신뢰성 평가척도를 중심으로한 교환 소프트웨어 최적 배포 시기 결정 및 신뢰도 평가

이 재 기† · 신 상 권† · 홍 성 백†

요 약

소프트웨어 개발시 소프트웨어의 Release 시점을 예측할 수 있다면 제품의 적기 공급 및 개발자원 이용 측면에서 매우 중요한 의미를 갖는다. 본 논문에서는 신뢰성 평가 척도와 비용 평가 측면에 따른 소프트웨어 최적의 배포 문제에 대해 살펴보고 두 개의 평가 기준을 동시에 고려한 최적 배포 문제를 기술하였다. 신뢰도 평가를 위하여 신뢰도 성장 모델 중 지수형 모델과 시험능력의존형 모델을 적용하여 보았으며, 소프트웨어 신뢰성 척도에 따른 배포시기를 결정하여 보았다. 두 모델을 상호 비교해 본 결과 시험능력의존형 모델이 교환시스템 소프트웨어 개발에 더 적합함을 확인하였다.

Optimal Release Time of Switching Software and Evolution of Reliablity Based on Reliability Indicator

Jae-Ki Lee† · Sang-Kwon Shin† · Sung-Back Hong†

ABSTRACT

On the aspect of on-time production and development resource use, it is very important to predict the software release time during the software development process. In this paper, we present the optimal release problem based on the evaluation indicator and cost evaluation. And also we show the optimal release point considered with both of them. We applied the Exponential Software Reliability Growth Model(E-SRGM) and Testing-effort dependent Software Reliability Growth Model(Te-SRGM) and decided the software release time according to software reliability indicator. As a result of two models comparison, we verify the Te-SRGM is more adopted in our switching system software.

1. 서 론

신뢰성이 높은 소프트웨어를 효율적으로 개발하려면 품질, 납기, 비용의 관점에서 소프트웨어의 각 공정에 대해 검토를 하는 개발관리 기술이 필요하다. 특히, 개발의 최종 단계인 시험단계는 시험 진행상황과 사용자에게 소프트웨어를 배포할 시기를 고려하여야 한다.

개발중인 소프트웨어에 대해 시험을 중단하고 사용자에게 소프트웨어를 인도할 시기를 정하는 방법은 개발관리 기술의 중요한 요소로 작용하며, 이것을 소프트웨어의 최적 배포문제(Optimal Software Release Problem) 라고 부른다.[1]

과거의 소프트웨어 배포 문제는 납기를 중요시하여 개발관리자의 경험이나 판단에 의해 좌우되는 경향이 많았다. 그러나 최적 배포시기를 결정하는 평가기준은 소프트웨어 신뢰도 성장모델로부터 도출된 신뢰성 평가척도와 총 기대되는 개발비용을 채택하여 결정된다.

본 논문의 구성은 2장에서 소프트웨어 배포시기 결정법에 대해 알아보고 3장에서 시험능력을 고려한 신뢰도 성장모델과 지수형 성장모델을 비교, 분석하고 이를

† 정 회 원 : 한국전자통신연구원 교환서비스연구부
논문접수 : 1998년 9월 21일, 심사완료 : 1999년 1월 12일

적용하여 얻어진 결과를 토대로 소프트웨어 최적 배포 시기를 결정, 이때 예측되는 소프트웨어 신뢰도를 구한다. 마지막 4장에서 결론과 향후 추가로 다루어야 할 사항에 대해 논하고 맺는다.

