韓國 軍事運營分析 學會誌
第 9 卷 第 1 號, 1983. 6

컴퓨터·소프트웨어 開發時의 完成時間에 關한 研究

(A Study on the Completion Time of Computer Software Development)

金 仲 煥*
金 聖 植**

ABSTRACT

The ideal way of eliminating errors in a large scale software system is to test the software with all possible inputs, providing sufficiently large amount of execution time.

However, in practice, the test must be performed within given budget and time limits. Therefore, to perform the test under given constraints, we have to properly select inputs and determine the execution time for each selected inputs. This paper studies the distribution of number of errors at a given time as well as the distribution of time required to reduce the number of errors to a certain level. We assumed that error occurrence times are distributed exponentially (not necessarily identical) and the number of errors at the initial stage is known or estimable.

1. 序 論

컴퓨터 시스템을 하드웨어(hardware)部分과 소프트웨어(software)部分으로 區分할 때 시스템의 開發費用은 일반적으로 소프트웨어 部分이 하드웨어 部分의 약 2배 이상 소요되는 것으로 알려져 있다. [1] 또한 最近에 하드웨어 部分의 開發費用 감소에 따라 전체비용에서 소프트웨어部分이 차지하는 比率은 상대적으로 증가되는 추세이다.

Boehm [1,2]은 컴퓨터 소프트웨어 시스템의 개발단계를 分析 및 設計, 프로그램 作成, 試驗 및 集積 단계로 區分지을 경우, 使用目的에 따라 다소의 차이는 있으나 각 단계별 開發費用의 比率은 각각 전체비용의 35%, 15%, 50% (군사목적 : 35%, 15%, 50%, 우주계획목적 : 35%, 20%, 45%, IBM 360 운용목적 : 35%, 15%, 50%)로 推定된다고 하였다.

이와같이 開發費用의 대부분이 試驗 및 集積 단계에서 소요되지만 시스템이 대형(10^5 단위 이상의 單語)인 경우 상당한 투자에도

(*) 陸士 數學科
(**) 高麗大學校 產業工學科

불구하고 시스템의 複雜한 내부구조와 誤謬의 특성(誤謬의 갯수 및 위치)을 정확히 파악하기는 매우 어렵다. 따라서 시스템의 試驗時에는 시스템에 부여될 가능성이 있는 잘 알려진 入力을 實行(execution)하여 얻어진 出力을 조사함으로써 시스템내에 誤謬가 있다는 것을 알게되고 이를 修正하려고 할 때 새로운 誤謬의 添加 또는 不正確한 誤謬修正의 可能性이 항상 存在하게 된다.

그러므로 대형 소프트웨어 시스템을 試驗할 때 가장 이상적인 試驗方法은 시스템에 부여될 수 있는 모든 가능한 入力들을 충분한 時間을 두고 實行하며 誤謬를 修正해 나가는 일이다.

그러나 現實的으로는 자원(時間, 費用등)의 의 제한을 받게되어 이러한 試驗方法은 사실상 불가능하다. 따라서 試驗時에 고려 되어야 할 중요한 문제는 어떠한 入力들을 選擇하는가 하는것과 選擇된 入力들을 實行하는 적절한 實行時間을 선정하는 것이다.

本 論文은 試驗을 위한 入力들이 選擇되어졌을 경우 주어진 소프트웨어의 品質을 어느 선에서 보장하기 위하여는 實行時間을 어떻게 정할것인가를 결정하는데 目的을 두고 있다.

이러한 문제를 해결하기 위하여 Green[3,4]은 소프트웨어 시스템의 내부구조를 파악할 수 있는 경우에, 實行하게 되는 入力의 갯수를 試驗過程의 統制를 위한 決定變數(decision variable)로 정하여, 實行되는 入力갯수의 증가에 따라 시스템내에 있는 誤謬갯수의 變化를 推定하는 模型을 개발하였다.

