 있는 상태에서 테스트 비용과 릴리즈 시기 문제를 고려하여 테스트를 종료 할 수밖에 없다. 이때 소프트웨어 내에 잔존하는 오류 건수에 대한 정보는 신뢰도를 가시화 시킬 수 있는 중요한 데이터가 된다.

#### (3) 소프트웨어 오류수집

테스트를 통해 얻어지는 오류들은 품질

평가 예측의 척도(향후 소프트웨어 오류 발생률, 경제적인 테스트 종료시기 결정 등)가 되며 다음과 같이 수집한다. 첫째로 소프트웨어 문제보고서(software trouble report)가 만들어져 제출된다. 둘째로 제출된 소프트웨어 문제보고서는 오류수정 절차에 따라 처리되어 소프트웨어 변경보고서(software change report)가 제출되어 프로그램 수정판에 적용된다.

생 시간간격을 나타낸다.

$$(k=0, 1, 2, \dots, X_0=0)$$

변수  $N(t)$ ,  $S_k$  및  $X_k$  는 확률 변수로 취급하며 변수  $S_k$  와  $X_k$  에는 식(1)과 같은 관계가 성립된다.

$$S_k = X_1 + X_2 + X_3 + \dots + X_k = \sum_{i=1}^k X_i, X_k = S_k - S_{k-1}$$

(1)

### 3. 소프트웨어 품질평가를 위한 기본척도

소프트웨어 품질을 정량적으로 예측하기 위해서는 그 특성을 추출할 수리적 척도가 필요하며 다음과 같다.

#### (1) 기본적 확률변수

테스트를 통해 수집된 오류 수나 고장 발생시기에 관련된 기본적인 변수는 다음과 같다.[7]

$N(t)$  - 테스트시각  $t$ 까지에 발견된 총오류수(고장수)를 나타낸다.

$$\{ N(t), t \geq 0 \}$$

$S_k$  -  $k$ 번째의 고장발생 시각을 나타낸다.

$$(k=0, 1, 2, \dots, S_0=0)$$

$X_k$  -  $(k-1)$ 번째와  $k$ 번째 사이의 고장발

#### (2) 소프트웨어 신뢰성 데이터

소프트웨어 신뢰성을 각 모델에 의해 정량적으로 평가하기 위해서는 잔존하는 오류 수와 고장발생 시간이라는 두 가지 평가기준이 고려된다.[8] 전자의 평가기준에서는 잔존 오류수가 적은 것이 소프트웨어 신뢰성이 높은 것으로 간주하며 후자의 평가기준에서는 고장발생 시간 간격이 긴 것이 소프트웨어 신뢰성이 높은 것으로 판단한다. 본 논문에서는 잔존하는 오류 수를 이용하여 제시된 5가지 모델에 적용하여 신뢰성을 평가한다.

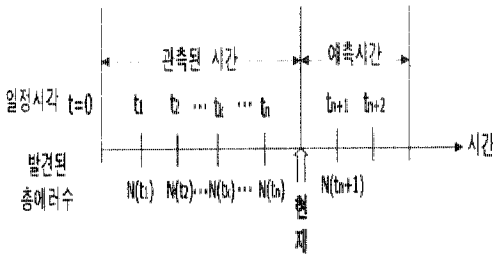
#### (3) 발견된 오류 수

일정의 테스트 시간간격  $(0, t_k]$  에 발견된 총오류수  $N(t_k)$ 에 관한 측정 데이터  $(k=1, 2, \dots, n; 0 < t_1 < t_2 < \dots < t_n)$  를 발견된 오류 수라고 하며 이 데이터를 모델에 적용하여 <그림 1>과 같이 테

스트된 과거의 데이터를 사용하여 미래의 일정시간  $t_{n+1}, t_{n+2}, \dots$  까지 발견될 총 오류 수  $N(t_{n+1}), N(t_{n+2}), \dots$  의 값을 예측한다.

#### 4. NHPP형 소프트웨어 품질 평가 모델

NHPP 모델은 발생한 고장수 또는 발견된 오류 수를 기초로한 확률 모델로서



<그림 1> 발견된 오류 수

일정한 시간간격내에 발견되는 오류수를 비동차 포아송 과정에 의해 기술한 모델로서 다음과 같은 가정을 둔다.[9]

- ① 특정시간에 남아있는 오류는 소프트웨어 고장을 일으킬수 있다.
- ② 임의의 테스트시간  $t$ 까지 발견되는 총오류수(오류누적수)  $N(t)$ 는 NHPP를 따른다.