2. 소프트웨어 배포시기 결정법

2.1 신뢰성 평가 척도에 의한 배포시기 결정 방법

소프트웨어 신뢰도 성장 모델에 근거한 최적 배포 문제는 모델로부터 도출한 정량적인 신뢰성 평가 척도를 설정된 목표치에 도달하는데 소요되는 총 시험 시간을 최적 배포시기로 삼는다. 즉 시험마다 검출된 결함 데이터를 누적시켜 축적된 결함수를 토대로 소프트웨어의 신뢰도를 계산해 내는 개수계측 모델로부터 고장 발생에 대한 평균치 함수 H(t)와 고장강도 함수(FDF : Failure Density Function) h(t)를 통한 NHPP(Non-Homogeneous Poisson Process) 모델을 고려할 수 있다. 이 모델의 신뢰성 평가 척도는

- 기대 잔존 에러수

$$n_r(t) = a - H(t) \tag{1}$$

- 소프트웨어 신뢰도

$$R(x|t) = e^{-H(t+x)-H(t)} \tag{2}$$

- 평균 고장시간 간격

$$MTBF(t) = \frac{1}{h(t)} \tag{3}$$

등 3개항을 채택한다. 위의 식 (1)과 식 (3)에 대해 t마다 시험을 종료할 때 기대잔존 에러수 및 평균 고장시간 간격을 고려해볼 때 그때마다 각각의 목표치를 n_0, M_0라고 하면 위의 식 (2)는 시험시각 t 일때 시험을 종료하고 이때 운용시간 x에 대한 소프트웨어 신뢰도를 R_0라고 하면 시험의 진척상황에 따라 계속 소프트웨어 내의 잔존 에러수는 감소하는 경향을 띤다. 이와 반대로 소프트웨어 신뢰도와 평균 고장시간 간격은 증가하게 되는 것을 확인할 수 있다. 즉, 소프트웨어 최적 배포시기 T^*는

- $\min[t | n_r(t) \le n_0]$,
- $\min[t | R(x|t) \ge R_0]$,
- $\min[t | MTBF(t) \ge M_0]$

를 만족하는 시험시간을 의미한다. 여기서 min{t | A}는 조건 A를 만족하는 시험시간 t의 최소치를 의미한다. NHPP 모델에서는 시험 시각 t에 대한 단조 증가함수를 의미하며, 위의 3가지 식은 총 시험시간을 T로 하였을 때 각각

$$n_r(T) = n_0,$$

$$R(x|T) = R_0(x \ge 0),$$

$$MTBF(T) = M_0$$

를 만족할 때 $T = T_1$을 최적 배포시기라 한다. 구체적으로 말하면 NHPP에 의거한 지수형 신뢰도 성장모델에 대한 최적 배포시기는

$$H(t) \cong m(t) = a(1 - e^{-bt}),\ (a, b > 0) \tag{4}$$

로 표현된다. 이때 위의 식들은

$$T_1 = \frac{1}{b} \ln[\frac{a}{n_0}],$$

$$T_1 = \frac{1}{b} \ln[\frac{-m(x)}{\ln R_0}],$$

$T_1 = \frac{1}{b} \ln[abM_0]$으로 각각 표현된다 이때 최적 배포시기 T^*가 구해지는데 그 조건은

$$n_r(0) \ge n_0,\ R(x|0) \le R_0,$$

$MTBF(0) < M_0$이다.

또 시험능력의존형 모델에 근거하여 배포시기를 결정하기 위한 평균치함수는

$$H(t) \cong m_w(t) = a(1 - e^{-\gamma W(t)}),\ (a > 0,\ 0 < \gamma < 1), \tag{5}$$

$$W(t) = \alpha(1 - e^{\beta t^m}),\ (\alpha > 0,\ \beta > 0,\ m > 0)$$

로 되며, 최적 배포시기 T^* 때의 소프트웨어의 기대잔존 에러수 및 소프트웨어 신뢰도, MTBF(Mean Time to Between Failure)는 각각

$$T^* = \sqrt[m]{\frac{-1}{\beta} \ln(\frac{1}{\alpha\gamma} \ln(\frac{n_0}{a}) + 1)},$$

$$T^* = [\,T|[e^{-\gamma W(t)} - e^{-\gamma W(T+x)}]] = -\frac{\ln R_0}{a}],$$

$$T^* = [\,T|[e^{-\gamma W(t)} - e^{-\gamma W(T+x)}] = \frac{1}{a}]$$을 얻는다.

