論文
4-4-14

SRGM을 이용한 소프트웨어 신뢰도 평가에 관한 연구
(A Study on Software Reliability Evaluation Using SRGM)

신 경 애(Kyung-Ae Shin)[1]

요 약

현재까지 연구된 SRGM을 이용하여 정해진 시각에 수집된 테스트 데이터를 가지고 예상되는 소프트웨어 고장이나 잔존 에러수를 추정할 수 있다. 그러므로 소프트웨어 신뢰성 달성 정도 및 운용 단계에서 소프트웨어 신뢰도를 예측할 수 있다. 그러나 어느 모델을 선택하는가에 따라 신뢰도 평가는 달라질 수 있다. 그러므로 본 연구에서는 에러제거 비용을 고려한 SRGM으로서 테스트 비용을 에러 검출 및 에러제거 비용까지도 고려한 SRGM을 제시하고자 한다. 또한 이를 이용하여 소프트웨어에 있는 잔존 에러수와 릴리즈 이후 신뢰도 값과 최적 릴리즈 시기를 추정하여 보다 더 정확한 신뢰성 평가를 할 수 있다.

ABSTRACT

Can presume number of software failure or remaining fault that is expected with test data that is collected by decided time using SRGM that is studied until present. Therefore, can forecast software reliability achievement degree and software reliability use step. But, reliability evaluation according to if choose any model can change. Therefore, we present SRGM that consider test cost to error detection and error delete cost as SRGM that consider error delete cost in this research. Using this SRGM, can presume number of remaining fault in software, reliability and optimal release time.

1. 서론

소프트웨어 신뢰성 평가에 대한 연구는 현재 정보화사회에서 소프트웨어 제품 개발 및 공정을 관리하는데 매우 중요한 일이다. 일반적으로 컴퓨터 시스템의 고장은 하드웨어적인 것 보다 소프트웨어적인 원인이 더 많이 있다[1]. 소프트웨어적인 원인을 찾고 개발하기 위하여 소프트웨어 공학 기술을 이용하여 신뢰성이 높은 소프트웨어를 개발하는 것이 중요한 부분이다. 이런 부분을 해결하기 위하여 대부분 테스트 단계를 거친다. 테스트는 소프트웨어 개발 과정에 적용할 수 있지만 가장 효율적으로 신뢰성을 평가할 수 있는 단계가 주로 마지막 단계에서 이루어진다. 테스트는 개발 과정의 진행 사항이나 개발의 결과를 정량적인 척도에 의하여 파악할 수 있어야한다. 테스트 기간에 신뢰성을 높이는 방안으로 Jelinski와 Moranad 가 처음 소프트웨어 신뢰도 성장 모델

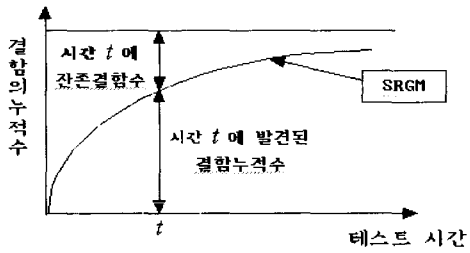
1) 종신회원 : 동주대학 컴퓨터정보통신계열 부교수
※본 연구는 2002학년도 동주대학 교내학술연구비 지원에 의하여 수행되었음.
논문심사 : 2003. 6. 23.
심사완료 : 2003. 7. 18.