발견된 총오류수 ( $N(t), t \geq 0$ )는 위의 두가정에 의해 식(2)와 같은 평균치함수

$H(t)$ 를 가지는 NHPP로 나타낼수 있다.

$$\Pr \{N(t) = n\} = \frac{\{H(t)\}^n}{n!} e^{-H(t)}, t \geq 0, (n=0, 1, 2, \dots) \quad (2)$$

식(2)에서  $\Pr \{N(t) = n\}$  은 테스트시간  $t$ 까지  $n$ 개의 오류가 발견되는 오류발견 사상확률을 말하며  $H(t)$ 는  $N(t)$ 의 평균치를 나타내는 평균치함수이며, 이는 테스트 시간  $t$ 까지 발견되는 총기대 오류수를 나타낸다. 테스트 시간  $t$ 에서 소프트웨어 내의 기대잔존 오류 수를  $\bar{N}(t)$ 라 할 때  $\bar{N}(t) = N(\infty) - N(t)$ 로 이때의 평균치함수는 식(3)이 된다.

$$n_r(t) \equiv E[\bar{N}(t)] = E[N(\infty) - N(t)] = a - H(t) \quad (3)$$

식(3)에서 모수  $a$ 는 테스트 시작 전에 잔존하는 총기대 오류 수를 나타낸다. 또한, 테스트가 시간  $t$ 까지 진행된 후 시간구간  $(t, t+x]$  에서 오류가 발생하지 않을 확률은 식(4)와 같이 되며 이를 소프트웨어 신뢰도라고 한다.

$$R(x | t) = \exp[-(H(t+x) - H(t))] \quad (t \geq 0, x \geq 0) \quad (4)$$

여러 가지 테스트 환경요인을 고려하여 식(2)의 평균치함수  $H(t)$ 에 적절한 함수형을 변형하여 다음과 같은 대표적인 NHPP 모델들이 제안되었으며 본 논문에서는 실측데이터를 이들 모델들에 적용, 비교하여 실측자료에 적합한 모델을 유도한다.

(1) 지수형 모델

지수형(exponential) 모델은 가장 기본적인 NHPP 모델로서 Goel과 Okumoto[10]에 의해서 제안되었다. 기본개념은 단위 시간당 발견된 오류 수는 그 시각에 소프트웨어 내에 잔존하는 오류 수에 비례하는 것으로 가정하며 평균치함수는 식(5)가 된다. 여기서 모수  $b$ 는 테스트시 발견되는 잔존하는 오류발견율을 나타내는 정수이다.

$$H_e(t) = a(1 - e^{-bt}) \quad (a, b > 0) \quad (5)$$

신뢰성 평가척도로서 기대 잔존오류수와 신뢰도는 각각 식(6)과 식(7)이 된다.

$$n_x(t) = ae^{-bt} \quad (6)$$

$$R(x|t) = \exp[-e^{-bt}m(x)] \quad (7)$$

지수형 모델은 물리적 의미가 간단하며, 모수 추정이 용이하여 미국의 AT&T사의 벨연구소를 중심으로 많이 사용되고

있다.

그리고 경험적으로 대규모 소프트웨어에 적합한 모델로 알려져 있다.

(2) 지연 S자형 모델

지연 S자형(delayed S-shaped)모델은 山田茂[11]에 의해 제안되었다. 이 모델의 기본개념은 소프트웨어 고장발생 시간에서부터 그 고장의 원인이 되는 오류를 제거하는 시간까지의 시간지연을 고려하고 있다. 즉, 테스트에 의해 오류를 발견하기 위해서는 먼저, 고장현상을 확인한 후 원인분석을 해야 한다는 개념이다. 지연 S자형 모델의 평균치함수는 식(8)이 된다.

$$H_d(t) = a[1 - (1 + bt)e^{-bt}] \quad (a, b > 0) \quad (8)$$

기대잔존 오류 수와 신뢰도는 각각 식(9)와 식(10)이 된다.

$$n_x(t) = a(1 - bt)e^{-bt} \quad (9)$$

$$R(x|t) = \exp[-a\{(1 + bt)e^{-bt} - (1 + b(t+x))e^{-b(t+x)}\}] \quad (10)$$

지연 S자형 모델은 경험적으로 비교적 중소규모의 소프트웨어 신뢰성 모델에 보다 더 적합하다고 알려져 있다.

(3) 습속 S자형 모델

습속 S자형(inflexion S-shaped) 모델은 山田茂, Ohab[12]에 의해서 제안되었으며 테스트 팀의 소프트웨어에 대한 습속도를 고려한 소프트웨어 고장 발견 사상 모델이다. 기본개념은 하나의 오류가 다른 오류와 무관하게 발견 가능한 것과 하나의 오류가 발견되기 위해서는 다른 오류가 그 이전에 발견되어 제거되어야 하는 것으로 분류한다. 평균치함수는 식(11)이 되며 모수  $c$ 는 오류발견 능력에 대한 습속도(inflexion rate)이다.

$$H_i(t) = \frac{a(1 - e^{-bt})}{(1 + ce^{-bt})} \quad (a, b, c > 0) \quad (11)$$

신뢰성 평가척도로서 기대잔존 오류 수와 신뢰도는 식(12)와 식(13)이 된다.