2.2 비용 평가 기준에 의한 최적 배포 문제

소프트웨어 신뢰도 성장 이론에 근거하여 관측된 고장 데이터에 대한 해석은 실제 문제로서 매우 흥미 있는 일이다. 즉, 소프트웨어 최적 배포시기를 정하는데 있어서 상반된 요인에 대한 고려를 할 수 있는데, 소프트웨어 내에 잠재하고 있는 에러수로 평가된 소프트웨어 신뢰도와 에러를 발견, 수정하는데 시험에 투입된 공수(工數)나 유지보수 비용과의 관계이다. 다시 말해서 신뢰도 목표치를 정하고 달성된 신뢰도와 관련된 비용의 Trade-off를 그림으로 나타내는 방법이다.

시험에 투입되는 시간이 길어지면 소프트웨어 내의 에러들이 많이 발견되어 운용단계에 이르면 신뢰도는 증가한다. 이와 반대로 소프트웨어의 운용이 지연되면 시험에 허비되는 시간이 많아지고 비용이 증대하게 된다. 즉, 운용단계에 발견되는 에러가 많을수록 유지보수 비용이 많아진다. 일반적으로 운용단계에서 발견되는 에러 당 수정에 필요한 비용은 시험단계보다 높은데 시험단계로부터 운용단계로 소프트웨어를 이행하는 과정의 최적배포시기를 구하는 Trade-off 문제가 존재하게 된다. 그러면 소프트웨어 최적 배포시기를 결정하기 위한 최적 배포 문제를 구하기 위해서 수식화 해보면 시험시간이 T 일 때 총 기대 소프트웨어 비용은 아래와 같이 정의된다.

$$C(T) = C_1 H(T) + C_2[H(T_{LC}) - H(T)] + C_3 T \quad (6)$$

T_{LC} : Software life-cycle time

C_1 : 시험단계에서 발견된 에러 당 수정에 소요되는 비용

C_2 : 운용단계에서 발견된 에러 당 수정에 소요되는 비용($C_2 > C_1$)

C_3 : 시험에 소요되는 단위시간 당 비용

평균치 함수 m(t)를 지수형 신뢰도 성장모델로 가정하면 위의 식 (6)은

$$C(T) = C_1 m(T) + C_2[m(T_{LC}) - m(T)] + C_3 T \quad (7)$$

로 된다. 식 (7)을 시간 T에 대해 미분하면

$$h(T) = \frac{dm(T)}{dT} = abe^{-bT},$$

$$\frac{dC(T)}{dT} = -(C_2 - C_1)h(T) + C_3$$

로 되고 좌변을 0로 하면 $h(T) = \frac{C_3}{C_2 - C_1}$이 된다. 이것은 단위 시험시간 당 발견되는 에러수를 표현하는 강도함수를 의미한다. 즉, T = 0 일 때

$$h(0) = ab,\ h(0) \geq \frac{C_3}{C_2 - C_1} \quad (8)$$

이되며, h(T)에 대한 유한값의 해를 구하면 $T_0 = \frac{1}{b}\ln[ab\frac{(C_2 - C_1)}{C_3}]$가 된다. 시험시간 T에 대해서 최적 배포시기를 정리하면 $0 < T < T_0$ 일때 $\frac{dC(T)}{dT} < 0$, $T > T_0$ 일때 $\frac{dC(T)}{dT} > 0$가 되어 최적 배포시기 T^*는

$T^* = \min[T_0, T_{LC}]$로 정리될 수 있다.

이와 같이 소프트웨어 개발 비용에 대한 연구는 Boehm [2], Wolverton[3]에 의해 제기되어 활발히 연구되고 있다. 또 신뢰도 모델 중 개발비용을 평가기준으로 삼아 소프트웨어의 최적 배포시기를 연구한 예로는 Forman and Singpuwalla[4], Koch and Kubat[5], Goel-Okumoto[6], Yamada and Osaki[7], 大寺浩志[8] 등이 있다.