(SRGM : Software Reliability Growth Model)을 소개한 이후로 여러 가지 SRGM이 연구 및 적용되고 있다. SRGM은 소프트웨어의 신뢰성의 달성 정도를 정량적으로 평가하는 수리모델이다. SRGM은 소프트웨어 테스트 단계에서 관측된 에러수의 시간적 추이를 소프트웨어가 성장해 오는 과정으로 하여 확률 및 통계론에 기초해서 기술하는 것이다. 즉, 테스트 단계에서 검출된 에러 데이터를 가지고 이를 SRGM에 적용하여 개발된 소프트웨어의 신뢰도를 정량적으로 예측하는 것이다. 따라서 소프트웨어 개발자는 릴리즈 이후의 소프트웨어 내에 잔존하는 에러 수, 에러 발견율, 평균 소프트웨어 고장 시간 간격 등의 예측이 가능하며, 한편 사용자에게는 소프트웨어 신뢰성을 확인하고 보증하게 된다. 본 연구에서는 테스트 단계에서 검출된 에러를 수정하는데 소요되는 비용을 테스트 비용에 포함시킨 신뢰도 성장 모델을 제시하여 신뢰성 평가에 대하여 살펴보고자 한다. 본 연구의 구성은 2장에서 SRGM모델과 NHPP모델에 대하여 고찰하고, 3장에서는 기존의 SRGM의 문제점을 분석하고 새로운 에러제거 비용을 고려한 SRGM을 제안하고, 또한 제안된 모델에 테스트된 데이타를 적용하여 잔존 에러수와 릴리즈 이후 신뢰도 값과 최적 릴리즈 시기를 구하고자 한다. 4장은 결론 및 향후과제 이다.

2. 관련연구

2.1 SRGM의 개념

소프트웨어 신뢰성 평가는 테스트 단계에서 관측된 데이터를 가지고 신뢰도 성장이라는 관점에서 논의 될 수 있다. 소프트웨어 신뢰도란 주어진 일정기간 주어진 환경아래에서 소프트웨어가 고장없이 주어진 명세서대로 정상적으로 운용될 수 있는 확률로 정의된다[3]. 소프트웨어 신뢰도는 테스트 시간이 경과함에 따라 점차적으로 신뢰도는 증가하며, 소프트웨어 고장 발생 시간은 점점 더 길어진다. 이것을 수리적으로 표현한 것이 SRGM이다. 일반적인 SRGM은 [그림 1]과 같다[2][4][5][6][7]. SRGM은 1972년에 Jelinski와 Moranad가 처음 소개한 이후로 Wolverton, Goel, Okumoto, Yamada등에 의해 연구 되고있다[6]. SRGM은 크게 시간계측모델, 개수계측모델, 가용시간모델, 경향곡선모델로 분류한다[5][6].



[그림 1] 소프트웨어 신뢰도 성장 모델
[Fig. 1] Software Reliability Growth Model

2.2 NHPP 모델

NHPP(NonHomogeneous possion process)모델은 개수계측 모델로서 소프트웨어 개발의 테스트 단계에서 발견되는 결함 수나 발생하는 고장수를 측정하여, 이것을 확률변수 $N(t)$에 적용하여 포아송 과정을 가정하는 SRGM이다[5]. 포아송 과정 중에서도 다음과 같은 4가지 성질을 만족할 때 $N(t)$는 NHPP에 따른다고 말한다[5][6][7].

① $N(0)=0$ 즉, 테스트 시작 시각 0 에서는 결함은 발견되지 않는다.
② $\{N(t), t\geq 0\}$ 는 독립 증분을 갖는다. 즉, 서로 다른 테스트 시간 구간에서 발견된 결함수는 통계적으로 독립이다.
③ $\Pr\{N(t+\Delta t)-N(t)\geq 2\}=0(\Delta t)$이다. 즉, 임의의 테스트 시간 구간 Δt에서는 2개 이상의 결함이 발견될 확률은 거의 없다.

④ $\Pr\{N(t+\Delta t)-N(t)=1\}=h(t)\Delta t+0(\Delta t)$이다. 즉, 임의의 테스트 시간 구간 Δt에서 1개의 결함이 발견될 확률은 발견율 및 소프트웨어 고장율에 비례한다.

