韓國軍事運營分析學會誌
第16卷, 第2號, 1990. 12. 31.

결함유형에 따른 소프트웨어 신뢰도와 소프트웨어 상품화 최적 시기 전략

A Cost-Reliability Model for the Optimal Release Time of a Software System

金永輝 · 李杭炯*

Abstract

This paper classifies faults into three types : simple, degenerated, and regenerated faults. This paper also deals with the characteristics of each type of fault to determine the software reliability based on the assumption; i.e., a system consisting of several subsystems (modules) which may be debugged simultaneously.

For each type of fault, several formulas are developed to obtain the failure rate and the expected number of failures found during debugging.

A model is developed based on the formulas of the failure rate and the expected number of failures to decide the optimal release time of a new software; minimizing the total cost with constraints restricting to the failure rate of each module in the software. By using this model, optimal release times are found for some cases; the eliminated faults are assumed simple faults only, regenerated faults only, simple and degenerated faults, and so on.

* 고려대학교

1. 서 언

현대사회에서의 제반문제는 복잡하고 유기적으로 연결되어 있기 때문에 이를 해결하기 위해서는 신속한 정보의 교환과 처리가 요구된다. 정보체계의 수요증가는 컴퓨터 hardware 부문의 급진적인 발전의 요인이 되었으며 최근 20년 동안에 hardware 개발비용은 1/1000 정도로 감소되었으나 정보처리 성능은 200배 정도의 향상에 그치고 있다. 이는 적은 비용으로 고성능의 hardware체계를 개발할 수 있는 반면 software technology의 부진으로 software 체계의 개발 및 보전에 막대한 비용이 소요됨을 보여주는 좋은 예이다. 본 연구에서는 software 체계 내에 존재하는 fault의 유형을 식별하고 각 유형별 fault가 software의 reliability와 failure rate에 미치는 영향을 이해함으로써 최적의 software release 시기에 대한 전략을 도출코자 한다.

2. 가정 및 용어 설명

Myers[10]는 fault와 failure에 관하여 기술하였다. 그는 Software가 계획된 사양(specification)에 따라 실행되지 않을 때 failure가 발생된다고 보았으며 이때 software failure를 발생시키는 요인을 software의 fault로 정의하였다. Myers[10]는 software의 fault를 data type data declaration, interface, I/O, computation, comparison, control flow로 분류하였다. Shooman[13]은 major fault와 minor fault로 구분하였으며 major fault에는 infinite loop, system crash, memory overflow, database corruption 등을 포함시켰고 minor fault에는 잘못된 입출력, 부정확한 계산 등에 의한 부분적인 기능마비를 포함시키고 있다.

Software reliability와 관련되는 fault는 3가지 유형을 고려할 수 있다. 어떤 fault가 수정되거나 제거될 때 system 내에 존재하는 다른 어느 fault에 영향을 미치지 않는 경우, 어떤 fault가 수정되거나 제거될 때 동일한 속성을 가진 fault가 자동적으로 제거되는 경우, 그리고 마지막으로 어떤 fault가 수정되거나 제거될 때 새로운 fault를 유발시키는 경우이다.

위의 3가지 유형의 fault들을 각각 simple fault, degenerated fault, regenerated fault로 정의하며 이하 S-type, D-type, R-type으로 표기하자. S-type은 가장 보편적으로 일어나는 fault의 경우이며, D-type은 DO-LOOP 등 실행이 반복되는 program에서 하나의 fault가 발견 수정되면 반복회수 만큼의 fault가 수정되는 경우이고, R-type은 program 작성과정에서 error list를 보고 error를 수정하여 다시 execution 시켜보면 이번에는 전혀 예상하지 않은 부분에서 새로운 error가 발생됨과 같은 경우이다.

본 연구에서는 개발된 software를 얼마동안의 test 기간을 거쳐 software를 release 시킬 것인

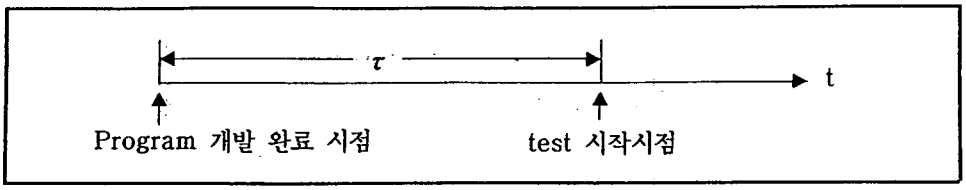


그림 1-1 test design

가를 결정하는데 주안이 있는데 test 경우에 있어서도 software가 개발된 직후부터 test하는 것이 아니고 개발된 후로 부터 τ시간이 경과한 후부터 test하는 경우를 모델화 했다. 경과시간 τ를 허용한 것은 보다 일반적인 모형을 의미한다. 왜냐하면 τ를 허용할 필요가 없을 경우는 $\tau=0$로 놓으면 되기 때문이다. 또한 경과시간 τ동안 발생된 fault의 유형을 식별하고 그 식별된 유형에 따라 test기간에 적용될 failure rate distribution이 달라지기 때문이다. test design을 요약하면 〈그림 1-1〉과 같다.

본 연구에서 사용된 가정들은 다음과 같다.

① Program이 개발완료된 시점에서의 fault 수는 N개이다.

② 경과시간 τ까지 발견되고 수정된 fault 수는 X개이다.

③ Fault는 detect된 후에 식별 가능하며 즉시 수정된다.

④ Fault의 occurrence rate(Φ는 gamma(α, β)이다.

⑤ Program내에 k개의 fault가 존재할때 program의 failure rate는 $\Lambda=\Phi_1+\Phi_2+\cdots+\Phi_k$이다.

⑥ Program이 고장날때까지의 시간은 exp(Λ)를 따른다.

또한 자주 사용될 용어들을 정의하면 다음과 같다.