$$n_r(t) = \frac{a(1+c)e^{-bt}}{(1+ce^{-bt})} \quad (12)$$

$$R(x,t) = \exp\left[ -\frac{a(1+c)}{1+ce^{-bt}} e^{-bx} \right] \quad (13)$$

(4) Logistic곡선 및 Gompertz곡선을 이용한 모델

소프트웨어의 품질 평가를 위해 소프트웨어 개발단계의 테스트 공정에서 발견된

총 오류 수가 S자형 성장곡선을 나타낸다고 하여 S자형 성장곡선을 나타내는 오류 발견수 데이터에 직접 경향곡선을 적용시켜서 통계해석을 행하는데 Logistic 곡선이나 Gompertz 곡선이 흔히 채용되고 있다. 이들 각 곡선의 수렴 값을 소프트웨어 내에 잠재하는 총 오류수로하여 이것을 회귀분석에 의해 추정하는 것이 주목적을 가진다.[13]

Logistic 곡선 모델 및 Gompertz 곡선 모델은 테스트에 의해 발견된 총 오류수의 시간적 경향을 S자형 성장곡선에 의해 기술하는 것이다. Logistic 곡선은 테스트 시각  $t$ 까지에 발견되는 오류 수를 식(14)와 같이 나타낸다.

$$L(t) = \frac{k}{1 + me^{-at}} \quad (k > 0, m > 0, a > 0) \quad (14)$$

여기서  $m, a$  및  $k$ 는 정수 모수이다. 식(14)에 의해 식(15)를 도출할 수 있으며 이때의 모수  $k$ 는 테스트 시작 전에 소프트웨어 내에 잠재하는 총 오류 수를 나타낸다.

$$L(\infty) = \lim_{t \rightarrow \infty} L(t) = k \quad (15)$$

Gompertz 곡선은 테스트 시각  $t$ 까지에 발견되는 총 오류 수를 식(16)과 같이 나타낸다.

$$G(t) = ka^t \quad (k > 0, 0 < a < 1, 0 < b < 1) \quad (16)$$

여기서 모수  $a, b$  및  $k$ 는 정수이다. 식 (16)에 의해 모수  $k$ 는 테스트 시작 전에 소프트웨어 내에 잠재하는 총오류수이며 식(17)과 같이 나타낼 수 있다.

$$G(\infty) = \lim_{t \rightarrow \infty} G(t) = k \quad (17)$$

Logistic 곡선과 Gompertz 곡선 모델은 테스트 공정에 있어서 오류 발견 사상에 대하여 확률 값을 가정하지 않은 결정론적 모델이므로 Gompertz 곡선과 Logistic 곡선 모델에 확률값으로서 NHPP 가정을 도입하여 소프트웨어 신뢰도 성장 모델을 재구축할때 NHPP의 평균치 함수는 각각 식(18),(19)가 된다.

$$G(t) = G_0(t) - ka = k(a^{b^t} - a) \quad (18)$$

$$L(t) = L_0(t) - \frac{k}{(1+m)} = k \left\{ \frac{1}{(1+me^{-at})} - \frac{1}{(1+m)} \right\} \quad (19)$$

식(18)과 (19)의  $G(t)$ 와  $L(t)$ 를 NHPP의 평균치 함수로 할 때 NHPP의 성질을 만족하므로 테스트시각  $t$ 까지  $n$ 개의 오류가 발견되는 Gompertz곡선과 Logistic 곡선에 대한 오류발견 사상확률은 각각 식(20)과 식(21)이 된다.

$$\Pr\{N(t) = n\} = \frac{\{G(t)\}^n}{n!} \exp[-G(t)] \quad (n=0, 1, 2, \dots) \quad (20)$$

$$\Pr\{N(t) = n\} = \frac{\{L(t)\}^n}{n!} \exp[-L(t)] \quad (n=0, 1, 2, \dots) \quad (21)$$

식(20)과 식(21)에 의해 정식화된 NHPP 모델로부터 도출되는 Gompertz곡선과 Logistic곡선 모델에 대한 소프트웨어 내에 잔존하는 기대 잔존오류수는 각각 식(22)와 식(23)이 된다.

$$r_g(t) \equiv E[N(\infty) - N(t)] = k(1 - a^{b^t}) \quad (22)$$

$$r_l(t) \equiv E[N(\infty) - N(t)] = \frac{k(m^2 e^{-at} - 1)}{(1+m)(1+me^{-at})} \quad (23)$$

또한 테스트가 시각  $t$ 까지 진행하고 있을 때 시간구간  $(t, t+x)$  ( $x \geq 0$ )에서 소프트웨어 고장이 발생하지 않을 확률 즉, Gompertz곡선과 Logistic곡선 모델에 대한 소프트웨어 신뢰도는 각각 식(24), 식(25)가 된다.

$$R_g(x|t) = \exp[-\{G(t+x) - G(t)\}] = \exp[k\{a^{b^{t+x}} - a^{b^{t+n}}\}] \quad (t \geq 0, x \geq 0) \quad (24)$$



$$\begin{aligned}
 R(x,t) &= \exp[-\{L(t+x) - L(t)\}] \\
 &= \exp\left[-\left(\sum_{i=1}^m \frac{kx_i^{m-1} - at_i^{m-1}}{m} + \sum_{i=1}^n \frac{ax_i^{m-1}}{m}\right)\right] \\
 &\quad (k \geq 0, x \geq 0) \\
 &\quad (25)
 \end{aligned}$$

### Ⅲ. 소프트웨어 품질평가를 위한 신뢰성 데이터 분석과정

본 논문에서는 특정의 테스트 데이터에 적용하여 소프트웨어 품질평가 척도들을 도출하기 위하여 NHPP형에 따르는 신뢰도모델의 분석과정을 다음과 같이 나타낸다.