이상과 같은 4개의 가정으로 확률변수 $N(t)$를 산출할 수 있는 NHPP는 식(1)과 같다.

$$\Pr\{N(t)=n\}=\frac{\{H(t)^n\}}{n!}e^{-H(t)} \quad (n=0,1,2,\cdots) \qquad \text{식(1)}$$
$$H(t)=\int_0^t h(x)dx \quad (t>0)$$

여기서, $H(t)$는 평균치 함수이며, $N(t)$의 평균치 즉, 테스트시각 t까지 발견되는 총기대 에러수 또는 발생할 총 기대 소프트웨어 고장수를 말한다. 또한 $h(t)$는 강도함수(intensity function) 즉, 테스트시각 t에서 순간 에러발견율 또는 소프트웨어 고장율을 나타낸다. 또한, 테스트 시작 전에 잔존하는 총 기대 에러수를 $a(a>0)$라고 할 때 $H(\infty)=a$로 된다. 그러므로 식(1)에 의해 $N(\infty)$의 확률분포($N(t)$의 극한분포)는 식(2)가 된다.

$$\lim_{t\to\infty}\Pr\{N(t)=n\}=\frac{a^n}{n!}e^{-a} \quad (n=0,1,2,\cdots) \text{식(2)}$$

식(2)에서 충분한 시간에 걸쳐서 테스트할 때 $N(t)$의 극한분포는 평균치가 a의 포아송 분포에 따른다는 것을 알 수 있다. 또한, 식(1)에서 정식화된 NHPP 모델에서 신뢰도 성장모델에 유용한 정량적 평가척도를 도출할 수 있으며, 테스트시각 t에서 소프트웨어 내의 기대 잔존 에러수는 확률변수 $\{N(\infty)-N(t)\}$의 기대치 고려로 식(3)가 된다.

$$n_r(t)=E[N(\infty)-N(t)]=a-H(t) \qquad \text{식(3)}$$

여기서, 테스트시각 t에서 에러가 발생한다는 조건이라면, 시간구간 $(t, t+x)$에서 에러가 발생하지 않을 확률은 식(4)가 된다. 식(4)를 소프트웨어 신뢰도라 한다.

$$R(x|t)=\exp\{-[H(t-x)-H(t)]\} \qquad \text{식(4)}$$
$$(t\geq 0, x\geq 0)$$

그리고, NHPP의 강도함수 $h(t)$의 식(3)이 정규화할 때 식(5)이 된다.

$$d(t)=\frac{h(t)}{[a-H(t)]} \qquad \text{식(5)}$$

여기서, $d(t)$ 시각 t의 잔존 에러 1개당의 에러 발견율을 나타낸다.

3. 모델 제안 및 평가

지금까지 연구된 SRGM을 이용해서 정해진 시각에 수집된 테스트 데이터로부터 예상되는 소프트웨어 고장 및 잔존 에러수를 추정하여 신뢰성의 달성 정도나 테스트 종료 후의 운용단계의 신뢰도를 예측할 수 있다. 그러나 여러 가지 다양한 모델들이 제시되어 있기 때문에 소프트웨어 개발자가 모델선택을 잘못할 경우에는 평가결과에 신뢰성이 떨어질 수 있다. 또한 모델을 적용하는데 제시하고 있는 가정이 소프트웨어 개발실제에 부적합한 경우도 있다[6]. 그러므로, 본 연구에서는 SRGM을 위해서 다음과 같이 가정한다.

① 임의의 시간 t까지 발견된 누적 에러수 {N(T), t≥0}는 NHPP(NonHomogeneous Poisson Process)를 따른다.
② 특정시간에 남아있는 에러는 소프트웨어 고장을 일으킬 수 있다.
③ 시간 (t,t+Δt)에서 검출된 평균 에러수는 테스트 비용과 평균 잔존 에러수에 비례한다.

④ 모든 검출된 에러는 제거할 수 있고 제거하는 과정에서 어떤 새로운 에러도 추가되지 않는다.
⑤ 테스트 비용은 웨이블 곡선으로 나타낼 수 있다[12].
⑥ 테스트 비용에는 에러를 검출하는 비용뿐 아니라 에러제거비용도 포함된다.

한편, 본 연구에서 사용되는 표기법은 다음과 같다.