本 硏究에서는 이와는 달리 選擇된 入力을 實行하는 實行時間을 試驗過程의 統制를 위한 決定變數로 정하였다. 즉 本 硏究는 試驗前 시스템內의 誤謬數를 推定할 수 있는 경우 時間에 따른 이의 감소현상 또는 誤謬의 數가 어떤 特定數까지 줄어드는데 걸리는 實行時間의 分布 문제를 다룬다. 일반적으로 시스템內에 있는 誤謬가 試驗時 나타나는 時點사이의 간격은 確率變數이며 또한 어떤 特定數까지 誤謬의 數가 줄어드는데 걸리는 시간도 確率分布로 表示되게 된다.

本 硏究는 이들의 分布를 구하는데 그 주안점을 두고 있다.

2. 試驗模型

試驗過程은 初期時間, $t=0$에 시스템이 未知갯수 n개의 誤謬를 包含한 狀態(state)에서 최초의 入力을 처리함으로써 시작되며 試驗중 誤謬를 발견하면 이를 故障(failure)의 발생으로 보아 誤謬를 즉시 修正함으로써 修正에 소요되는 時間은 고려하지 않고 컴퓨터가 실제 入力을 實行하는 實行時間만을 다룬다.

또한 誤謬修正時시는 誤謬를 正確히 修正하여 시스템에서 誤謬 한개를 完全히 제거하게 되는 경우와 不正確한 修正으로 인하여 그대로 시스템내에 다시 남아있게 되는 경우의 두가지 가능성만을 고려한다.

위와같은 경우에 故障과 故障사이의 實行時間이 指數分布를 한다고 假定하면 W_{ij}를 시스템이 i개의 誤謬를 가졌을때 $(j-1)$번째 故障과 j번째 故障사이의 實行時間을 나타낼때

$$\text{Prob.}\{W_{ij}\leq t\}=1-e^{-\lambda_i t}\cdots\cdots\cdots(1\cdot1)$$

$$i=1,2,\cdots\cdots\cdots,n$$

단, λ_i는 시스템내에 誤謬가 i개 있을 때 단위시간당 발견되는 誤謬의 갯수(고정된 故障率 : constant failure rate)

이 된다.

또한 誤謬修正時에 誤謬를 正確히 修正할 確率을 p라 하면 시스템에서 誤謬 한개를 完全히 제거하기 위하여 必要한 誤謬修正 횟수, N은 기하분포를 하므로

$$\text{Prob.}\{N=k\}=p(1-p)^{k-1}\cdots(1\cdot2)$$

$$k=1,2,\cdots$$

이다.

그러므로 i개의 誤謬를 가진 시스템에서 誤謬 한개를 完全히 제거하는데 소요되는 實行

時間, T_i 는

$$T_i = W_{i1} + W_{i2} + \cdots\cdots + W_{iN} \quad (1\cdot3)$$

과 같이 표시될 수 있다.

(1·1)식과 (1·2)식에 의하여

$$\mathrm{Prob.}\{T_i \leq t\} = \sum_{k=1}^{\infty} \mathrm{Prob.}\{T_i \leq t \mid N=k\}\cdot \mathrm{Prob.}\{N=k\}$$

$$= \sum_{k=1}^{\infty} \left[\int_0^t \frac{\lambda_i (\lambda_i t)^{k-1} e^{-\lambda_i t}}{(k-1)!} dt\right]\cdot p(1-p)^{k-1}$$

$$= 1 - e^{-p\lambda_i t}, \quad i = 1, 2, \cdots, n \quad (1\cdot4)$$

와 같이 된다.

따라서 試驗시작 初期時間에 n개의 誤謬를 가진 시스템에서 모든 誤謬를 완전히 제거하여 시스템을 完成시키는데 소요되는 實行時間 즉 시스템의 完成時間, T는

$$T = \sum_{i=1}^{n} T_i \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (1\cdot5)$$

와 같이 표시된다.