#### 1. 모델선택

본 논문에서 제시한 5가지의 NHPP형의 모델중 하나를 선택한다.

#### 2. 데이터 입력

일정 시간구간인  $(0, t_k]$  에 있어서 발견된 총오류수  $y_k$ 에 대한 데이터를 충분히 입력받는다.

#### 3. 모델 모수추정

신뢰도모델을 이용하기 위해서는 먼저 실측데이터로 모수를 추정한다. 모수를 추정하는 통계적 방법으로 여기서는 최대우도법(Method of Maximum Likelihood Method)과 최소자승법(Least Square Estimation)으로 모수를 추정한다.[14]

#### (1) 최대우도법에 의한 모수추정

지수형, 지연 S자형, 습속 S자형모델의 모수를 최대우도법으로 추정한다. 일정한 테스트 시간간격  $(0, t_k]$  에서 총오류수  $y_k$ 에 대한  $n$ 조의 측정 데이터가 실측되는 것으로 하며 이데이터에 대한 평균치함수  $H(t)$ 를 가지는 NHPP모델의 우도함수는 식(26)이 된다.

$$\begin{aligned}
 L &= \Pr\{N(t_1) = y_1, N(t_2) = y_2, \dots, N(t_n) = y_n\} \\
 &= \exp[-H(t_n)] \prod_{k=1}^n \frac{\{H(t_k) - H(t_{k-1})\}^{(y_k - y_{k-1})}}{(y_k - y_{k-1})!} \\
 &\quad (26)
 \end{aligned}$$

식(26)에서  $t_0 = 0, y_0 = 0$  으로 초기 값을 주고 양변에 자연대수를 취하면 식(27)의 대수우도함수가 된다.

$$\begin{aligned}
 \ln L &= \sum_{k=1}^n (y_k - y_{k-1}) \ln[H(t_k) - H(t_{k-1})] \\
 &\quad - H(t_n) - \sum_{k=1}^n \ln[y_k - y_{k-1}]! \\
 &\quad (27)
 \end{aligned}$$

식(27)의  $H(t)$ 에 식(5),(8),(11)의 평균치함수를 대입하고 각모델의 모수에 대해 편 미분하면 각모델의 모수 최우추정값이 구해진다.[14]

(2) 최소자승법에 의한 모수추정

오류 데이터  $(t_k, y_k) (k=1, 2, \dots, n)$  가 실측될 때 Logistic과 Gompertz 곡선 모델에 대해서 미지의 모수추정은 회귀분석으로 한다. 먼저 Logistic 곡선에 대한 추정 과정은 다음과 같다. 식(14)를  $t$ 에 대해 미분하면 식(28)과 같은 미분방정식을 얻는다.

$$\frac{dL(t)}{dt} = \frac{a}{k}[k-L(t)]L(t) \quad (28)$$

다시 식(28)에 의해 식(29)를 얻는다. 따라서 발견되는 오류수의 증가율은 그 시점 까지에 발견되는 총오류수의 1차식으로 감소하는 것을 알 수 있다.

$$\frac{dL(t)}{dt} \Big/ L(t) = \frac{a}{k}[k-L(t)] \quad (29)$$

다음은 Gompertz 곡선에 대한 추정과정은 다음과 같다. 식(16)으로부터  $t$ 에 대한 미분 결과는 식(30)과 같은 미분방정식을 얻는다.

$$\frac{dG(t)}{dt} = (\ln a)(\ln b)b' \cdot G(t) \quad (30)$$

다시 식(30)에 의해 식(31)이 되며 발견되는 오류수의 증가율은  $b'$ 에 비례하고 있는 것을 알 수 있다.

$$\frac{dG(t)}{dt} \Big/ G(t) = (\ln a)(\ln b)b' \quad (31)$$

식(29)에서 식(32)로 놓고 식(31)에서 식(33)을 놓는다면 양곡선은 모두  $Y=AX+B$  의 직선으로 나타난다. 이 관계는 양 곡선에 포함되는 모수를 추정하는 것으로 회귀분석을 이용할 때에 사용된다. 그리고 식 (29)와 식(30)에 의해 임의의 테스트 시각에 있어서 발견되는 오류 발견수의 증가율은 Logistic 곡선 모델에서는 그 시점까지에 발견된 총 오류수의 1차식으로서 감소하고 Gompertz 곡선 모델에서는 기하급수적으로 감소하고 있는 것을 알 수 있다

$$\begin{aligned} Y &= \frac{dL(t)}{dt} \Big/ L(t), \\ X &= L(t), \\ A &= -\frac{a}{k}, \\ B &= a \end{aligned} \quad (32)$$

$$\begin{aligned}
 Y &= \ln \left\{ \frac{dG(t)}{dt} / G(t) \right\}, \\
 X &= t, \\
 A &= \ln b, \\
 B &= \ln \{ (\ln a)(\ln b) \}
 \end{aligned}
 \tag{33}$$