Λ : Program의 failure rate

λ : 확률변수 Λ의 realization

Φ : Fault의 occurrence rate

ϕ : 확률변수 Φ의 realization

$R_m(t)$: 시간 t에서 module의 reliability

$R(t)$: 시간 t에서 system의 reliability

$E\{N(T)\}$: test 시작으로 부터 T시간이 경과한 후의 평균 failure 수

3. Module의 Reliability와 Failure Rate

3.1 S-type의 경우

시험을 시작한후 t시간이 경과한 후의 module의 reliability와 failure rate에 대해서는 Littlewood[6, 7, 8]에 의해 연구되었다.

Littlewood에 의하면

$$R_m(t) = \left(\frac{\beta+\tau}{\beta+\tau+t}\right)^{(N-X)\alpha} \qquad \cdots\cdots\cdots\cdots (3.1)$$

$$\lambda(t) = \frac{(N-X)\alpha}{\beta+\tau+t} \quad \cdots\cdots\cdots\cdots (3.2)$$

이다.

3.2 D-type의 경우

τ가 경과하기 까지 Z개의 fault가 발견되고 매 fault 수정시 d개씩 동시에 수정되었다면 test가 시작될때 module에 남아있는 fault 수는 N-Xd 개이다. 여기서 잔존 fault에 의한 발생률(Φ)에 대한 확률 및 도함수(이하 pdf)은 다음과 같다.

$$f\{\Phi|\text{No failure caused by this fault } (0,\tau)\}$$
$$= C\ \Pr\{\text{No failure caused by this fault } (0,\tau) \mid \Phi=\phi\} f(\phi) \quad \cdots\cdots\cdots\cdots (3.3)$$

여기서 failure간의 interarival time은 exponetial distribution이므로

$$\Pr\{\text{No failure caused by this fault } (0,\tau) \mid \Phi=\phi\} = e^{-\phi\tau} \quad \cdots\cdots (3.4)$$

이다.

또한 occurrence rate는 gamma(α, β)이므로

$$f(\phi) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} \phi^{\alpha-1} \cdot e^{-\beta\phi} \quad \cdots\cdots (3.5)$$

이다.

식 (3.4)와 (3.5)를 식(3.3)에 대입하고 정리하면

$$f\{\phi|\text{No failure caused by this fault } (0,\tau)\}$$
$$= \frac{(\beta+\tau)^{\alpha}}{\Gamma(\alpha)} \phi^{\alpha-1} \cdot e^{-(\beta+\tau)\phi} \quad \cdots\cdots (3.6)$$

이 된다.

따라서 module 내 잔존해 있는 fault의 발생율은 gamma$(\alpha, \beta+\tau)$이다. 한편 N-Xd개의 fault가 잔존해 있고 각각의 발생율은 gamma$(\alpha, \beta+\tau)$이므로 module failure rate는 N-Xd개의 iid gamma$(\alpha, \beta+\tau)$의 합이므로

$f(\lambda) \sim$ gamma$\{(N-Xd)\alpha, \beta+\tau)$이다. 즉

$$f(\lambda) = \frac{(\beta+\tau)^{(N-Xd)\alpha}}{\Gamma[(N-Xd)\alpha)]} \lambda^{(N-Xd)\alpha-1} \cdot e^{-(\beta+\tau)\lambda} \cdots (3.7)$$

이다.

module의 reliability는 module의 time to failure에 대한 pdf를 구하면 쉽게 구할 수 있으므로 먼저 module의 time to failure에 대한 pdf는 아래와 같다.

$$f(t) = \int_0^{\infty} f(t|\Lambda=\lambda)\, f(\lambda)\, d\lambda$$
$$= \int_0^{\infty} \lambda e^{-\lambda t} f(\lambda)\, d\lambda \quad \cdots\cdots (3.8)$$

식(3.7)를 식(3.8)에 대입하여 적분하면

$$f(t) = \frac{(\beta+\tau)^{(N-Xd)\alpha}}{\Gamma[(N-Xd)\alpha]} \int_0^{\infty} \lambda^{(N-Xd)\alpha} \cdot e^{-(\beta+\tau+t)\lambda} d\lambda$$
$$= \frac{(N-Xd)\alpha \cdot (\beta+\tau)^{(N-Xd)\alpha}}{(\beta+\tau+t)^{(N-Xd)\alpha+1}} \quad \cdots\cdots (3.9)$$

이 된다.

module의 reliability는 t시간 이후의 probability의 합이므로

$R_m(t) = \int_t^{\infty} f(u)\, du$에 의해 구해진다.

따라서 식(3.9)를 대입하여 적분하면

$$R_m(t) = (N-Xd)\alpha(\beta+\tau)^{(N-Xd)\alpha} \cdot \int_t^{\infty}(\beta+\tau+u)^{-[(N-Xd)\alpha+1]}du \quad \cdots\cdots (3.10)$$

위 식을 적분하여 간소화 하면 아래와 같이 된다.

$$R_m(t) = \left(\frac{\beta+\tau}{\beta+\tau+t}\right)^{(N-Xd)\alpha} \quad \cdots\cdots\cdots\cdots (3.11)$$

한편 $\lambda(t) = \frac{f(t)}{R_m(t)}$ 이므로 식(3.9)와 식(3.11)로부터 $\lambda(t)$를 쉽게 구할 수 있다.

$$\lambda(t) = \frac{(N-Xd)\alpha(\beta+\tau)^{(N-Xd)\alpha} \cdot (\beta+\tau+t)^{-[(N-Xd)\alpha+1]}}{(\beta+\tau)^{(N-Xd)\alpha} \cdot (\beta+\tau+t)^{-(N-Xd)\alpha}}$$

$$= \frac{(N-Xd)\alpha}{(\beta+\tau+t)} \quad \cdots\cdots\cdots (3.12)$$

3.3 S-type과 D-type의 혼합형인 경우

발견된 X개의 fault중 j개가 S-type이라면 X-j개는 D-type이다. 따라서 test 시작 시점에서의 fault수는 N-j-d(X-j)개 이고 각각의 occurrence rate는 gamma$(\alpha, \beta+\tau)$이므로 3.2절에서 적용된 절차를 통해서 module의 reliability와 failure rate를 구할 수 있다.