- $w(t)$: 단위시간 t에서 소요되는 테스트 비용
- $W(t)=\int_0^t w(x)dx$: 테스트 종료 때까지의 테스트 비용(즉, 누적비용)
- C_B : 테스트 종료시간 T 전에 검출된 에러 하나를 수정하는데 소요되는 비용
- C_A : 테스트 종료시간 T 후의 검출된 에러 하나를 수정하는데 소요되는 비용
- C_T : 단위시간당 테스트 비용
- a : 테스트 전 총 에러수
- α : 테스트에 필요한 총비용
- β : 척도 파라메터
- m : 테스트 비용의 투입 형태를 나타내는 형상 파라메터
- γ : 테스트 단위 비용당 에러 검출율
- T_{LC} : 소프트웨어 라이프 사이클 전체시간
- T : 총 테스트 시간
- $C_B \cdot H(T)+C_A \cdot \{H(T_{LC})-H(T)\}+C_T \cdot W(t)$=총 테스트 비용($C(T)$) : 테스트 종료 전 검출된 에러 제거 비용 + 테스트 종료 후 검출된 에러제거 비용 + 테스트 종료 때까지의 테스트 비용

3.1 파라미터 값 추정

가정 ①과 ②에 의해서 검출된 에러의 누적수 즉, $\{N(t),\ t\geq 0\}$는 평균치 함수 $m(t)$를 가진 NHPP로서 식(6)과 같이 나타낼 수 있다[5].

$$\Pr\{N(t)=n\} = \frac{\{m(t)\}^n}{n!}e^{-m(t)} \quad (n=0,1,2,\cdots) \qquad \text{식(6)}$$

여기서, $m(t)$는 (0,t]구간에서 발견될 총 기대 에러수 및 검출 에러들의 평균수를 나타내며 식(7)과 같다.

$$m(t)=a(1-\exp[-rW(t)]) \qquad \text{식(7)}$$

여기서, 누적 비용 $W(t)$는 테스트 비용의 투입량으로 웨이블 곡선을 따르며[12], 식(8), 식(9)와 같이 나타낼 수 있다.

$$w(t) = \alpha\beta m t^{m-1}\exp[-\beta t^m] \quad (\alpha>0,\ \beta>0,\ m>0) \qquad \text{식(8)}$$

$$W(t)=\alpha(1-\exp[-\beta t^m]) \qquad \text{식(9)}$$

여기서, 파라미터 α, β, m은 측정 데이터 쌍 즉, (t_k, w_k) $(k=1,2,\cdots,n)$에 대해 최소자승법과 편차자승법에 의해 추정이 가능하며 식(10)과 같다.

$$\ln w(t) = \ln\alpha+\ln\beta+\ln mt(m-1)\ln t-\beta t^m$$

$$S(\alpha,\beta,m) = \sum_{k=1}^{n}\{\ln w_k-\ln w(t_k)\}^2 = \sum_{k=1}^{n}\{\ln w_k-\ln\alpha-\ln\beta-\ln m-(m-1)\ln t_k+\beta t_k^m\}^2 \qquad \text{식(10)}$$

여기서, 비용을 최소로 하는 추정치 $\hat{\alpha}$, $\hat{\beta}$, $\hat{m}$을 구할 수 있으며, 식(10)에서 식(11)을 구할 수 있다.

$$\frac{\partial S}{\partial\alpha}=\frac{\partial S}{\partial\beta}=\frac{\partial S}{\partial m}=0 \qquad \text{식(11)}$$

그리고 식(8)을 가지고 남아있는 에러의 평균수는 식(12)을 가지고 구할 수 있다.

$$a \cdot \exp[-rW(t)] \qquad \text{식(12)}$$

여기서, 파라미터 a, r은 측정 데이터 쌍 즉, (t_k, y_k) $(k=1,2,\cdots,n)$에 대해 최우법에 의해 추정이 가능하며 식(13)과 같다.

$$\begin{aligned} L &= P_r\{N(t_1)=y_1,\ N(t_2)=y_2,\ \cdots, N(t_n)=y_n\} \\ &= \exp[-m(t_n)] \prod_{k=1}^{n} \frac{\{m(t_k)-m(t_{k-1})\}(y_k-y_{k-1})}{(y_k-y_{k-1})!} \end{aligned} \qquad \text{식(13)}$$