정값  $\hat{a}$ ,  $\hat{k}$ ,  $\hat{b}$  가 구해지면 식(36)과 같은 평균치함수가 구해져 시간에 따른 추이과정을 알 수 있다.

$$\hat{G}(t) = \hat{k} \cdot \hat{a}^{b^t}
 \tag{36}$$

#### 4. 평균치함수 $H(t)$ 추정

지수형, 자연S자형, 습속 S자형모델의 평균치함수  $H(t)$ 의 최우추정값  $\hat{H}(t)$ 에서 식(34)를 얻는다.

$$P_{i,N}(t=n) = \frac{\{\hat{H}(t)\}^n}{n!} \exp[-\hat{H}(t)] \quad (n=0,1,2,\dots)
 \tag{34}$$

Logistic, Gompertz곡선 모델에서는 다음과 같이 평균치함수를 얻는다. 추정된 모수에 의해 누적오류수 즉, 평균치 함수를 추정할 수 있다. Logistic 곡선 모델의 모수 추정 값이  $\hat{a}$ ,  $\hat{k}$ ,  $\hat{m}$  이라고 하면 평균치 함수는 식(35)가 되므로 시간  $t$ 의 추이에 따른 평균치 함수의 과정을 추정할 수 있다.

$$\hat{L}(t) = \frac{\hat{k}}{1 + \hat{m}e^{-t\hat{a}}}
 \tag{35}$$

Gompertz 곡선 모델의 경우도 모수 추

#### 5. 적합도 검정

평균치함수 추정이 실측된 데이터에 적합한가를 통계적으로 판정하기 위하여 적합도 검정을 한다. 여기서는 지수형, 자연 S자형, 습속 S자형모델에서는 콜모루그로브 스미르노(Kolmogorov-Smirnov) 적합도 검정법(K-S검정법)을 적용하였으며 Logistic 및 Gompertz 곡선 모델에서는 회귀분석에 의한 F검정을 사용하였다.

##### (1) K-S검정법에 의한 적합도 검정

오류발견수 데이터  $(t_k, y_k)(k=1, 2, \dots, n)$ 가 실측될 때에 식(37)로 되는 K-S통계량을 계산한다.

$$\begin{cases}
 D = \max_{1 \leq i \leq n} (D_i) \\
 D_i = \max \left\{ \left| \frac{H(t_i)}{H(t_n)} - \frac{y_i}{y_n} \right|, \left| \frac{H(t_i)}{H(t_n)} - \frac{y_{i-1}}{y_n} \right| \right\}
 \end{cases}
 \tag{37}$$

K-S 검정법에서는 데이터로부터 계산된 검정 통계량  $D$ 와 위험율이  $\alpha$  (1% 또는 5%)

일 때의 기각한계와 비교된다. 식(37)에 대해서 자유도가  $n$ 일 때의 기각한계  $D_{\alpha}$ 와 비교하여 식(38)의 조건을 만족하면 위험률  $\alpha$ 에서 실측된 테스트 데이터에 대해 식(34)의 지수형, 지연S자형, 습속S자형 모델은 적합하다고 판정한다.

$$D < D_{\alpha} \quad (38)$$

식(38)의 조건에 맞지 않게 되어 선택한 모델이 적합하지 않다면 추가 데이터를 수집하여 다시 모델을 선택하여 적합한 모델을 추정하거나 다른 NHPP 모델을 적용한다.

## (2) F 검정법에 의한 적합도 검정

모수추정을 기초로 단순회귀분석에 있어서는 변수  $y$ 의 증감은 변수  $x$ 의 증감에 비례하는 것에 의하여  $y = a + bx$  이라고 하는 1차식이 성립하는 것으로 가정한다. 여기서  $a$  및  $b$ 는 정수 모수이다. 실측된 데이터가  $(x, y)$  평면 상에서 직선 경향을 나타낸다면 위의 식 가설이 바른 것으로 된다. 먼저 최소자승법(least square method)에 의해 모수  $a$  및  $b$ 를 추정한다. 다음으로 실측 데이터  $(x_i, y_i) (i=1, 2, \dots, n)$  에 대하여 회귀직선에서 추정하는 것에 의미가 있는지 여부를 판정할 필요가 있다. 즉,

독립변수로서 취급되는 요인이 종속변수의 변동에 영향을 미치고 있는지를 통계적으로 검정한다. 실측치  $y_i$ 의 추정값  $\hat{y}_i$ 는 식(39)가 된다.

$$\hat{y}_i = \bar{y} + \hat{b}(x_i - \bar{x}) \quad (i=1, 2, \dots, n) \quad (39)$$