$$R_m(t) = \left(\frac{\beta+\tau}{\beta+\tau+t}\right)^{\{N-j-d(X-j)\}\alpha} \quad \cdots\cdots\cdots\cdots (3.13)$$

$$\lambda(t) = \frac{\{N-j-d(X-j)\}\alpha}{\beta+\tau+t} \quad \cdots\cdots\cdots\cdots (3.14)$$

3.4 R-type의 경우

R-type과 이후에 논의될 R-type과의 혼합형의 경우 각 fault의 생성된 시점이 다르기 때문에 test 시작시점에서 잔존해 있는 각 fault의 occurrence rate의 distribution은 iid가 아니므로 occurrence rate의 합으로 이루어지는 module의 failure rate distribution를 구하고자 여러가지 시도를 했으나 실패했다.

따라서 위 2가지 경우에 대하여 각 fault의 occurrence rate는 exponential(β)로 가정하여 접근한다. 시간 Zero에서 N개의 fault가 있었고 시간간격 $(0, \tau)$사이에 X개의 fault가 발견되고, 발견된 모든 fault가 R-type이었으며 각 fault 제거시 마다 r개의 fault가 생성된다면 test시점에 잔존해 있는 fault의 숫자는 N+X(r-1)이 된다. 따라서 앞에서 적용되었던 유도절차를 통해서 test 시간후 t시간 후의 module의 reliability와 module의 failure rate를 구할 수 있다.

$$R_m(t) = \left(\frac{\beta}{\beta+t}\right)^{N+X(r-1)} \quad \cdots\cdots\cdots\cdots (3.15)$$

$$\lambda(t) = \frac{N+X(r-1)}{\beta+t} \quad \cdots\cdots\cdots\cdots (3.16)$$

3.5 S-type, D-type, R-type의 혼합형인 경우

τ가 경과하는 동안 X개의 fault중 S-type이 X_1개, D-type이 X_2개, 그리고 R-type X_3개라 하자. 즉 $X=X_1+X_2+X_3$ 여기서 D-type은 d개씩 줄고 R-type은 r개씩 늘어난다고 가정하면 test 시작시점에서 잔존해 있는 fault 수는 $N-X_1$

$-dX_2+rX_3$가 된다.

앞에서 적용되었던 유도절차를 통해서 $R_m(t)$와 $\lambda(t)$를 구할 수 있다.

$$R_m(t) = \left(\frac{\beta}{\beta+t}\right)^{N-X_1-dX_2+rX_3} \quad \cdots\cdots\cdots\cdots (3.17)$$

$$\lambda(t) = \frac{N-X_1-dX_2+rX_3}{\beta+t} \quad \cdots\cdots\cdots\cdots (3.18)$$

4. System의 Reliability

지금까지는 N개의 fault를 가지고 있는 module reliability와 failure rate에 대해서 알아보았다. 그러나 본 연구에서 관심있는 것은 m개의 module로 구성된 system이기 때문에 앞에서 구해진 module의 reliability $R_m(t)$와 failure rate $\lambda(t)$를 통해서 system의 reliability $R(t)$와 failure rate $\lambda(t)$를 유도하고자 한다.

4.1 S-type의 경우

m개의 module중 하나라도 failure가 발생되면 system의 failure가 발생되므로 $Y_1=\min(T_1, T_2, \cdots, T_m)$으로 표현된다.

여기서 Y_1 : time to failure for a system

T_i : time to failure for a module i

따라서 $R(t)=Pr(Y_1>t)$

$$=Pr(T_1>t,\ T_2>t, \cdots, T_m>t) \quad \cdots (4.1)$$

각각의 module은 independent이므로

$$R(t)=Pr(T_1>t)\ Pr(T_2>t), \cdots, Pr(T_m>t)$$

$$=R_1(t)R_2(t), \cdots, R_m(t) \quad \cdots\cdots (4.2)$$

이다.

따라서 식(3.1)로 부터

$$R(t) = \left(\frac{\beta+\tau}{\beta+\tau+t}\right)^{\Sigma(N_i-X_i)\alpha_i} \quad \cdots\cdots\cdots\cdots (4.3)$$

이 된다.

여기서 Σ은 $\sum_{i=1}^{m}$을 의미한다.

system의 time to failure에 대한 pdf는

$f(t)=\frac{dF(t)}{dt}$에 의해 구할 수 있는데 $F(t)=1-R(t)$이므로

$$f(t) = \frac{d[1-R(t)]}{dt}$$

$$= -\frac{dR(t)}{dt} \quad \cdots\cdots (4.4)$$

이다.

식(4.3)을 식(4.4)에 대입하여 정리하면 아래와 같이된다.

$$f(t)=\Sigma(N_i-X_i)\alpha_i(\beta+\tau)^{\Sigma(N_i-X_i)\alpha_i} \cdot (\beta+\tau+t)^{-\Sigma(N_i-X_i)\alpha_i-1} \quad \cdots\cdots (4.5)$$

한편 $\lambda(t)=\frac{f(t)}{R(t)}$ 이므로 식(4.5)를 식(4.3)으로 나누어 간략히 정리하면

$$\lambda(t) = \frac{\Sigma(N_i-X_i)\alpha_i}{(\beta+\tau+t)} \quad \cdots\cdots (4.6)$$

이다.

4.2 D-type의 경우

이 경우 module i의 reliability 즉 $R_i(t)$는 식 (3.11)에 의해

$$R_i(t) = \left(\frac{\beta+\tau}{\beta+\tau+t}\right)^{(N_i-X_id_i)\alpha_i}$$

이므로 system reliability는

$$R(t) = \left(\frac{\beta+\tau}{\beta+\tau+t}\right)^{\Sigma(N_i-X_id_i)\alpha_i} \quad \cdots\cdots\cdots\cdots (4.7)$$

이다.