한편 시스템의 狀態를 시스템내에 있는 誤謬의 갯수로서 區分지으면 시스템의 狀態空間 (state space), E는 아래와 같이 표시된다.

$$E = \{0, 1, 2, \cdots\cdots, n-1, n\} \quad (1\cdot6)$$

Y_t를 時間 t에 시스템내에 있는 誤謬의 갯수라고 한다면 $\{Y_t = j\}$는 時間 t에 시스템내에 j개의 誤謬가 있는 狀態를 의미한다. 또한 각 狀態에 머무는 時間의 分布는 (1·4)식에 의해 指數分布임을 알 수 있다.

따라서 試驗過程, $Y = \{Y_t : t \in [0, \infty]\}$는 Markov 過程이며 시스템의 推移를 나타내는 Markov 그라프는 그림 1과 같이 된다.

표시를 간략하게 하기 위하여 $\mathrm{Prob.}\{Y_t = k\} = P_k(t)$로 표시하면 그림 1과 일치하는 微分方程式은 다음과 같다.

$$\dot{P}_{n-k}(t) + p\lambda_{n-k} P_{n-k}(t) = p\lambda_{n-k+1} \cdot P_{n-k+1}(t) \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (1\cdot7)$$

$$k = 0, 1, 2, \cdots\cdots, n$$

단, $\lambda_0 = 0$, $P_{n+1}(t) = 0$, $\dot{P}_k(t) = \dfrac{dP_k(t)}{dt}$

初期條件 : $P_n(0) = 1$, $\dot{P}_n(0) = -p\lambda_n$

$P_{n-k}(0) = 0$, $\dot{P}_{n-k}(0) = 0$,

$k = 1, 2, \cdots\cdots, n$

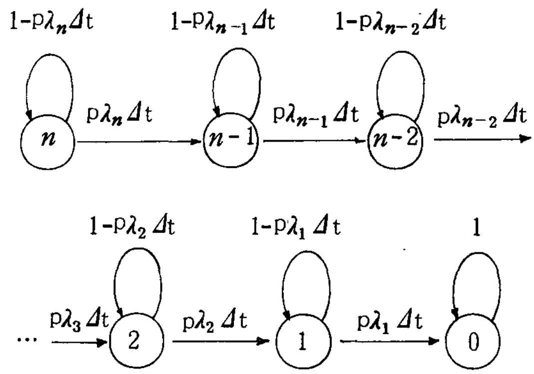

그림 1. 시스템의 推移를 나타내는 Markov 그라프

3. 시스템의 完成時間

微分方程式 (1·7)은 Laplace變換을 하고 初期條件을 대입하면

$$F_{n-k}(s) = \frac{\prod_{i=1}^{k} p\lambda_{n-i+1}}{\prod_{i=0}^{k} (s + p\lambda_{n-i})},$$

$$k = 1, 2, \cdots\cdots, n \quad \cdots\cdots\cdots \quad (2\cdot1)$$

단, $F_n(s) = (s + p\lambda_n)^{-1}$

$$F_k(s) = L\{P_k(t)\} = \int_0^{\infty} e^{-st} P_k(t) dt$$

과 같이 되고 (2·1)식을 逆變換 $P_k(t) = L^{-1}\{F_k(s)\}$를 취하면 다음과 같은 解를 구할 수 있다.

$$P_n(t) = e^{-p\lambda_n t} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (2\cdot2\cdot1)$$

$$P_{n-k}(t)=\sum_{j=0}^{k}\frac{\left[\prod_{i=1}^{k}\lambda_{n-i+1}\right]e^{-p\lambda_{n-j}t}}{\prod_{\substack{i=0\\i\neq j}}^{k}(\lambda_{n-i}-\lambda_{n-j})} \quad (2\cdot2\cdot2)$$

$k=1,2,\cdots\cdots,n$

단, $\lambda_0=0$, $\lambda_{n-i}\neq\lambda_{n-j}$

한편 (1·4)식에서 시스템이 i 狀態에 머무는 總時間의 確率密度函數는

$$f_{Ti}(t)=p\lambda_i e^{-p\lambda_i t} \quad \cdots\cdots\cdots\cdots\cdots\cdots (2\cdot3)$$