2.3 신뢰도 및 평가 기준을 동시에 고려한 최적 배포 문제

소프트웨어 신뢰도와 평가 기준을 동시에 고려한 최적 배포 시기 결정법은 신뢰성 평가 척도와 소프트웨어 비용을 가지고 달성된 신뢰도에 대한 비용을 평가해 보는 방법으로서 투입된 비용의 유효성(cost-effectiveness)을 파악해 보는 매우 흥미 있는 일이다. 즉, 소프트웨어 비용과 신뢰도라는 2개의 평가 기준을 고려하여 최적 배포시기를 구하는 것이다. NHPP 모델에 대한 평균치 함수 H(t)와 시험을 진행시켜 달성된 신뢰도 목표치 R_0를 달성할 때 식 (6)의 총 기대 소프트웨어 비용 C(T)를 최소로 하는 경우의 총 시험시간을 정하는 것이다. 즉, 운용시간 x(x > 0)에 대해 정리하면

$$\text{minimize } C(T),$$
$$R(x|T) \geq R_0,\ T \geq 0$$

를 만족하는 총 시험시간 $T^* = T$를 구하면 된다. 이것

을 다시 지수형 신뢰도 성장모델의 평균치 함수를 m(t)라고 하면 위의 식은

$$\min imize[c_1 m(T) + c_2[m(T_{LC}) - m(T)] + c_3 T],$$
$$e^{(-[m(T+x)-m(T)])} \geq R_0,\ T \geq 0$$

로 정리되며, 위의 식 (8)로부터

$c_2 > c_1 > 0,\ c_3 > 0,\ x \geq 0,\ 0 < R_0 < 1$이고

(1) $h(0) > \frac{c_3}{c_2 - c_1}$, $R(x|0) < R_0$를 만족하면 이때 최적 배포시기 T^*는

$$T^* = \max[T_0,\ T_1],$$

(2) $h(0) > \frac{c_3}{c_2 - c_1}$, $R(x|0) \geq R_0$이면

$$T^* = T_0,$$

(3) $h(0) \leq \frac{c_3}{c_2 - c_1}$, $R(x|0) < R_0$이면

$$T^* = T_1,$$

(4) $h(0) \leq \frac{c_3}{c_2 - c_1}$, $R(x|0) \geq R_0$이면

$T^* = 0$가 된다.

3. 지수형 성장 모델과 시험 능력을 고려한 시험능력의존형 모델의 비교 분석

이 절에서는 시험에서 발견된 소프트웨어 결함이 발견 즉시 수정(correction)된 것으로 가정하여 시험시각마다 발견된 에러를 누적시켜 평가하는 지수형 신뢰도 성장 모델과 소프트웨어 내에 잠재하고 있는 에러들을 찾아내기 위해 투입된 시험능력을 기반으로 하는 시험능력의존형 신뢰도 평가 모델에 대해 살펴본다.

3.1 지수형 신뢰도 성장 모델

"단위시간 당 결함 검출수는 소프트웨어 내에 잔존하는 에러수에 비례한다"는 개념으로 출발한 지수형 성장 모델은 고장 발생간격은 서로 독립적이고 소프트웨어 내에 잠재하고 있는 초기에러는 유한하지 않다고 가정한 모델이다. 이 모델의 일반형인 평균치 함수 m(t)와 소프트웨어 신뢰도 R(x | t)는 각각

$$m(t) \cong H(t) = a(1 - e^{-bt}),\ (a, b > 0)$$

a : 시험 시작전 소프트웨어내 잔존 에러수

b : 에러(고장) 발견율

$$R(x|t) = e^{[-H(t+x) - H(t)]} \quad (9)$$

로 표현되며, NHPP 모델에 근거한 지수형 신뢰도 성장 모델을 이용해서 소프트웨어 최적 배포시기를 구체적으로 수치를 이용해 제시한 연구 결과도 있다.[9][10]