또한 R(t)를 적분하여 f(t)를 구한다음 f(t)를 R(t)로 나누면 $\lambda(t)$를 구할 수 있다.

$$\lambda(t) = \frac{\Sigma(N_1-X_1d_1)\alpha_1}{(\beta+\tau+t)} \quad \cdots\cdots (4.8)$$

4.3 S-type과 R-type의 혼합형인 경우

식(3.13)과 식(3.14)로 부터 아래와 같이 R(t)와 $\lambda(t)$를 구할 수 있다.

$$R(t) = \left(\frac{\beta+\tau}{\beta+\tau+t}\right)^{\Sigma\{(N_i-J_i-d_i(X_i-J_i)\}\alpha_i} \cdots\cdots\cdots\cdots (4.9)$$

$$\lambda(t) = \frac{\Sigma(N_1-J_1-d_1(X_1-J_1))\alpha_1}{(\beta+\tau+t)} \quad \cdots\cdots (4.10)$$

4.4 R-type의 경우

이 경우 식(3.15)에 의해 module i의 reliability는

$$R_i(t) = \left(\frac{\beta}{\beta+t}\right)^{(N_i+X_i(r_i-1))}$$

이므로 R(t)와 $\lambda(t)$은 아래와 같이 된다.

$$R(t) = \left(\frac{\beta}{\beta+t}\right)^{\Sigma(N_i+X_i(r_i-1))} \quad \cdots\cdots (4.11)$$

$$\lambda(t) = \frac{\Sigma(N_i+X_i(r_i-1))}{(\beta+t)} \quad \cdots\cdots (4.12)$$

4.5 S-type, D-type, R-type의 혼합형인 경우

식(3.17)과 앞에서 적용되었던 절차를 통해서 R(t)와 $\lambda(t)$는 아래와 같이된다.

$$R(t) = \left(\frac{\beta}{\beta+t}\right)^{\Sigma(N_i-X_{1i}-dX_{2i}+rX_{3i})} \quad \cdots\cdots (4.13)$$

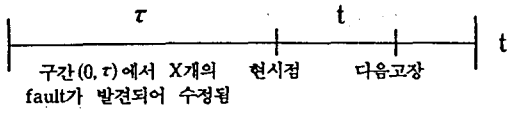
$$\lambda(t) = \frac{\Sigma(N_1-X_{1i}-dX_{2i}+rX_{3i})}{(\beta+t)} \quad \cdots\cdots (4.14)$$

5. T시간 경과후의 Failure Rate와 Average Failure 수

Optimal release time을 결정하는데 있어서 필수적인 요소는 현시점 τ로 부터 T시간이 경과한 후의 failure rate와 T시간이 경과하기 까지의 평균 failure 수이다.

5.1 S-type, D-type, S-type과 D-type의 혼합형인 경우

먼저 각 module의 failure rate와 평균 failure수에 대해서 알아보고 m개 module로 구성된 system의 failure rate와 평균 failure수에 대해서 유도코자 한다.



만약 경과시간 $T(\tau, \tau+T)$ 동안에 k개의 fault가 발견되어 수정되었다면 시간 $\tau+T$에서의 module의 條件附 信賴度는 식(3.1)로부터 구해진다.

$$R_m(t|K=k) = \left(\frac{\beta+\tau+T}{\beta+\tau+T+t}\right)^{(L-k)\alpha} \quad \cdots\cdots (5.1)$$

여기서 L은 현시점 τ에서 module내에 잔존해 있는 fault의 수이고 k는 0과 L사이의 정수값이다. 따라서 시간 $\tau+T$로 부터 module의 unconditional reliablility는 다음과 같이 구해진다.

$$R_m(T) = \Sigma R_m(t|K=k)\ \Pr(K=k) \quad \cdots\cdots\cdots (5.2)$$

한편 L개의 fault중 기간 $(\tau, \tau+T)$중 발견되어 수정될 확률은 binomial distribution 이므로

$$\Pr(K=k) = \binom{L}{k} P^k \cdot (1-P)^{L-k} \quad \cdots\cdots (5.3)$$

이다.

여기서 필요로 하는 것은 P의 값이다. P는 binomial distribution의 정의에 의하면 각 fault가 현시점 τ로 부터 T시간이 경과한 후인 $\tau+T$까지 발생될 확률이다. 이는 식(3.1)로 부터 쉽게 구해진다. Fault의 수는 1개이고 경과시간은 T이므로

$$R(t) = \left(\frac{\beta+\tau}{\beta+\tau+t}\right)^{\alpha} \quad \cdots\cdots (5.4)$$

이다. 한편 지금으로 부터 T시간이 경과하기까지 fault가 발생될 확률은

$F(t) = 1-R(t)$ 이므로 $F(T) = 1-\left(\frac{\beta+\tau}{\beta+\tau+T}\right)^{\alpha}$

이다.

