5ZC－4

# 障害の大きさを考慮したソフトウェアの逐次信頼性実証試験に関する離散型モデル

澤田　清　　三道　弘明
流通科学大学 情報学部

## 1. はじめに

信頼性実証試験[1]は，ハードウェア製品の開発段階終了後，そのハードウェアに目標とする信頼性が十分に実現されているかどうかの実証・確認を目的として考案された．ソフトウェアの品質保証が問題となっている今日，ソフトウェア製品に対してもこのような信頼性実証試験を実施することは，信頼性という意味での品質向上に貢献すると考えられる．

このような考え方に基づき，筆者らはこれまで，計算機のＯＳや生産システムの制御ソフトウェアのように時間に関して連続的に用いられるソフトウェア（連続型モデル），および数値計算ソフトウェアや通常のアプリケーションソフトウェアのように時間に関して離散的に使用されるソフトウェア（離散型モデル）に対して，信頼性実証試験の適用を試みた[2]．そこでは，あらかじめ決められた試験時間（連続型モデルの場合），もしくは試験入力数（離散型モデルの場合）の間に生起するソフトウェア障害の回数により試験の合否を判定する方法を考えた．さらに筆者らは，ソフトウェア障害による損失の大きさ（以後，ソフトウェア障害の大きさとよぶ）が何らかの方法で定量的に評価できるものと仮定し，ソフトウェア障害回数だけでなく，ソフトウェア障害の大きさの累積をも試験の合否の判定に使用する方法を提案した[3]．

本研究では，時間に関して離散的に用いられるソフトウェアを対象として，ソフトウェア障害の大きさを考慮した逐次信頼性実証試験を考える．これは，試験入力数があらかじめ決められているのではなく，１回の試験入力のたびに，障害の大きさの累積に基づき合格・不合格・試験続行を判定する試験である．

## 2. 問題の設定

ここでは，ソフトウェア障害の大きさを考慮した次のような逐次信頼性実証試験を考える．すなわち，対象ソフトウェアに対する可能な入力の中から任意に選んだ１つずつの入力を用いて対象ソフトウェアをテスト実行し，１回のテスト実行終了ごとに，次の(1)～(3)の判定を行う．ただし，$n(n=1,2,\cdots)$番目の入力までの間に生起したソフトウェア障害の大きさの累積を$R_n$とする．

(1) $R_n \leq A(n)$ならば，対象ソフトウェアを合格とする．

(2) $R_n \geq B(n)$ならば，対象ソフトウェアを不合格とする．

(3) $A(n) < R_n < B(n)$ならば，合否の決定は下さず，$(n+1)$番目の入力を行う．

このとき，$A(n)$，$B(n)$をいかに設定するかが問題である．

ここで，ソフトウェアの生産者（開発者）がその開発を受注したときのソフトウェア障害の大きさの平均（以後，平均ソフトウェア障害サイズとよぶ）に対する契約の値，および消費者（ユーザ）が受け入れ可能な平均ソフトウェア障害サイズの上限値をそれぞれ$\mu_0$，$\mu_1$と書くこととする（$\mu_0 < \mu_1$）．ただし，ここで，平均ソフトウェア障害サイズは，障害が生起しなかった入力についても障害の大きさが0であるとしてその計算に含めるものとする．

また，ここでは，次のように仮定する．

(I) 試験中に発生したソフトウェア障害に対するフォールト（ソフトウェア障害の原因となるプログラム内の誤り）の検出・修正は，試験終了後にまとめて実施する．すなわち，平均ソフトウェア障害サイズは試験の間は変化しない．

(II) $\mu_0$，$\mu_1$は試験開始時点での平均ソフトウェア障害サイズ対する値を表す．

## 3. モデル１

ここでは，ソフトウェア障害の大きさの分布がポアソン分布である場合を考える．すなわち，上に述べた逐次信頼性実証試験の$i$番目の入力に対して，生起したソフトウェア障害の大きさを表す確率変数を$X_i$としたとき，対象ソフトウェアの平均ソフトウェア障害サイズの値が$\mu$であるときの$X_i$の確率関数は

$$f(x_i|\mu) = \frac{e^{-\mu}\mu^{x_i}}{x_i!} \quad (x_i = 0,1,2,\cdots) \qquad (1)$$

で与えられる．このとき，$x_i = 0$はソフトウェア障害が生起しないことを，$x_i = 1,2,\cdots$はソフトウェ

---

Discrete Models for Sequential Software Reliability Demonstration Testing Considering Damage Size of Software Failures
Kiyoshi Sawada and Hiroaki Sandoh
University of Marketing and Distribution Sciences
3-1 Gakuen-nishi-machi, Nishi-ku, Kobe 651-2188, Japan

ア障害が生起することを表す．すなわち，この場合のソフトウェアの不信頼度（1つの入力に対して障害が生起する確率）は，$f(x_i = 1, 2, \cdots | \mu) = 1 - e^{-\mu}$ である．