독립변수  $x$ 에 대한 회귀관계에서 설명되는 변동은 식(40)으로 된다.

$$S_b = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2 = \hat{b}^2 \sum_{i=1}^n (x_i - \bar{x})^2 = \hat{b}^2 S_x \quad (40)$$

한편 회귀관계에서 설명되지 않는 변동오차는 식(41)이 된다.

$$S_e = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n e_i^2 = S(\hat{a}, \hat{b}) \quad (41)$$

여기서 종속변수의 변동이 식(42)와 같이 분해된다.

$$S_r = \sum_{i=1}^n (y_i - \bar{y})^2 = S_b + S_e \quad (42)$$

따라서 실측 데이터에 대한 회귀직선의 분산분석표는 <표 1>과 같이 정리된다.

<표 1> 회귀직선  $y = a + bx$  분산분석표

요인	자유도 f	변동 S	불편분산 V	분산비 F
회귀 (b)	1	$\hat{b}^2 \sum_{i=1}^n (x_i - \bar{x})^2$	$\hat{b}^2 \sum_{i=1}^n (x_i - \bar{x})^2$	$\frac{V_b}{V_e}$
오차 (e)	$n-2$	$S(\hat{a}, \hat{b})$	$\frac{S(\hat{a}, \hat{b})}{(n-2)}$	
계	$n-1$	$\sum_{i=1}^n (y_i - \bar{y})^2$		

<표 1>의 분산분석표에 있어서 분산비는 식(43)을 검정 통계량으로 하여 F검정을 한다.

$$F = \frac{V_b}{V_e} = \frac{S_b}{[S_e/(n-2)]} \quad (43)$$

여기서 분산비 F는 F 분포표의 자유도

$(v_1, v_2) = (1, n-2)$  에 대한 유의수준  $\alpha$ 일 때의 F값  $F_\alpha(v_1, v_2)$  와 비교되어 식(44)의 조건을 성립하면 추정된 회귀직선이 실측 데이터에 적합하다고 판정한다.

$$F > F_\alpha(v_1, v_2) \quad (44)$$

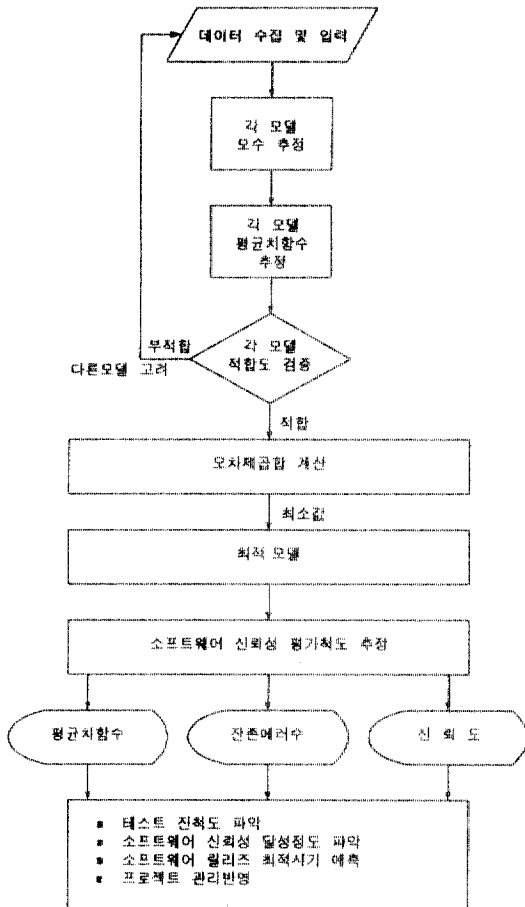
## 6. 오차제곱합에 의한 최적모델 선택

실측 데이터에 대한 최적모델의 적합성을 비교하고 평가하는 기준으로 일정시간

동안 테스트되어 발견된 총오류수인 실측 데이터들과 그때까지 추정된 추정 데이터들 간의 오차가 최소 값을 가질 때 미래의 오류 데이터를 보다 정확하게 추정할 수 있다. 이 때의 오차란 모델을 실제 실측된 데이터에 적용시켰을 때 실측된 발견된 오류 데이터의 수값과 그 추정 값과의 차이를 말한다. 보다 나은 최적모델 선택을 위해서 오차제곱합에 의한 모델 선택을 한다.[15] 이때 오차제곱합 값이 최소인 값을 가지는 모델이 실제 실측된 데이터에 보다 더 적합한 최적모델로 결정한다. 오차제곱합은 테스트에 의해서 발견된 총 오류수의 실측 데이터  $y_k$ 와 그 추정 데이터  $\hat{y}_k$ 와의 오차제곱합의 식(45)를 사용한다.

$$S = \sum_{k=1}^n (y_k - \hat{y}_k)^2 \quad (45)$$

지금까지 제시한 일련의 신뢰성 데이터 분석과정에 의한 모델 시뮬레이션으로 기대잔존 오류 수  $n_r(t)$ , 소프트웨어 신뢰도  $R(x|t)$  등의 정도 높은 신뢰성 평가척도를 추정·예측할 수 있다. 또한, 산출된 평가척도로 소프트웨어 최적 릴리즈 시기 예측 및 프로젝트 관리에 반영할 수 있다. 신뢰성 데이터분석의 시뮬레이션 과정은 <그림 2>와 같이 나타낼 수 있다.