3.2 시험능력의존형 신뢰도 성장 모델

이 모델에서는 시험단계에서 시험 공정에 투입된 시험능력을 고려하여 소프트웨어 신뢰성을 평가하는 모델이다. 시험에 투입되는 자원의 변화량과 소프트웨어 신뢰도와의 관계를 정하는 문제로 신뢰도 목표치에 이를 때까지 시간에 따른 시험자원의 투입량과 시험에서 검출되는 에러 검출율로 표시하며 이 모델은 개발 과정의 최종단계 소프트웨어 개발관리에 적합한 모델이다. 즉, 시험단계의 진보관리에 따른 시험 자원의 할당 문제와 소프트웨어 개발단계에서 운용단계로 이행하는데 필요한 최적 배포시기를 결정할 수 있는 대표적인 모델이다. 특히 이 모델에서는 시험 능력의 투입량은 개발시간에 대해 Rayleigh Curve를 따르는 것으로 알려져 있으며 시험능력 투입량 및 평균치 함수는 아래와 같이 표시된다.

$$\omega(t) = \alpha(1 - e^{-\beta t^m}),\ (\alpha > 0,\ \beta > 0,\ m > 0) \quad (10)$$

α : 총 시험 능력량, β : 시험능력 투입율

m : Parameter(m = 1 지수형, m = 2 Weibull형)

$$H(t) = a(1 - e^{-\gamma\omega(t)}) \quad (11)$$

$\omega(t)$: 시험 능력 함수, γ : 에러 발견율

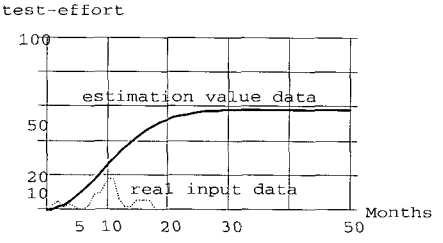
또 임의의 시간에 대한 시험능력의존형 모델의 소프트웨어 신뢰도는 앞의 식(9)에 평균치 함수인 식(11)을 대입하면

$$R(x|t) = \mathrm{Exp}[-a(\mathrm{Exp}[-\gamma\omega(t)] - \mathrm{Exp}[-\gamma\omega(t+x)])] \quad (12)$$

a : 시험전 소프트웨어내 잠재 에러수, x : 운용 시간

로 표시된다. 에러 발견율 γ는 시험의 질과 양을 나타내는 것으로 이 값이 1 보다 크면 시험의 효과가 매우 높고 1 일때는 지수형을 나타낸다. 일반적으로 실제 이 값은 0과 1사이의 값을 나타내고 있으며, 최종 배포

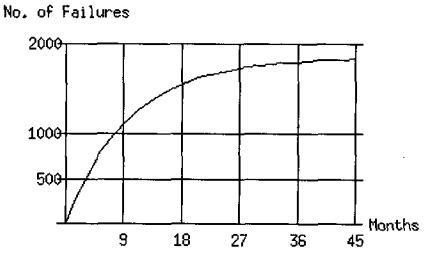
단계로 갈수록 점점 높아지는 경향을 띤다. 지수형 신뢰도 성장 모델에서는 b로 표시하기도 한다. 아래 (그림 1)은 시험에 투입된 시험 능력 예측치를 나타낸다.



(그림 1) 시험능력 투입량 예측치
(Fig. 1) Prediction of test-effort

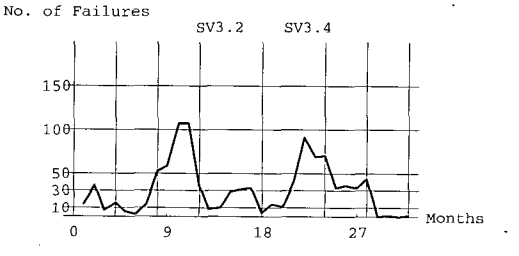
3.3 지수형과 시험능력의존형 모델 분석 결과

실제 소프트웨어 개발에 투입된 시험능력에 의해 검출된 결함들을 지수형 성장모델에 의거하여 살펴보면 (그림 2)와 같다.