여기서, $t_0=0$, $y_0=0$이다. 식(13)에서 대수 우도 함수는 식(14)와 같다.

$$\begin{aligned} \ln L &= \sum_{k=1}^{n} (y_k - y_{k-1}) \ln[m(t_k) - m(t_{k-1})] \\ &\quad - m(t_n) - \sum_{k=1}^{n} [(y_k - y_{k-1})!] \end{aligned} \qquad \text{식(14)}$$

그러므로, 테스트 비용을 고려한 SRGM의 대수 우도함수는 식(15)와 같다.

$$\begin{aligned} \ln L = {} & y_n \cdot \ln a + \sum_{k=1}^{n} (y_k - y_{k-1}) \ln[\exp[-rW(t_{k-1})] \\ & - \exp[-rW(t_k)]] - a(1-\exp[-rW(t_k)]) \\ & - \sum_{k=1}^{n} \ln[(y_k - y_{k-1})!] \end{aligned} \qquad \text{식(15)}$$

여기서, 식(15)를 정리하면 우도방정식이 된다. 즉, 식(16), (17)과 같다.

$$a = \frac{y_n}{(1-\exp[-rW(t_n)])} \qquad \text{식(16)}$$

$$\begin{aligned} & \frac{y_n W(t_n)\exp[-rW(t_n)]}{(1-\exp[-rW(t_n)])} \\ & - \sum_{k=1}^{n} \frac{(y_k - y_{k-1})(W(t_k)\exp[-rW(t_k)] - W(t_{k-1})\exp[-rW(t_{k-1})])}{(\exp[-rW(t_{k-1})] - \exp(-rW(t_k))} \end{aligned} \qquad \text{식(17)}$$

이들을 수치적으로 풀어서 파라미터의 값인 최우추정치 $\hat{a}$, $\hat{r}$을 구할 수 있다.

3.2 평균치 함수 추정

식(16), 식(17)를 이용하여 평균치 함수 값을 추정할 수 있다. 즉, 식(18)과 같다.

$$\begin{aligned} H(t) &= a(1-\exp[-rW(t)]) = a(1 \\ &\quad -\exp[-r\alpha(1-\exp[-\beta t^m])]) \end{aligned} \qquad \text{식(18)}$$

여기서, $H(t)$는 평균치 함수 값이다. 즉, 테스트 시각 t 까지 발견되는 총기대 에러수를 말한다.

3.3 기대 잔존 에러 수 추정

테스트 시각 t에 있어서 소프트웨어 내의 잔존하는 기대잔존 에러수를 추정할 수 있다. 즉, 식(19)와 같다.

$$\begin{aligned} n_r(t) &= E[N(\infty) - N(t)] = a - H(t) \\ &= a - a(1-\exp[-r\alpha(1-\exp[-\beta t^m])]) \\ &= a\exp[-r\alpha(1-\exp[-\beta t^m])] \end{aligned} \qquad \text{식(19)}$$

3.4 신뢰도 값 추정

소프트웨어 신뢰도는 식(20)에 의하여 추정할 수 있다.

$$\begin{aligned} R(x|t) &= \exp\{-[H(t-x) - H(t)]\} \\ &= \exp\{-[a(1-\exp[-r\alpha(1-\exp[-\beta(t-x)^m])]) \\ &\quad - a(1-\exp[-r\alpha(1-\exp[-\beta t^m])])\} \quad (t \geq 0, x \geq 0) \end{aligned} \qquad \text{식(20)}$$

3.5 최적 릴리스 시기

최적 릴리스 시기는 목표치 신뢰도와 소요된 총 테스트 비용 사이의 관계를 고려하여 구할 수 있다. 최적 릴리스 시기는 식(21)에 의해서 추정할 수 있다.

$$T_0 = \left[\frac{-1}{\beta} \ln\left\{1 - \frac{\ln[\alpha\gamma(C_A - C_B)/C_T]}{\gamma\alpha}\right\}\right]^{1/m} \qquad \text{식(21)}$$