<그림 2> 신뢰성데이터 분석과정

#### IV. 모델적용 오류데이터 분석

<표 2>의 실측 데이터는 山田茂, Ohba[16]가 사용한 데이터이다. 이 데이터는 발견된 오류 데이터의 수로서 어셈블리 언어로 작성된 약 3000 LOC(Line of Code)로 구성된 온라인 단말제어

프로그램의 기능 테스트에서 수집된 것이다. 또한, 이 데이터는 10조의 실측 데이터  $(t_k, y_k)(k=1, 2, \dots, 10)$  이며 측정단위는 일(日)수 이다.

<표 2> 10조의 검출된 실측 오류데이터

테스트 시각(日)	누적 오류 수	테스트 시각(日)	누적 오류 수
1	1	6	21
2	3	7	23
3	11	8	27
4	15	9	30
5	18	10	31

##### 1. 모델 모수값 추정

<표 2>의 오류 데이터에 지금까지 기술한 5가지의 각모델의 모수값은 식(27), 식(33)를 이용하여 최우추정값을 구할 수 있으며 이때의 각모델 모수값은 <표 3>과 같다.

<표 3> 각모델 모수값 추정

구분 모델	a	b	c	k	m
지수형	50.35	0.11			
지연S형	62.63	0.18			
습속S자형	32.41	0.60	9.00		
Gompertz	0.0049	0.6585		34.1612	
Logistic	1.5912			26.4663	186.0201

## 2. 평균치함수 $H(t)$ 추정

<표 3>에 나타난 각모델의 모수값을 각모델의 평균치함수에 대입하면 평균치함수값은 다음과 같이 계산된다.

$$\hat{H}_e(t) = 50.35(1 - e^{-0.11})$$

$$\hat{H}_d(t) = 62.63[1 - (1 + 0.18t)e^{-0.18}]$$

$$\hat{H}_f(t) = 32.41(1 - e^{-0.60t}) / (1 + 9.00 \cdot e^{-0.60t})$$

$$\hat{H}_g(t) = 34.16 \cdot 0.0049^{0.6585t}$$

$$\hat{H}_h(t) = \frac{26.466}{1 + 186.02 \cdot e^{-1.5912t}}$$

## 3. 모델 적합도 검정 및 최적모델

각모델의 K-S검정 통계량과 Gompertz, Logistic모델의 F검정 결과는 유의수준  $\alpha$ 가 5%일 때의 적합도 검정결과이며 <표 4>와 같다.

<표 4> 각 모델 적합도 검정

모 델	K-S통계량및F검정	기각한계	적합도 판정
지수형	0.214179	0.391220	적합
지연S자형	0.267618	0.391220	적합
습속S자형	0.165360	0.391220	적합
Gompertz	5.91474	5.32	적합
Logistic	0.01917	5.32	부적합

모델 적합도 판정결과 적합한 모델이 하나이면 문제가 되지 않지만 하나 이상이면 최종적으로 보다 나은 최적모델 선택을 위해서 오차제곱합에 의한 모델선택을 한다. <표 4>의 적합도 판정에서 4가지 모델이 모두 채택되므로 최종적으로 오차제곱합 값을 비교하면 <표 5>와 같다.

<표 5> 각 모델의 오차제곱합

모 델	오차제곱합
지수형	138.444506
지연S자형	72.562995
습속S자형	78.721689
Gompertz	28.994511
Logistic	144.052227

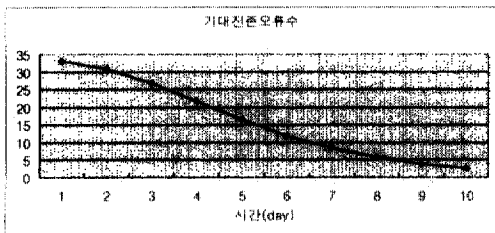
<표 5>의 오차제곱합에서 K-S검정한 3가지 모델 중에서 지연S자형 모델이 F검정에서 적합한 모델은 Gompertz 모델이 오차제곱합이 가장 적게 나타났다. 따라서 이 두모델이 실측 데이터에 적합하다고 판단된다.

그러나 실측데이터와 추정데이터간의 오차제곱합이 가장 적은 Gompertz곡선 모델이 <표 2>의 실측데이터에 보다 적합한 최적모델로 추정된다.