따라서 P대신 F(T)를 식(5.3)에 대입하면

$$\Pr(K=k) = \binom{L}{C}\left[1-\left(\frac{\beta+\tau}{\beta+\tau+T}\right)^{\alpha}\right]^{k}\left(\frac{\beta+\tau}{\beta+\tau+T}\right)^{(L-k)\alpha} \quad \cdots\cdots\cdots (5.5)$$

가 된다. 따라서 식(5.1)과 식(5.5)를 식(5.2)에 대입하면

$$R_M(t) = \Sigma\left(\frac{\beta+\tau+T}{\beta+\tau+T+t}\right)^{(L-k)\alpha} \cdot \binom{L}{k}\left[1-\left(\frac{\beta+\tau}{\beta+\tau+T}\right)^{\alpha}\right]^{k}\left(\frac{\beta+\tau}{\beta+\tau+T}\right)^{(L-k)\alpha} \quad \cdots\cdots\cdots (5.6)$$

된다. 약간의 algebra를 통해서 간략화 하면

$$R_M(t) = \left[1-\left(\frac{\beta+\tau}{\beta+\tau+T}\right)^{\alpha}+\left(\frac{\beta+\tau}{\beta+\tau+T}\right)^{\alpha} \cdot \left(\frac{\beta+\tau+T}{\beta+\tau+T+t}\right)^{\alpha}\right]^{L} \quad \cdots\cdots\cdots (5.7)$$

이 된다. 여기서 $R'_M(t) = \frac{d\{R_M(t)\}}{dt}$로 놓으면 failure rate $\lambda(t)$는 $\frac{f(t)}{R(t)}$ 이므로 $\lambda(t) = \frac{-R'_M(t)}{R_M(t)}$ 에서 구해진다.

한편 식(5.7)로 부터 $R'_M(t)$가 구해지는데,

$$R'_M(t) = L\left[1-\left(\frac{\beta+\tau}{\beta+\tau+T}\right)^{\alpha}+\left(\frac{\beta+\tau}{\beta+\tau+T}\right)^{\alpha} \cdot \left(\frac{\beta+\tau+T}{\beta+\tau+T+t}\right)^{\alpha}\right]^{L-1} \cdot \left(\frac{\beta+\tau}{\beta+\tau+T+t}\right)^{\alpha}$$

$$\cdot (\beta+\tau+T)^{\alpha} \cdot (-\alpha) \cdot (\beta+\tau+T+t)^{-(\alpha+1)} \quad \cdots\cdots\cdots (5.8)$$

이다. 우리가 관심있는 것은 τ로 부터 T시간이 경과한 후의 failure rate와 평균 failure수이다. 따라서 $\tau+T$에서 경과시간이 0 즉 $t=0$에서의 failure rate는 $\lambda(0)=\frac{-R'(0)}{R_M(0)}$에서 구해진다. 식(5.7)로 부터 $R_M(0)=1$이므로 $\lambda(0)=-R'_M(0)$ 이다. 식(5.8)로 부터

$$\lambda(0)=-R'_M(0)$$

$$=\frac{L\alpha(\beta+\tau)^{\alpha}}{(\beta+\tau+T)^{\alpha+1}} \quad \cdots\cdots (5.9)$$

이다.

지금까지 시간 $\tau+T$에서의 module의 failure rate에 대해서 알아보았다. 다음은 T시간 동안 발생될 평균 failure수에 대해서 알아보자.

각 fault가 일어남은 binomial with $P=1-(\frac{\beta+\tau}{\beta+\tau+T})^{\alpha}$ 이고 현시점에서 잔존 fault의 수는 L개 이므로 T시간 동안에 일어날 평균 failure수는 $E\{N(T)\}=LP$

$$=L\{1-(\frac{\beta+\tau}{\beta+\tau+T})^{\alpha}\} \quad \cdots\cdots (5.10)$$

이다. 이제 m개의 module로 구성된 system의 $\tau+T$에서의 failure rate와 T시간이 경과하기까지 발생될 평균 failure수에 대해서 알아보자.

여기서 L_i를 시간 τ에서 module i에 잔존해 있는 fault의 수라면 각 module은 independent이므로 system의 reliability는 다음과 같이 표현된다.

$$R(t)=\left[1-(\frac{\beta+\tau}{\beta+\tau+T})^{\alpha_1}+(\frac{\beta+\tau}{\beta+\tau+T})^{\alpha_1} \cdot (\frac{\beta+\tau+T}{\beta+\tau+T+t})^{\alpha_1}\right]^{L_1}$$

$$\left[1-(\frac{\beta+\tau}{\beta+\tau+T})^{\alpha_2}+(\frac{\beta+\tau}{\beta+\tau+T})^{\alpha_2} \cdot (\frac{\beta+\tau+T}{\beta+\tau+T+t})^{\alpha_2}\right]^{L_2}$$

$$\vdots$$

$$\left[1-(\frac{\beta+\tau}{\beta+\tau+T})^{\alpha_m}+(\frac{\beta+\tau}{\beta+\tau+T})^{\alpha_m} \cdot (\frac{\beta+\tau+T}{\beta+\tau+T+t})^{\alpha_m}\right]^{L_m} \quad \cdots\cdots (5.11)$$

시간 $\tau+T$에서의 system의 failure rate는 $\frac{-R'(0)}{R(0)}$ 에 의해서 구해진다.

한편 식(5.11)에 의해서 $R(0)=R_1(0)\ R_2(0)\ \cdots\ R_m(0)$인데 $R_i(0)=1$이므로 $R(0)=1$이 된다. 따라서 시간 $\tau+T$에서의 system의 failure rate는 $-R'(0)$이다. 한편 식(5.11)로 부터

$$R'(t)=\frac{dR_1(t)}{dt}R_2(t)\ \cdots\ R_m(t)$$

$$+R_1(t)\frac{dR_2(t)}{dt}R_3(t)\ \cdots\ R_m(t)$$

$$+R_1(t)\ R_2(t)\ \cdots\ \frac{dR_m(t)}{dt} \quad \cdots\cdots (5.12)$$

이다.