(그림 2) 지수형 성장 모델 누적 고장수
(Fig. 2) Cumulative failures of Exponential Model

개발 초기에 적용이 용이한 지수형 성장 모델은 고장 발견과 해결이 동시에 이루어지는 것으로 가정하기 때문에 실제 문제 해결 방법과는 다소 차이가 있으나 프로젝트전체의 개발 진도를 예측해 보는 모델로 적합하기 때문에 초기에 많이 적용된다. 아래 (그림 3)은 소프트웨어를 시험하는 과정에서 검출된 고장들에 대한 월별 현황이다.

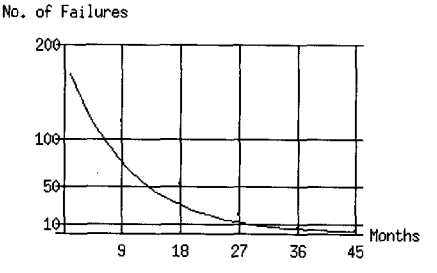


(그림 3) 월별 검출된 고장수
(Fig. 3) Detected failures per month

앞의 (그림 1)과 비교해 볼 때 시험 시작 10개월 전후와 22개월 후에 제일 많은 고장을 발견하고 있다. 즉, 이 시기에 시험능력을 최대로 투입하여 소프트웨어 내에 잠재하고 있는 에러들을 제거하고 있음을 알 수 있다.

1차 소프트웨어(SV3.2) 배포시기인 18개월과 2차 배포시기인 32개월 후에는 발견되는 고장이 적은 것으로 나타나 소프트웨어 배포 직전에 발견된 에러들을 수정함으로써 소프트웨어가 안정되어 가는 추세를 보인다. 특히 그림에서도 알 수 있듯이 시험시작 10개월 후와 22개월 후는 개발확인시험과 상용시제품시험이 병행된 시기여서 소프트웨어 에러 발견을 위해 시험능력이 집중된 시기다.

이와 같이 시험에서 검출된 고장데이터를 가지고 지수형 소프트웨어 신뢰도 성장 모델(E-SRGM : Exponential Software Reliability Growth Model)로 추정한 소프트웨어내의 잔존 에러수는 (그림 4)와 같은 추세를 띤다. 대표적인 신뢰도 성장 모델을 비교한 결과 시험능력의존형 모델은 S-Shaped Model과 유사한 특징을 보였고, 시험시작 12개월 후에 원하는 신뢰도 목표치(95%)에 도달하며, 소프트웨어 현장 배포시기인 18개월후에 97%, 36개월 후에는 98%를 상회하는 것으로 추정되었다.



(그림 4) 지수형 신뢰도 성장모델의 고장 강도 함수
(Fig. 4) Failure density of E-SRGM

시험시작 36개월 후 3.2항의 시험능력 의존형으로 소프트웨어 신뢰도를 추정하면 앞의 식 (12)에 식 (10), (11)에서 구한 추정치를 적용해 추정할 수가 있다. 이 때 적용되는 각 변수들에 대한 값은 <표 1>에 언급되었다.

〈표 1〉 지수형과 시험능력의존형 추정치
〈Table 1〉 Evaluation of values(Exponential and Test-effort dependant Model)

모델명	a	b	α	β	γ
지수형	1842	0.1015	-	-	-
시험능력의존	940.82	-	58.255	0.0057	0.0657

이때 소프트웨어 신뢰도 R(x|t)는

$$R(x|t=36)=\mathrm{Exp}[a(\mathrm{Exp}[-\gamma\omega(t)]-\mathrm{Exp}[-\gamma\omega(t+x)]]$$

로 되어 운용시간에 대해 추정한 소프트웨어 신뢰도는 아래 <표 2>와 같다

지수형 신뢰도 성장 모델에 의한 신뢰도 추정치는 식 (9)에 <표 1>의 지수형 모델 파라미터 추정치 a, b 값을 적용하면 된다. 이것을 다시 정리하면

$$R_m(x|t)=\mathrm{Exp}[-e^{-bt}m(x)] \quad (13)$$

m(x) : 운용시간(x)에 따른 평균치 함수값

으로 표현된다.