〈표 1〉 Brooks와 Motley의 실험 데이터
〈Table 1〉 experimental data of Brooks & Motley

테스트 시간(t_k)	누적 에러수 (y_k)	테스트 노력량 (w_k)	테스트 시간(t_k)	누적 에러수 (y_k)	테스트 노력량 (w_k)
1	14	7.76	19	900	42.24
2	33	3.10	20	977	40.17
3	58	6.90	21	1030	96.55
4	107	7.24	22	1090	88379
5	137	6.90	23	1116	56355
6	160	12.24	24	1153	79.31
7	181	19.83	25	1161	73.97
8	225	52.41	26	1186	65.00
9	275	46.90	27	1219	67.24
10	312	95.69	28	1240	117.24
11	361	55.35	29	1247	88.28
12	405	59.31	30	1277	77.93
13	479	43.28	31	1279	37.93
14	558	44.48	32	1283	40.52
15	618	41.90	33	1286	54.31
16	674	75.35	34	1293	63.10
17	754	63.62	35	1300	39.48
18	839	73.97			

4. 실제 데이터 적용 결과

4.1 파라미터 추정 값

실제 테스트에서 관측된 데이터를 가지고 본 연구에서 다룬 테스트 비용을 고려한 SRGM에 적용한다. 테스트 데이터는 Brooks와 Motley의 실험결과에서 얻은 측정 데이터로 〈표 1〉과 같다[12]. 〈표 1〉를 가지고 식(10)에 의해서 $\hat{a}, \hat{\beta}, \hat{m}$를 구해보면 $\hat{a}=2253.0$, $\hat{\beta}=0.0004499$, $\hat{m}\simeq 2.25700$이다. 또한 식(16), (17)에 의해 $\hat{a}$, $\hat{r}$을 구하면 $\hat{a}=1394.1$, $\hat{r}=0.0015934363$이다. 그러므로 추정된 평균치 함수는 식(18)에 의해서 구하면 다음과 같다.

$$\begin{aligned} H(t) &= a(1-\exp[-rW(t)]) \\ &= a(1-\exp[-r\alpha(1-\exp[-\beta t^m])]) \\ &= 1394.1(1-\exp[-0.0015934363\times 2253.0 \\ &\quad (1-\exp[-0.0004499\times t^{2.25700}])] \end{aligned}$$

4.2 기대 잔존 에러수

테스트 시간의 경과에 따른 기대잔존 에러수는 식(19)를 이용하여 구하면 다음과 같다.

$$\begin{aligned} \hat{r}(t) &= \hat{a}-H(t) = a-a(1-\exp[-r\alpha(1-\exp[-\beta t^m])]) \\ &= a\exp[-r\alpha(1-\exp[-\beta t^m])] \\ &= 1394.1\times\exp[-0.0015934363\times 2253.0 \\ &\quad (1-\exp[-0.0004499\times t^{2.25700}])] \end{aligned}$$

여기서, 기대 잔존 에러 수를 시각에 따라 구하면 〈표 2〉와 같다. 〈표 2〉를 살펴보면 테스트를 하는 시각이 길어질수록 잔존하는 에러 수가 점점 더 줄어든다는 것을 알 수 있다. 즉, 테스트 기간 중에 에러가 검출됨에 따라 소프트웨어 내에 잔존하는 에러 수는 감소한다는 것을 의미한다.

4.3 소프트웨어 신뢰도

테스트 시각의 경과에 따른 소프트웨어 신뢰도는 식(20)을 이용하여 구하면 다음과 같다.