여기서 $R_1(0)=1$ 이므로 $R'(0)=R'_1(0)+R'_2(0)+\cdots+R'_m(0)$ ……(5.13)

따라서 system의 failure rate는 $-R'(0)=-R'_1(0)-R'_2(0)-\cdots-R'_m(0)$

한편 $-R'_1(0)=\dfrac{L_1\alpha_1(\beta+\tau)^{\alpha_1}}{(\beta+\tau+T)^{\alpha_1+1}}$ 이므로 system의 failure rate

$\lambda(T)=-R'(0)$

$$=\Sigma\frac{L_1\alpha_1(\beta+\tau)^{\alpha_1}}{(\beta+\tau+T)^{\alpha_1+1}} \quad \cdots\cdots(5.14)$$

가 된다.

module은 각각 독립이므로 system의 평균 failure수는 각 module내 발생될 평균 failure수의 합으로 나타낼수 있다.

$$E(N(T))=\Sigma L_1\{1-(\frac{\beta+\tau}{\beta+\tau+T})^{\alpha_1}\} \quad \cdots\cdots(5.15)$$

지금까지 유도된 식(5.9), (5.10), (5.14), (5.15)로 부터 S-type, D-type, S-type과 D-type의 혼합형인 경우에 대한 module의 failure rate와 평균 failure수, m개로 구성된 system의 failure rate와 평균 failure수를 현시점 τ에서 잔존해 있는 fault로 L_1를 치환함으로써 쉽게 구할 수 있다.

5.2 R-type과 S+D+R-type 경우

앞절에서 유도된 절차에 의해 module의 reliability를 구하면

$$R_M(t)=\left[1-\frac{\beta}{\beta+T}+\frac{\beta}{\beta+T+t}\right]^L \quad \cdots\cdots(5.16)$$

이다.

식(5.16)로 부터 약간의 algibra를 통해서 간략히 하면 module의 T시간이 경과한 후의 failure rate는

$$\lambda_M(T)=\frac{\beta L}{(\beta+T)^2} \quad \cdots\cdots(5.17)$$

이다. 한편 failure의 일어남은 binomial with $P=1-\dfrac{\beta}{\beta+T}$ 이므로 module의 평균 failure 수는

$$E\{N(T)\}=L\,(1-\frac{\beta}{\beta+T}) \quad \cdots\cdots(5.18)$$

이다. 또한 앞절에서 사용된 절차를 통해서 system의 failure와 평균 failure수에 대해서 유도해 보면

$$\lambda(T)=\frac{\beta\Sigma L_1}{(\beta+T)^2} \quad \cdots\cdots(5.19)$$

$$E\{N(T)\}=\Sigma L_1\,(1-\frac{\beta}{\beta+T}) \quad \cdots\cdots(5.20)$$

이다.

R-type, S-D-R-type에서는 fault의 occurrence rate distribution을 exponential로 가정했음으로 전절에서 유도된 식들보다 간편한 form을 갖게 된다.

각 유형에 따른 module의 failure rate와 평균 failure수, system의 failure rate와 평균 failure

수는 L_1를 시간 τ에 잔존해 있는 fault 수로 대치함으로써 쉽게 구할 수 있다.

6. 소프트웨어 상품화 최적시기 결정모델

본 절에서는 전절에서 유도된 system의 failure rate와 평균 failure 수를 이용하여 소프트웨어 상품화 최적시기를 결정하는 모델을 발전시키고자 한다.

$$\text{Minimize } C_1E[N(T)]+C_2\{E[N(T_{LC})]-E[N(T)]\}+C_3g(T)$$

$$\text{subject to } \lambda_0(T_0) \le R_0$$
$$\lambda_1(T_1) \le R_1$$
$$\vdots$$
$$\lambda_m(T_m) \le R_m$$

여기서 $C_2 > C_1 > 0$, $C_3 > 0$이며

$E[N(T)]$: 기간 T동안 발생될 평균 failure수

$E[N(T_{LC})]$: 수명주기 동안 발생될 평균 failure수

$g(T)$: test시간 T와 관계되는 비용 함수

$\lambda_0(T_0)$: 시간 T에서의 system의 소프트웨어 고장율

$\lambda_i(T_i)$: 시간 T에서의 module i의 소프트웨어 고장율

R_0 : 전체 system의 한계 고장율

R_i : module i의 한계 고장율

본 모델은 Yamada and Osaki[14] 모델과 Kubat와 Koch 모델들의 결함들을 보완한 것이다.

이 모형을 발전시키기에 앞서 소프트웨어 테스트 기간중에 일어날 수 있는 일반적인 현상들을 먼저 생각해 보자.

일반적으로 fault를 제거하는데 소요되는 시간은 경과된 시스템 테스트 시간에 비례한다. 즉 시스템을 테스트하는 초기에는 fault들이 발견되기 쉽고 또 이러한 fault들은 대체로 간단한 것들이어서 제거시간이 적게 소요되지만 시스템을 테스트하는 시간이 경과될수록 발견되는 fault수가 점점 감소하고 발견된 fault의 수정도 용이하지 않아 긴 시간이 소요된다는 것이다.

먼저 발견된 fault의 제거 시간이 후에 발견된 fault 제거시간보다 언제나 짧은 것은 아니지만 대체로 초기에 발견된 fault들은 간단하여 fault 제거시간이 적게 걸리지만 테스트 시간이 많이 경과되어 대부분의 fault들이 제거된 경우에는 남아있는 fault들을 발견하는데 긴 시간이 걸리고 발견된 fault들은 일반적으로 근본적인 문제들로 제거작업에 많은 시간이 소요되는 것을 경험할 수 있다. 따라서 하나의 fault 제거시간이 시스템 테스트 경과시간에 비례한다는 것은 일반적으로 타당한 가정이다.

테스트 기간중에 발견된 fault는 fault 제거팀에 의해 제거된다고 본다. 이것은 Yamada와 Osaki 모형에서 전체 테스트 비용 C_3T와 fault 제거비용 $C_1E[N(T)]$를 고려할때 필요한 가정이

다. 즉 테스트 기간중에 fault를 발견한 요원이 직접 제거한다면 시스템의 테스트 기간중 비용은 C_3T만으로 표현될 수 있으므로 발견된 fault 수에 비례하는 비용인 $C_1E[N(T)]$는 무의미한 항이 될 것이다. 일반적으로 생각할때 단순한 시스템에서는 fault를 테스트한 요원이 직접 제거할 수도 있겠으나 좀 복잡한 시스템에서는 테스트 요원과 fault 제거요원을 구분 운용하여 발견된 fault는 제거팀에 의해 제거된다고 생각할 수 있다.