〈표 2〉 운용시간에 따른 신뢰도 추정치
〈Table 2〉 Estimation of Reliability with operation times

x(운용시간)	w(t+x)	R(x|t) (%)
0.1	58.2204	99.7981
0.5	58.2257	99.0881
1.0	58.2312	98.3569

즉, $t=36$개월 때 소프트웨어 신뢰도 R(x|t)는 62% 정도인 것으로 추정되었으며, 이때 운용시간에 따른 소프트웨어내의 잔존 에러수 및 순간에러 발생율(그림 4 참조)을 구하면 아래 <표 3>과 같다.

〈표 3〉 지수형 성장 모델 신뢰도 추정치
〈Table 3〉 Estimated values (Exponential Model)

t+x (t = 36, x)	잔존 에러수	순간 에러 발생율(개)
0.1	47	4.8
0.5	45	4.6
1.0	43	4.3

또 시험능력의존형 모델에 의한 소프트웨어내의 기대 잔존 에러수를 추정하면 시험능력의존형 모델의 평균치 함수

$$m_w(t)=a(1-e^{\gamma\omega(t)})$$

를 구해 추정할 수 있다. 이때 기대 잔존 에러수($n_{W(t)}$)는

$$n_{W(t)}=n_w(t)=m_w(\infty)-m_w(t)=a(e^{-\gamma\cdot\omega_R(t)}-e^{-\gamma\cdot\omega_R(\infty)})$$

로 표시되며,

이것을 추정하면 <표 4>와 같다.

〈표 4〉 시험능력의존형 모델 신뢰도 추정치
〈Table 4〉 Estimated values (Testing-effort Model)

운용시간(x)	잔존에러수	순간에러 발생율
0.1	20.5257	0.0487194
0.5	20.5186	0.0487363
1.0	20.5112	0.0487539

위와 같이 2개의 신뢰도 모델을 비교해 본 결과 시험 시작후 36개월 후의 소프트웨어 내에 잠재하고 있는 에러수는 시험능력의존형이 20개, 지수형이 46개 정도로 추정되었고 순간에러 발견율은 0.05개와 5개로 추정되었다. 기존 연구결과에서도 밝혀진 것과 같이 시험 단계의 교환시스템 개발에 적합한 모델은 지수형보다는 시험능력의존형 모델이 더 적합하다고 할 수 있다. 다시 말해서 개발 종료 단계인 시험 단계에서는 고장데이터를 고장발견 과정과 고장인지 과정으로 나누어 해석하는 S-Shaped(S-자형) 모델과 유사한 시험능력의존형 모델이 우리가 추진하고 있는 교환시스템 소프트웨어 개발 프로젝트에 더 적합하다는 것을 알았다.

4. 결 론

지금까지 잘 알려져 있고 널리 사용되는 소프트웨어 신뢰도 성장 모형의 대표적 모델로는 Goel-Okumoto (G-O) model, S-Shaped model, 그리고 Musa-Okumoto (M-O) model 들이다. 이 모델들은 각 모델의 특성에 따라 다소 차이가 있으나 개발 초기에는 예측 모델로 G-O 모델이 널리 이용되고 시스템 개발 최종 단계인 시험 단계에서는 S-Shaped 모델이 적합하다는 것을 이미 연구 발표한 바 있다.[11]

NHPP 모델에 근거한 소프트웨어 신뢰도 성장 모델을 이용하고 소프트웨어 개발관리의 응용으로서 소프트웨어 최적 배포문제는 개발자는 물론 사용자에게도 대단히 중요한 사항이다. 최적 배포문제는 소프트웨어의 시험 공정관리에 의한 진보관리 문제와 시험능력의 최적 배분에 의한 배포시기 결정 방법이 있다. 본 논문에서는 소프트웨어 평가 기준에 의한 소프트웨어 신뢰도 성장모델로부터 도출된 신뢰도 평가척도를 가지고

사용자가 원하는 소프트웨어 납기를 결정하고 소프트웨어 운용단계에 이르는 단계에서의 에러 발견사상에 대해 논했다.