이므로 시스템을 完成시키는데 소요되는 實行時間, T의 確率密度函數는 (1·8)식에 의해

$$f_T(t)=f_{T_1}(t)*f_{T_2}(t)*\cdots\cdots*f_{T_n}(t) \quad (2\cdot4)$$

단, * : convolution 演算

가 되고 Laplace變換에 의해

$$f_T(s)=L\{f_T(t)\}$$

$$=\prod_{i=1}^{n}\frac{p\lambda_i}{(s+p\lambda_i)} \cdots\cdots\cdots\cdots\cdots (2\cdot5)$$

이 되므로 (2·5)식을 逆變換하면

$$f_T(t)=L^{-1}\{f_T(s)\}$$

$$=\sum_{j=1}^{n}\frac{\left[\prod_{i=1}^{n}\lambda_i p\right]e^{-\lambda_j pt}}{\prod_{\substack{i=1\\i\neq j}}^{n}(\lambda_i p-\lambda_j p)}$$

$$=\sum_{j=1}^{n}\frac{\left[P\prod_{i=1}^{n}\lambda_i\right]e^{-P\lambda_j t}}{\prod_{\substack{i=1\\i\neq j}}^{n}(\lambda_i-\lambda_j)},$$

단, $\lambda_i\neq\lambda_j$ $\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (2\cdot6)$

: 一般型 Erlang分布

이고 T의 平均과 分散은 각각

$$E[T]=\sum_{i=1}^{n}\frac{1}{p\lambda_i} \quad \cdots\cdots\cdots\cdots (2\cdot7\cdot1)$$

$$V(T)=\sum_{i=1}^{n}\left(\frac{1}{p\lambda_i}\right)^2 \cdots\cdots\cdots (2\cdot7\cdot2)$$

이 된다.

4. 特定한 故障率의 경우

特定한 故障率의 경우에 앞에서 구해진 解는 평소 우리에게 잘 알려진 結果들이 된다.

시스템내에 誤謬가 ℓ개 있을때의 故障率, λ_ℓ은 誤謬 한개에 대한 故障率을 λ라 할 때

$$\lambda_\ell=\ell^{\alpha}\lambda,\ \ell=1,2,\cdots\cdots,n \ \cdots\ (3\cdot1)$$

단, $0\leq\alpha\leq1$, α : smoothing parameter 와 같이 나타낼 수 있다고 하면,

1) $\alpha=0$인 경우

$\alpha=0$이면 (3·1)식에서 $\lambda_\ell=\lambda$ ($\ell=1,2,\cdots\cdots,n$) 이므로 (2·1)식에서

$$F_{n-k}(s)=\frac{p^k\lambda^k}{(s+p\lambda)^{k+1}} \cdots\cdots\cdots\cdots (3\cdot2)$$

이 되며 逆變換에 의하여

$$P_{n-k}(t)=\frac{(P\lambda t)^k e^{-P\lambda t}}{k!},\ k=0,1,2,\cdots\cdots,n \ \cdots\cdots\cdots\cdots\cdots (3\cdot3)$$

: Poisson 分布

가 되고 (2·5)식에서

$$f_T(s)=\left(\frac{p\lambda}{s+p\lambda}\right)^n \cdots\cdots\cdots\cdots (3\cdot4)$$

이므로 逆變換에 의하여

$$f_T(t)=\frac{(\lambda p)^n t^{n-1}e^{-p\lambda t}}{(n-1)!} \cdots\cdots (3\cdot5)$$

: Gamma 分布

이며

$$E[T]=\frac{n}{p\lambda} \cdots\cdots\cdots\cdots\cdots\cdots (3\cdot6\cdot1)$$

$$V(T)=\frac{n}{(p\lambda)^2} \cdots\cdots\cdots\cdots\cdots (3\cdot6\cdot2)$$

이다.