ここで，$H_0 : \mu = \mu_0$ を帰無仮説，$H_1 : \mu = \mu_1$ を対立仮説とした場合の逐次検定比は

$$\phi_n = \frac{\prod_{i=1}^{n} f(x_i|\mu_1)}{\prod_{i=1}^{n} f(x_i|\mu_0)} = \left(\frac{\mu_1}{\mu_0}\right)^{R_n} e^{-(\mu_1-\mu_0)n} \quad (2)$$

と計算できる．このとき，第1種の過誤の確率（生産者リスク），第2種の過誤の確率（消費者リスク）の上限値がそれぞれ，$\alpha$，$\beta$ として与えられているとすると，Wald[4] の逐次解析理論より，次の結果が得られる．

(i) $R_n \leq -k_0 + sn$ ならば，対象ソフトウェアを合格とする．

(ii) $R_n \geq k_1 + sn$ ならば，対象ソフトウェアを不合格とする．

(iii) $-k_0 + sn < R_n < k_1 + sn$ ならば，合否の決定は下さず，$(n+1)$ 番目の入力を行う．

ただし，

$$k_0 = \frac{\ln \dfrac{1-\alpha}{\beta}}{\ln \dfrac{\mu_1}{\mu_0}} \quad (3)$$

$$k_1 = \frac{\ln \dfrac{1-\beta}{\alpha}}{\ln \dfrac{\mu_1}{\mu_0}} \quad (4)$$

$$s = \frac{\mu_1 - \mu_0}{\ln \dfrac{\mu_1}{\mu_0}} \quad (5)$$

である．

## 4. モデル2

モデル1では，ソフトウェアの不信頼度が $1 - e^{-\mu}$ であるので，ソフトウェア不信頼度を現実的な小さい値に設定するには，$\mu$ の値をかなり小さくする必要がある．そこで，モデル2では，モデル1の $X_i = 1, 2, \cdots$ の確率にそれぞれ $r(0 < r \leq 1)$ を乗じて $X_i = 1, 2, \cdots$ の確率を小さくした分布を考え，ソフトウェアの不信頼度の値が小さくなるようにする．すなわち，平均ソフトウェア障害サイズの値が $\mu$ であるときの $X_i$ の確率関数を

$$f(x_i|\mu) = 1 - r(1 - e^{-\frac{\mu}{r}}) \quad (x_i = 0) \quad (6)$$

$$f(x_i|\mu) = \frac{e^{-\frac{\mu}{r}} \left(\dfrac{\mu}{r}\right)^{x_i} r}{x_i!} \quad (x_i = 1, 2, \cdots) \quad (7)$$

とする．ただし，$r$ の値は既知とする．ここで，$r = 1$ とした場合は，モデル1と同じである．すなわち，モデル2はモデル1を一般化したものであると考えることができる．

ここで，仮説 $H_0 : \mu = \mu_0$，$H_1 : \mu = \mu_1$ の逐次検定比を求めると

$$\phi_n = \left(\frac{\mu_1}{\mu_0}\right)^{R_n} e^{-\frac{\mu_1-\mu_0}{r}T_n} \left[\frac{1 - r(1 - e^{-\frac{\mu_1}{r}})}{1 - r(1 - e^{-\frac{\mu_0}{r}})}\right]^{n-T_n} \quad (8)$$

となる．ただし，$T_n$ は，$n(n = 1, 2, \cdots)$ 番目の入力までの間でソフトウェア障害を引き起こした入力の数である．このとき，モデル1と同様に，生産者リスク，消費者リスクの上限値をそれぞれ $\alpha$，$\beta$ として，Wald の逐次解析理論を適用すると，次の結果が得られる．

(i) $R_n \leq -k_0 + an + bT_n$ ならば，対象ソフトウェアを合格とする．

(ii) $R_n \geq k_1 + an + bT_n$ ならば，対象ソフトウェアを不合格とする．

(iii) $-k_0 + an + bT_n < R_n < k_1 + an + bT_n$ ならば，合否の決定は下さず，$(n+1)$ 番目の入力を行う．

ただし，

$$a = \frac{-\ln \dfrac{1 - r(1 - e^{-\frac{\mu_1}{r}})}{1 - r(1 - e^{-\frac{\mu_0}{r}})}}{\ln \dfrac{\mu_1}{\mu_0}} \quad (9)$$

$$b = \frac{\dfrac{\mu_1 - \mu_0}{r} + \ln \dfrac{1 - r(1 - e^{-\frac{\mu_1}{r}})}{1 - r(1 - e^{-\frac{\mu_0}{r}})}}{\ln \dfrac{\mu_1}{\mu_0}} \quad (10)$$

である．

## 参考文献


[1] H. F. Martz and R. A. Waller, "Bayesian Reliability Analysis", John Wiley & Sons, New York, pp.466–486, 1982.

[2] K. Sawada and H. Sandoh, "A summary of software reliability demonstration testing models", *International Journal of Reliability, Quality and Safety Engineering*, Vol.6, No.1, 1999 (to appear).

[3] 澤田　清, 三道弘明, "障害の大きさを考慮したソフトウェアの信頼性実証試験", 電子情報通信学会論文誌, Vol.J81–A, No.1, pp.98–109, 1998.

[4] A. Wald, "Sequential Analysis", John Wiley & Sons, New York, 1947.