향후 연구 방향은 총 기대되는 소프트웨어 비용 평가와 운용단계에 발견되는 에러 발견 사상 등을 포함한 소프트웨어 최적 배포시기를 결정하는 방법이다. 실제 소프트웨어 개발에 투입된 비용을 분석하여 생산성을 평가한 예[12]가 발표되었으나 매우 미흡한 상태로 관련된 많은 연구가 필요하다.

참 고 문 헌

[1] S. Yamada, "ソフトウェア信頼性評價技術 : ソフトウェア信頼度成長モデル入門", HBJ Publishing, Integrated libraries No.42, pp.195-206, 1989.

[2] B. W. Boehm, "Software Engineering Economics," Prentice-Hall, New Jersey. 1981.

[3] R. W. Wolverton, "The cost of developing large-scale software," IEEE Trans. Computers. Vol.C-23, No.6, pp.615-636, Jun. 1974.

[4] E. H.Forman and N. D. Singpuwalla, "Optimal time interval for testing hypotheses on computer software errors," IEEE Trans. Reliability, Vol.R-28, No.3, pp.250-253, Aug. 1979.

[5] H. S. Koch and P. Kubat, "Optimal release time of computer software," IEEE Trans. Software Eng., Vol.SE-9, No.3, pp.323-327, May 1983.

[6] K. Okumoto and A. L. Goel, "Optimal release time for software system based on reliability and cost criteria," J. System and Software, Vol.1, No.4, pp.315-318, 1980.

[7] S. Yamada and S. Osaki, "Optimal software release policies with simultaneous cost and reliability criteria," European J. Operational Research, Vol.31, No.1, pp.46-51, 1987.

[8] 大寺浩志, 山田 茂, 成久洋旨, "ソフトウェアの 運用段階におけるエラー發見事象を 考慮した最適リリース", 電子情報通信學會論文誌, Vol.J71-D, No.7, pp.1338-1340, Japan, 1988. 7.

[9] S. Yamada and S. Osaki, "Cost reliability optimal release polices for software system," IEEE Trans. Reliability, Vol.R-34, No.5, pp.422-424, Dec. 1985.

[10] A. Goel, "Software error detection model with applications," J. System and Software, Vol.1, pp. 243-249, 1989.

[11] 이재기 외 2, "고장데이터 분석을 통한 교환소프트웨어 특성 연구", 한국통신학회논문지, 제23권 8호, pp.1915-1925, 1998. 9.

[12] 신용연 외 2, "소프트웨어 개발비용 분석 및 생산성 평가", 제3회 통신 소프트웨어 학술대회 COMSW'98, 한국통신학회, pp.135-138, 1998. 7.

이 재 기

e-mail : jkilee@nice.etri.re.kr

1989년 청주대학교 대학원 전자공학과 졸업(공학석사)

1983년~1999년 현재 한국전자통신연구원 교환전송연구소 S/W종합검증팀 선임연구원

관심분야 : Software Test & Verification, 소프트웨어 신뢰도 등

신 상 권

e-mail : skshin@nice.etri.re.kr

1985년 중경공업전문대학 전자과 졸업

1988년~1999년 1월 현재 한국전자통신연구원 교환전송연구소 S/W종합검증팀

관심분야 : Software Integration & Test

홍 성 백

e-mail : sbhong@nice.etri.re.kr

1990년 연세대학교 대학원 전자공학과 졸업(공학석사)

1999년 1월 현재 한국전자통신연구원 교환전송연구소 S/W종합검증팀, 책임연구원

관심분야 : Software Integration & Test, Software Reliability