〈표 2〉기대 잔존 에러수
〈Table 2〉 remaining fault count

시간	기대잔존에러수	시간	기대잔존에러수
1	1391.8507	19	487.6474
2	1383.3900	20	438.7621
3	1367.5518	21	393.9780
4	1343.8453	22	353.2578
5	1312.1948	23	353.2578
6	1272.8657	24	283.4547
7	1226.4159	25	253.9539
8	1173.6489	26	227.7168
9	1115.5614	27	204.4700
10	1053.2858	28	183.9375
11	988.0295	29	165.8489
12	921.0154	30	149.9461
13	853.4277	31	135.9872
14	786.3641	32	123.7490
15	720.8000	33	113.0282
16	657.5624	34	103.6413
17	597.3171	35	95.4246
18	540.5647		

$$R(x|t) = \exp\{-[H(t+x)-H(t)]\}$$
$$= \exp(-(1394.1*(1-\exp(-0.0015934363*2253.0*(1-\exp(-0.0004499*(t+x)^m)))))$$
$$1394.1*(1-\exp(-0.0015934363*2253.0*(1-\exp(-0.0004499*t^m)))))$$

여기서, 소프트웨어 신뢰도를 구하면 〈표 3〉과 같다. 〈표 3〉을 살펴보면 테스트를 하는 시간이 길어질수록 잔존하는 에러 수가 점점 더 줄어들며, 이것은 테스트 시간이 경과함에 따라 신뢰도 함수 값이 점차적으로 증가한다는 것을 보여준다. 즉, 소프트웨어 내에 잔존하는 에러가 발견되어 잔존하는 에러 수는 점점 줄어들어 소프트웨어의 신뢰도는 증가함을 보여준다.

4.4. 최적 릴리스 시기

최적 릴리스 시기는 목표치 신뢰도와 소요된 총 테스트 비용 사이의 관계를 고려함으로써 얻어질 수 있다. 테스트 기간에 검출된 에러 하나를 제거하기 위해 필요한 비용 C_B에 대해, 검출된 에러제거를 위한 평균 비용은 $C_B \cdot H(t)$로 나타내며, 테스트 단위 시간당 비용 C_T에 대해서 테스트 비용은 $C_T \cdot W(t)$로 구할 수 있다.

〈표 3〉 소프트웨어 신뢰도값
〈Table 3〉 Reliability values of Software

테스트 시간	신뢰도 함수값	테스트 시간	신뢰도 함수값
1	0.950247	19	0.601081
2	0.886354	20	0.626153
3	0.820631	21	0.652348
4	0.757782	22	0.679068
5	0.700484	23	0.705765
6	0.650247	24	0.731956
7	0.607794	25	0.757238
8	0.573308	26	0.781290
9	0.546624	27	0.803882
10	0.527370	28	0.824860
11	0.515065	29	0.844146
12	0.509180	30	0.861722
13	0.509173	31	0.877618
14	0.514494	32	0.891903
15	0.524590	33	0.904672
16	0.538895	34	0.916034
17	0.556820	35	0.926108
18	0.577758		

그리고, 총 테스트 비용 $C(T)$는 3장에서 제시한 대로 $C(T) = C_B \cdot H(T) + C_A \cdot \{H(T_{LC}) - H(T)\}$ $C_T \cdot W(t)$이다. 또한, C_B와 C_T의 관계를 고려하면 식(22)과 같이 나타낼 수 있다.

$$C_B + C_T = \frac{T}{H(T)}, \quad C_A = \frac{T_{LC} - T}{H(T_{LC}) - H(T)} \quad \text{식(22)}$$

식(22)을 이용하여 C_B, C_T, C_A를 구하면, $C_B = 0.4038$, $C_T = 0.4038$, $C_A = 3.9769$ 이다. 이 값을 식(21)에 적용하여 최적 테스트 종료 시기를 구하면 $T^* = 39.275$이다. 이 값은 에러 제거 비용을 고려한 값이다. 즉, 기존의 연구에서는 테스트 시 에러 검출 비용만 가정했을 경우에 39.6과 본 연구에서 가정한 에러 제거 비용을 고려한 것과 약 10일간의 차이가 난다는 것을 알 수 있다. 그러므로 사용자에게 소프트웨어를 인도할 시점에서 비용 추정을 할 때 본 연구에서 제시한 방법을 이용한다면 보다 더 정확한 비용을 예측할 수 있을 것이며, 소프트웨어 개발자에게는 더욱 더 현실성 있는 개발 계획을 세울 수 있을 것이라 생각된다.