2) $0 < \alpha < 1$ 인 경우

이 경우에는 (2·2)식에서

$$\prod_{i=1}^{k} \lambda_{n-i+1} = \prod_{i=1}^{k} (n-i+1)^{\alpha}\lambda = \lambda^{k}[k!\binom{n}{k}]^{\alpha}$$

$$\prod_{\substack{i=0\\ i\neq j}}^{k} (\lambda_{n-i} - \lambda_{n-j}) = \lambda^{k} \prod_{\substack{i=0\\ i\neq j}}^{k} [(n-i)^{\alpha} - (n-j)^{\alpha}]$$

이므로

$$P_{n-k}(t) = \sum_{j=0}^{k} \frac{[k!\binom{n}{k}]^{\alpha}\, e^{-(n-j)^{\alpha} p\lambda t}}{\prod_{\substack{i=0\\ i\neq j}}^{k} [(n-i)^{\alpha} - (n-j)^{\alpha}]} \quad \cdots\cdots (3\cdot 7)$$

$k = 1, 2, \cdots\cdots, n$

단, $\binom{n}{k} = \dfrac{n!}{k!(n-k)!}$

이 되며 (2·6)식에서

$$\prod_{i=1}^{n} \lambda_i = \prod_{i=1}^{n} i^{\alpha}\lambda = \lambda^{n} \prod_{i=1}^{n} i^{\alpha} = \lambda^{n}(n!)^{\alpha}$$

$$\prod_{\substack{i=1\\ i\neq j}}^{n} (\lambda_i - \lambda_j) = \prod_{\substack{i=1\\ i\neq j}}^{n} [i\lambda - j^{\alpha}\lambda] = \lambda^{n-1} \prod_{\substack{i=1\\ i\neq j}}^{n} (i^{\alpha} - j^{\alpha})$$

이므로 T의 確率密度函數는

$$f_{T}(t) = p\lambda (n!)^{\alpha} \sum_{j=1}^{n} \frac{e^{-j^{\alpha} p\lambda t}}{\prod_{\substack{i=1\\ i\neq j}}^{n} (i^{\alpha} - j^{\alpha})} \quad (3\cdot 8)$$

이고 T의 平均과 分散은

$$E[T] = \frac{1}{p\lambda} \sum_{i=1}^{n} \left(\frac{1}{i^{\alpha}}\right) \quad \cdots\cdots\cdots\cdots (3\cdot 9\cdot 1)$$

$$V(T) = \left(\frac{1}{p\lambda}\right)^{2} \sum_{i=1}^{n} \left(\frac{1}{i^{2\alpha}}\right) \quad \cdots\cdots (3\cdot 9\cdot 2)$$

이다.

3) $\alpha = 1$ 인 경우

$\alpha = 1$ 이면 시스템의 故障率, λ_l는 그때 시스템에 包含되어 있는 誤謬의 갯수 ℓ에 비례하는 경우로서 (3·1)식에서

$$\lambda_e = \ell\lambda \ (\ell = 1, 2, \cdots\cdots, n)$$

이고

$$P_{n-k}(t) = \sum_{j=0}^{k} \frac{k!\binom{n}{k} e^{-(n-j)\lambda p t}}{\prod_{\substack{i=0\\ i\neq j}}^{k} (j-i)} \quad (3\cdot 10)$$

이며 (3·10)식에서

$$\prod_{\substack{i=0\\ i\neq j}}^{k} (j-i) = j!(-1)^{k-j}(k-j)!$$

이므로

$$P_{n-k}(t) = \sum_{j=0}^{k} \frac{k!\binom{n}{k} e^{-(n-j)\lambda p t}}{(-1)^{k-j} j!(k-j)!}$$

$$= \sum_{j=0}^{k} (-1)^{k-j} \binom{n}{k}\binom{k}{j} e^{-(n-j)\lambda p t}$$

$$= \binom{n}{k} [e^{-p\lambda t}]^{n-k} [1 - e^{-p\lambda t}]^{k},$$

$k = 1, 2, \cdots\cdots, n \quad \cdots\cdots\cdots (3 \cdot 11)$

: 이항분포

이다.