## 4. 테스트 종료시점의 소프트웨어 내에 잔존하는 기대 오류 수 및 신뢰도 추정

<표 2>의 실측데이터에 가장 적합한

모델인 Gompertz곡선 모델을 적용하여 나타낸 신뢰성 평가척도는 다음과 같다. 테스트 종료시점인  $t = t_{10} = 10$ 일 까지 발견된 총 오류 수(누적오류수)는 31개이며 이때, 테스트에 의해 아직 발견되지 않은 소프트웨어 내에 잔존하는 기대 잔존오류수는 식(22)에 의해 2.67270으로서 약 3개로 추정된다. 또한, 테스트 종료후 시각  $t+x=11$  일 짜의 소프트웨어 신뢰도는 식(24)에 의해 0.900184로서 약 90%가 된다. <그림 3>은 Gompertz곡선 모델을 적용하여 나타낸 기대 잔존 오류수의 발생현황을 나타낸 것이며 이는 테스트 시간의 경과에 따라서 기대 잔존 오류수가 점점 감소하여 신뢰도가 계속 증가하는 현상을 나타내고 있다.



<그림 3> Gompertz곡선 모델 적용시의 기대 잔존오류수 발생 현황

## V. 결론 및 향후연구

소프트웨어 최종단계인 테스트 단계에서 수집된 소프트웨어의 오류데이터를 이용하여 소프트웨어의 품질을 평가할 때는

오류데이터의 발생형태 또는 발생경향에 따라서 어떠한 모델을 선택하여 오류데이터를 해석하는가가 매우 중요하다. 본 논문에서는 기존의 신뢰도 모델들중 NHPP형에 따르는 오류데이터를 해석할 수 있는 신뢰도모델중 5가지의 신뢰도모델들에 대한 수리적 개념을 조사·정리하였으며 이들 모델들에 대한 신뢰성 데이터 분석 절차를 체계화하여 특정 데이터에 대한 최적모델을 쉽게 선택할 수 있도록 하였다. 또한 10일간의 테스트에서 수집한 실측데이터를 5가지 모델에 적용하여 비교한 결과 Gompertz곡선을 이용한 신뢰도 모델이 가장 적합한 모델로 추정된다. 이 모델을 실측데이터에 적용한 후 그 결과로서 테스트 종료 이후의 소프트웨어 내에 잔존하는 에러 수를 추정할 수 있으며 향후 소프트웨어의 신뢰성 달성정도 및 테스트이후 소프트웨어를 릴리즈 하여 운용단계에서의 신뢰성을 예측할 수 있다. 향후과제로서는 5가지 NHPP형의 신뢰도 모델들을 하나로 통합한 알고리즘을 만들어 프로그래밍화된 하나의 자동화된 도구를 구현하여 발전시키고자 한다.

## 참고문헌

- [1] Musa, J.D., Iannino, A. and Okumoto, K. : Software Reliability: Measurement, Prediction, Application, McGraw-Hill, 1987.
- [2] 平成元年度システムの高信頼化技術に



- 關する調査研究 (電子應用システム) 成果報告書, 日本規格協會, 1990.
- [3] Musa, J.D.: "A theory of Software reliability and its application", IEEE Trans. Software Engineering, Vol. SE-1, No.3, pp.312-327, 1975.
- [4] Capers Johnes : Applied Software Measurement : Assuring Productivity Quality, pp.279~338., 1991.
- [5] 山田茂, 大寺浩志 "ソフトウェアの信頼性~理論と 実践的應用~, ソフト・リサーチ・センター, 1990.
- [6] Thompson, Jr., W.A., "Piont Models with Application to Safety and Reliability", Chapman and Hall, 1988.
- [7] 高宗雄, "ソフトウェア信頼度成長モデルに基づく定量的 品質評價法", 九州大學學位論文, 1989.
- [8] M.L. Shooman, "Probabilistic Model for Software Reliability Prediction", in Staattistical Computer Performance Evaluaation, W. Freiberger(ed.), pp. 485-502, Academic Press, 1972.
- [9] Yamada, S., Ohba, M. and Osaki, S.: "S shaped reliability growth modeling for Software error detection", IEEE Trans. Reliability, Vol. R-32, No.5, p p.475-478, 484, 1983
- [10] Goel, A.L. and Okumoto, K : "Time dependent error detection rate model for Software Reliability Reliability and other performance measures", IEEE Trans. Reliability, Vol. R-28, No.3, pp.206-211, 1979.
- [11] 山田茂: "ソフトウェアの品質評價に關する考え方 と動向-ソフトウェア信頼度成長 モチルに基づく定量的品質評價法-", 情報處理, Vol.32, No.11, pp.1189-1202, 1991
- [12] Ohba, M., "Inflection S shaped software Reliability growth model in stochastic models in reliability theory, Osaki, S and Hatayama, Y", pp. 144-162, Springer-Verlag, Berl in, 1984.
- [13] 山田茂: "ソフトウェア信頼性評價技術", HBJ出版局, 1989.
- [14] 森田優三, "新統計概論", 日本評論社, 1974.
- [15] 市田嵩, 鈴木和幸, "信頼性の分布と統計", 日科技連出版社, 1984.
- [16] M. ohba: "Software quality=Test coverage×Test accuracy" Proc. 6th Int. Conf. Software Engineering, pp. 287-293(1982).