이럴 경우 테스트 기간중의 비용은 $C_1E[N(t)]$와 C_3T로 구분해서 표현하는 것이 타당하다.

소프트웨어 release 시간과 이윤은 반비례의 관계가 있다.

대체로 소프트웨어 테스트 비용이 release 후에 발견된 fault를 제거하는 비용보다 적기 때문에 총비용을 최소화하는 소프트웨어 release 시간은 길어지는 경향이 있다. 그러나 현실문제에서는 전체비용을 최소화 하기 위해 소프트웨어 release 시간이 길어질때 그 소프트웨어로 얻을 수 있는 이윤을 극대화 시키지 못하는 경우가 발생될 수 있다. 다시말하면 개발되는 각각의 소프트웨어는 시한적인 성질을 가지고 있어서 그 소프트웨어로 이윤을 최대화 할 수 있는 시간의 제약이 있다. 따라서 이러한 성질을 수식에 반영하여 release 시간이 너무 길어지는 것을 제한할 수 있다.

대체로 테스트 하려고 하는 시스템을 여러개의 모듈로 나눌 수 있다.

한 시스템을 m개의 모듈이 각각의 테스트요원에 의해 동시에 테스트한다는 테스트 방법을 기초하여 본 모형을 발전시켰다.

Yamada와 Osaki 모델은 test 시간과 관련되는 비용함수인 g(T)를 포함하고 있지 않으며 제약조건에 있어서도 각 모듈에 관계되는 제약조건은 없고 전체 system에 관련되는 하나의 제약조건만으로 간단한 모형이었다.

위 식으로 부터 상품화 최적시기를 나타내는 식을 유도하기는 대단히 어렵다.

따라서 Point Searching Alogrithm이 개발되었다.

Step 0. Initialization

a) given ε, R_0, T_0

b) $k=0$, $\Pi=2$

Step 1. $\lambda(T_k)$를 계산

Step 2.

a) $|R_0-\lambda(T_k)|\le\varepsilon$, go to Step 5

b) $|R_0-\lambda(T_k)|>\varepsilon$,

i) $R_0-\lambda(T_k)\ge 0$, go to Step 3

ii) $R_0-\lambda(T_k)<0$, go to Step 4

Step 3.

a) If $R_0-\lambda_0(T_{k-1})\ge 0$,

$T_k=T_k-\frac{T_k}{\Pi}$, $k=k+1$

go to Step 1

b) If $R_0-\lambda_0(T_{k-1})<0$,

$\Pi=\Pi+1$

$T_k = T_k - \frac{T_k}{II}$, $k = k+1$

go to Step 1

Step 4.

a) If $R_0 - \lambda_0(T_{k-1}) < 0$,

$T_k = T_k - \frac{T_k}{II}$, $k = k+1$

go to Step 1

b) If $R_0 - \lambda_0(T_{k-1}) \geq 0$,

$II = II + 1$

$T_k = T_k + \frac{T_k}{II}$, $k = k+1$

go to Step 1

Step 5. 종료

최적시기 $T_0^* = T_k$

모든 제약조건을 만족시키는 최적시기는 max $\{T_0^*, T_1^*, \cdots, T_m^*\}$이다.

식(5.15)에 의해 비선형 목적함수를 얻을수 있다.

$$f(T) = (C_1 - C_2)E[N(T)] + C_2E[N(T_{LC})] + C_3g(T)$$

$$= (C_1 - C_2)\Sigma(N_1 - M_1) \quad 1 - \left(\frac{\beta+\tau}{\beta+\tau+T}\right)^{\alpha_1} + C_2\Sigma(N_1 - M_1) \quad 1 - \left(\frac{\beta+\tau}{\beta+\tau+T_{LC}}\right)^{\alpha_1} + C_3g(T)$$

f(T)는 1차 및 2차 미분을 통해서 convex function 임을 쉽게 증명할 수 있다. Newton의 Point Searching Method에 의해 convex function의 minimum point를 얻을 수 있다. minimum point는 T_{k+1}와 T_k의 차이가 매우 적을때까지 아래 formula를 계속 적용함으로서 얻을 수 있다.

$$T_{k+1} = T_k - \frac{(C_1 - C_2)\Sigma(N_1 - M_1)\alpha_1 \dfrac{(\beta+\tau)^{\alpha_1}}{(\beta+\tau+T_k)^{\alpha_{1+1}}} + C_3g'(T_k)}{(C_1 - C_2)\Sigma(N_1 - M_1)\alpha_1(\alpha_1+1)\dfrac{(\beta+\tau)^{\alpha_1}}{(\beta+\tau+T_k)^{\alpha_{1+2}}} + C_3g''(T_k)}$$

T_c를 f(T)의 최적 point라 놓자.

따라서 비용을 최소화 하고 모든 제약식을 만족시키는 상품화 최적시기는 $\{T_c, T_0, T_1, T_2, \cdots, T_m\}$중에 있다. 2가지 경우가 가능하다; T_c가 보다 크거나 T_c가 max$\{T_0, T_1, \cdots, T_m\}$ 보다 작거나 같을 경우 2가지 경우를 고려할 때, 상품화 최적시기 T^*는 max$\{T_c, T_0, T_1, T_2, \cdots, T_m\}$이며 mimal point of f(T)는 모든 제약식들을 만족시킨다.

7. 산출된 결과

결과를 도출키 위해 개인용 컴퓨터인 Macintosh SE를 이용 FORTRAN-77의 언어에 의해 PROGRAM이 작성되었다. 산출된 결과는 아래 표 1과 같다.