5. 결론 및 향후 연구 과제

소프트웨어를 개발할 때 중요한 문제는 테스트를 언제 중단하고 소프트웨어 신뢰성이 있는 양질의 소프트웨어를 언제 사용자에게 인도할 것인가 하는 것이다. 테스트 시간이 길면 길수록 신뢰도는 향상되지만 그 만큼 비용은 증가한다. 그러므로 최적 릴리즈 시기를 찾아서 릴리즈 한다면 소프트웨어 신뢰도와 비용도 최적의 상태가 될 수 있다. 최적 릴리즈는 정확한 비용산정에 의하여 구할 수 있다. 그러므로 테스트비용으로 에러 검출 비용뿐만 아니라 에러제거 비용도 고려하여 최적릴리즈 시기를 구할 수 있다. 본 연구에서는 에러 검출 및 에러제거 비용까지도 고려한 SRGM을 제시하고, 또한 이를 이용하여 소프트웨어에 있는 잔존 에러수와 릴리즈 이후 신뢰도 값과 최적 릴리즈 시기를 추정하여 보다 더 정확한 신뢰성 평가를 하였다. 향후 연구 과제는 실제 테스트 시 검출된 에러 각각을 제거하는데 소요하는 비용과 비교 분석과정을 통해 추정치와 실제 값의 근사 정도를 확인하는 것이다.

※ 참고문헌

[1] 신경애, "NVP신뢰도 분석을 위한 유전자 알고리즘 적용", 한국컴퓨터산업교육학회 논문지, "2000, 10, Vol. 1., No. 1, October.

[2] 최신형, " 에러 제거비용을 고려한 소프트웨어 신뢰도 성장모델 ", 경남대학교 대학원 석사 학위논문, 1994. 12.

[3] 고건외, " 소프트웨어 신뢰도에 관한 연구 ", 정보과학회 가을학술집, PP286~291, 1981.

[4] P.Zeephongsekul, G.Xia, S.Kumar : "Software-Reliability Growth Model : Failures Generate Secondary-Faults Under Imperfect Debugging",IEEE Trans Reliability, Vol.43, No.3, 1994.

[5] 山田 茂, 高橋宗雄 著 : "ソフトウェアマネジメソトモデル 入門", 共立出版株式會社, 1993.

[6] 山田 茂 : "ソフトウェアの信賴性評價技術", HBJ出版局, 1989.

[7] S.Yamada, J.Hishitani and S.Osaki : "Software Reliability Growth with a Weibull Test-Effort : A Model & Application", IEEE Trans., Reliability, Vol. 42, NO.1, 1993.

[8] G.J.Schick and R.W.Wolverton : "An analysis of competing Software reliability models", IEEE Trans., Software Engineering, Vol.SE-4, No.2, 1978.

[9] A.L.Goel and K.Okumoto : "A Markovian model for reliability and other performance measures of software systems", Proc. National Computer Conf., 1979.

[10] Capers Jones : "Applied Software Measurement : Assuring Productivity Quality", 1991.

[11] J.D.Musa and A.Iannino and K.Okumoto : " Software Reliability : Measurement, Prediction, Application", McGraw-Hill, 1987.

[12] H.Ohtera, S.Yamada, "Optimal allocation and control problems for Software-Testing resources, "IEEE Trans. Reliability, Vol.39, No.2, 1990.

신 경 애

1986 한국방송대학교 경영학과 졸업(경영학사)

1991 한국방송대학교 전자계산학과 졸업(이학사)

1989 계명대학교 교육대학원 전자계산전공(교육학석사)

1999 경남대학교 대학원 컴퓨터공학과(공학박사)

1983~1989 동주여자상업고등학교 재직

1990~현재 동주대학 컴퓨터정보계열 부교수

관심분야 : 소프트웨어 비용예측, 결함허용, 신뢰도 분석, 소프트웨어 신뢰도 성장모델, CAI