또한 (2·6)식에서

$$f_{T}(t) = p\lambda (n!) \sum_{j=1}^{n} \frac{e^{-jp\lambda t}}{\prod_{\substack{i=1\\ i\neq j}}^{n} (i-j)} \quad (3\cdot 12)$$

이며 (3·12)식에서

$$\prod_{\substack{i=1\\ i\neq j}}^{n} (i-j) = (-1)^{j-1}(j-1)!(n-j)!$$

이므로 T의 確率密度函數는

$$f_{T}(t) = n!\, p\lambda \sum_{j=1}^{n} \frac{e^{-j\lambda p t}}{(-1)^{j-1}(j-1)!(n-j)!}$$

$$= \sum_{j=1}^{n} (-1)^{j-1} p\lambda j \binom{n}{j} e^{-j\lambda pt}$$

$$= \sum_{j=1}^{n} (-1)^{j} \binom{n}{j} \frac{d}{dt} e^{-j\lambda pt}$$

$$= \frac{d}{dt} \left[\sum_{j=0}^{n} \binom{n}{j} [-e^{-\lambda pt}]^{j} \right]$$

$$= \frac{d}{dt} [1 - e^{-\lambda pt}]^{n}$$

$$= n\lambda p \, e^{-\lambda pt} [1 - e^{-\lambda pt}]^{n-1} \quad (3\cdot13)$$

이고 T의 平均과 分散은

$$E[T] = \frac{1}{p\lambda} \sum_{i=1}^{n} \frac{1}{i} \quad (3\cdot14\cdot1)$$

$$V(T) = \left(\frac{1}{p\lambda}\right)^2 \sum_{i=1}^{n} \left(\frac{1}{i}\right)^2 \quad (3\cdot14\cdot2)$$

이다.

5. 結 論

Rizza 및 Hacker[5]는 컴퓨터 소프트웨어 시스템의 開發을 각 모듈(module)별로 開發하고 이를 集積하는 경우에 集積단계에서의 試驗費用은 각 모듈 개발단계에의 試驗費用보다 약 3배 가량 많이 소요된다고 하였다.

이러한 면에서 볼 때 Green[4]의 模型은 비교적 시스템의 크기가 작고 내부구조를 파악하기 쉬운 모듈 개발단계에서 실시하는 試驗에 적합하다고 보지만 여러개의 모듈이 集積되며는 시스템에 부과되는 모든 가능한 작업들의 集合으로 이루어진 入力空間의 位相(topology)이 모듈개발때와는 다르게 되므로 최종적인 소프트웨어 시스템의 品質을 보장하기에는 모듈개발단계에서의 試驗만으로는 미흡하다.

本 硏究에서의 結果들은 모듈개발단계에는 물론 集積단계에서의 試驗에도 적용될 수 있다는 점에서 보다 實效性이 있다.

本 硏究에서는 誤謬修正시에 正確한 修正과 不正確한 修正의 두가지 가능성만 고려하였으나 새로운 誤謬가 添加될 가능성까지 고려한다면 더욱 改善된 試驗模型이 될 것이다.

또한 本 硏究에서의 結果들을 이용하여 주어진 實行時間에서 소프트웨어의 品質을 推定하는 문제와 주어진 소프트웨어의 品質이 되기 위한 實行時間에 관한 문제는 계속 硏究되어야 할 문제로 보여진다.

參 考 文 獻

1. B.W. Boehm, "Software and Its Impact: A Quantitative Assessment." Datamation, 48-59(May 1973).
2. B.W. Boehm, "The High Cost of Software." Proceedings of the Symposium on the High Cost of Software, (Jack Goldberg, Ed.), Stanford Research Institute, 27-40(1973).
3. T.F. Green, G.H. Bradley, G.T. Howard, and N.F. Schneidewind, "Structure and Error Detection in Computer Software." Proceedings AIIE National Conference, 54-59(1975).
4. T.F. Green, N.F. Schneidewind, G.T. Howard, and R.J. Pariseau, "Program Structures, Complexity and Error Characteristics." Proceedings of the Symposium on Computer Software Engineering, Polytechnic Institute of New York, 139-154 (April 20-22, 1976).
5. J.B. Rizza and D. Hacker, "Quality Assurance Inspection and Test Tools – An Application." Proceedings of a Workshop on Currently Available Program Testing Tools, 1, 9-10(April 1975).