표 1. 상품화 최적시기

$\tau = 0$		$\tau \neq 0$	
S-type	S-type	D-type	S+D type
113.5 (T_8)	87.6 (T_8)	76.2 (T_3)	90.01 (T_3)
71.54 (T_9)	83.3 (T_9)	52.4 (T_7)	77.6 (T_7)
87.65 (T_7)	88.7 (T_7)	65.6 (T_7)	75.7 (T_7)
87.4 (T_7)	85.2 (T_7)	57.5 (T_8)	74.1 (T_7)
111.02 (T_7)	85.2 (T_5)	57.3 (T_7)	63.6 (T_{10})

모든 test problem은 11개의 제약조건을 포함하고 있다. system에 1개의 제약조건 그리고 10개의 모듈을 위해 10개의 제약조건이 필요하다.

표 2에서 $\tau=0$ 및 $\tau\neq0$인 경우에 대해 g(T)가 변화함에 따라 소프트웨어 상품화 최적시기가 어떻게 변하는가를 볼 수 있다.

표 2. g(T)의 변화에 따른 상품화 최적시기

	$\tau = 0$		$\tau \neq 0$	
g(T)	S-type	S-type	D-type	S+D type
$T^{0.5}$	111.4 (T_c)	79.8 (T_c)	58.2 (T_c)	67.8 (T_c)
$T^{0.75}$	42.6 (T_c)	44.01 (T_5)	35.4 (T_3)	40.1 (T_3)
T	33.3 (T_5)	44.01 (T_5)	35.4 (T_3)	40.1 (T_3)
$T^{1.25}$	33.3 (T_5)	44.01 (T_5)	35.4 (T_3)	40.1 (T_3)
$T^{1.5}$	33.3 (T_5)	44.01 (T_5)	35.4 (T_3)	40.1 (T_3)

표 1과 비교해 볼때 몇개의 상품화 최적시기는 objective function의 최소점들이다. 즉 T_c가 $T_c, T_0, T_1, \cdots, T_m$중 최대 값이며 $g(T)=T^{0.5}$ or $T^{0.75}$인 경우 상대적으로 상품화 최적시기가 길어진다.

이제 제약조건 없이 g(T)의 변화에 따른 목적함수의 최소점에 대해서 알아보면 표 3에서와 같다.

표 3. g(T)의 변화에 따른 상품화 최적시기

	$\tau = 0$		$\tau \neq 0$	
g(T)	S-type	S-type	D-type	S+D type
$T^{0.5}$	97.61	72.64	54.47	64.04
$T^{0.75}$	39.62	29.9	22.68	26.48
T	20.13	15.22	11.48	13.43
$T^{1.25}$	11.68	8.8	6.6	7.74
$T^{1.5}$	7.48	5.64	4.24	4.96

g(T)의 power를 증가시킴에 따라 상품화 최적시기는 급격하게 감소함을 알 수 있다. 모듈에 관한 제약조건 등을 고려했을 때는 상품화 최적시기가 g(T)의 power를 증가함에 따라 급격하게 감소하지 않은 것과 좋은 대조를 이루고 있다.

8. 결 언

본 논문에서는 신뢰도에 대한 새로운 정의를 도출하였고 기존의 소프트웨어 신뢰도 측정모델의단점을 보완하기 위하여 fault를 세가지 유형으로 구분하여 신뢰도모델을 발전시켰다. 즉 fault를 유발시키는 fault의 유형을 Simple Fault, Degenerated Fault, Regenerated Fault로 구분하여 fault의 유형에 따라 모듈의 failure rate 및 평균 failure 수를 산출하는 식들을 유도해 냈으며 이 유도된 식들은 소프트웨어 상품화 최적시기를 결정하는 일반적이고 합리적인 새로운 모형의 발전을 가능케 했다.

참 고 문 헌

[1] Catuneanu, V.M. and A.N. Mihalache, "Improving the Accuracy of the Littlewood-Verall Model", IEEE Trans., Vol. R-34, No. 5, 1985.

[2] Currit, P.A., M. Dyer and H.D. Mills, "Certifying the Reliability of Software", IEEE Trans., Vol. SE-12, No. 1, 1986.

[3] Goel, A.L, "Forward software Reliability", IEEE Trans., Vol. SE-12, No. 1, 1986.

[4] Goel, A.L, and K. Okumoto, "Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures", IEEE Trans., Vol. R-28, pp. 206~211,

1979.

〔5〕 Joe, H and N. Reid, "On the Software Reliability Models of Jelinoki-Moranda and Littlewood", *IEEE Trans.*, Vol. R-34, No. 3, 1985.

〔6〕 Littlewood, B., "How to Measure Software Reliability and How not to Measure Software Reliability", *IEEE Trans.*, Vol. R-28, pp. 103~110, 1979.

〔7〕 Littlewood, B., "Software Reliability Model for Modular Program Structure", *IEEE Trans.*, Vol. R-28, pp. 241~247, 1979.

〔8〕 _____, "A Bayesian Differential Debugging Model for Software Reliability", Proc. COMPSAS, pp. 511~519, 1980.

〔9〕 _____, "Stochastic Reliability-Growth : A Model for Fault-Removal in Computer-Programs and Hardware-Designs", *IEEE Trans.*, Vol. R-30, No. 4, 1981.

〔10〕 Myers, G. T., "The Art of Software Testing", John Wiley.

〔11〕 Pressman, R. S., *Software Engineering; A Practitioner's Approach*, McGrow-Hill, 1982.

〔12〕 Schick, G. T. and R. W. Wolverton, "An Analysis of Competing Software Reliability Models", *IEEE Trans.*, Vol. SE-4, No. 2, 1987.

〔13〕 Shooman, M. L. *Probabilistic Models for Software Reliability Production*, New York, Academic, pp. 485~502, 1972.

〔14〕 Yamada, S. and S. Osaki, "Cost-Reliability Optimal Release Policies for Software System", *IEEE Trans.*, Vol. R-34, No. 5, 1985.

〔15〕 Zelkowitz, M. V., "Perspectives on Software Engineering", *Computer Surveys*, Vol. 10, pp. 197~210, 1978.