

エラー発見率に基づく S 字形ソフトウェア信頼度成長モデルの考察†

山 田 茂^{††} 大 場 充^{†††}

本論文では、ソフトウェアの信頼性評価のために開発された二つの代表的な S 字形ソフトウェア信頼度成長モデルについて議論する。これらのモデルは、ソフトウェア開発の最終段階である試験において、発見された総ソフトウェアエラー数が S 字形成長曲線を示す現象を記述するものである。このようなエラー発見事象は、確率事象として取り扱うことにより、非同次ポアソン過程という確率過程を導入してモデル化することが可能となる。ここで考察の対象となるのは、遅延 S 字形ソフトウェア信頼度成長モデルおよび習熟 S 字形ソフトウェア信頼度成長モデルである。各モデルの特性について、ソフトウェアの信頼度成長を把握する 1 指標であるエラー発見率により考察する。このエラー発見率としては、瞬間エラー発見率と 1 個当りのエラー発見率を考え、実際のソフトウェアエラーデータに対する適用例により各モデルの特性を明らかにする。

1. ま え が き

近年、高信頼性のソフトウェア製品を効率良く開発するための方法論が、ソフトウェア工学の各分野で活発に研究されてきた。このうち、ソフトウェアの信頼性評価に関する研究は、ソフトウェアの開発管理を行う上で重要である。したがって、ソフトウェア開発の最終段階である試験により得られた観測データをもとに、開発の結果としてのソフトウェアに対して定量的な信頼性評価を行うことは意義のあることであり、よく取られるアプローチでもある。これにより、多大な経費が必要な試験に対する進捗管理やソフトウェア製品の出荷可否の決定が可能になる。

このような背景から、ソフトウェアの信頼性評価のために、種々の信頼性モデルが提案されてきた。ソフトウェアの信頼性をモデル化する際によく用いられる方法の一つは、信頼度成長理論 (reliability growth theory) に基礎をおくものである。これは、次のように説明される。ソフトウェア開発の最終段階では、ソフトウェア内に潜在するエラーを発見・修正するために試験 (テスト) が行われる。ここで、発見されたエラーはすべて修正・除去されるものと仮定する。もし、エラーの修正時に新しいエラーは作り込まれないものとする、試験が進行するにつれてソフトウェア内に潜在するエラー数は減少し、ソフトウェア故障の

発生しない確率は増加する。すなわち、ソフトウェアの信頼度は増加することになる。このようなソフトウェアエラーの発見事象を記述し、ソフトウェアの信頼性 (あるいは性能) 評価を行うために開発された信頼性モデルは、ソフトウェア信頼度成長モデル (software reliability growth model) (Ramamoorthy and Bastani¹⁾ 参照) と呼ばれている。一般に、ソフトウェア故障とは、ソフトウェア内に潜在するエラーにより、実行中に期待どおりの動作をしないことと定義する。また、1 個のソフトウェアエラーの発見は、1 回のソフトウェア故障の発生に対応しているものとする。

本論文では、ソフトウェア信頼度成長モデルの立場から、試験において発見された総エラー数が時間的傾向として S 字形成長曲線を示すような、エラー発見事象を記述するために提案された二つの代表的な確率モデルについて考察する。これらは、S 字形ソフトウェア信頼度成長モデル (S-shaped software reliability growth model) と呼ばれる。基本となる確率過程は、非同次ポアソン過程 (nonhomogeneous Poisson process, 以下 NHPP と略す) (Ascher and Feingold²⁾ 参照) である。特に、信頼度成長の指標の一つであるエラー発見率として、瞬間エラー発見率および 1 個当りのエラー発見率を取り上げ、各モデルの特性を議論する。さらにこれを明示するために、実際の試験において観測されたソフトウェアエラーデータを用いた具体例も与える。

2. ソフトウェア信頼度成長モデル

ソフトウェア信頼度成長モデルの分野では、ソフト

† S-Shaped Software Reliability Growth Models Characterized by Error Detection Rates by SHIGERU YAMADA (Department of Electronic Engineering, Okayama University of Science) and MITSURU OHBA (Science Institute, IBM Japan, Ltd.).

†† 岡山理科大学工学部電子工学科

††† 日本アイ・ビー・エム(株)サイエンス・インスティテュート

ウェア故障発生時間間隔あるいは発見された総ソフトウェアエラー数を確率量としてモデル化が行われる。本論文では、後者の確率量を基本とする。そこで、試験時刻 t までに発見された総ソフトウェアエラー数を表す計数過程 $\{N(t), t \geq 0\}$ を定義する。この確率過程に対して NHPP を仮定すると、

$$P_r \{N(t) = n\} = \frac{\{a(t)\}^n}{n!} \exp[-a(t)]$$

$$(n=0, 1, 2, \dots), \quad (1)$$

$$a(t) = \int_0^t \lambda(x) dx, \quad t \geq 0, \quad (2)$$

により、ソフトウェア信頼度成長モデルを定式化することができる。ここで $a(t)$ は、NHPP の平均値関数 (mean value function) と呼ばれ、時刻 t までに発見される総期待エラー数を表す。また $\lambda(t)$ は、NHPP の強度関数 (intensity function) と呼ばれ、時刻 t における瞬間エラー発見率、すなわち時刻 t において単位時間当りに発見される総期待エラー数を表す。

式(1)および式(2)から、エラー発見事象におけるソフトウェアの信頼度成長を特徴づける指標として、試験時刻 t における (残存エラー数に対する) 1 個当りのエラー発見率を、

$$d(t) \equiv \frac{\lambda(t)}{a - a(t)}, \quad (3)$$

のように定義することができる。ここでパラメータ a は、試験前にソフトウェア内に潜在する総期待エラー数を表す。Yamada and Osaki³⁾ は、式(3)のエラー発見率 $d(t)$ に対して次の定義を導入した：試験時間 t に関して $d(t)$ が非減少関数あるいは非増加関数であるならば、それぞれ平均値関数 $a(t)$ を増加型エラー発見率 (increasing error detection rate, 以下 IEDR と略す) 関数あるいは減少型エラー発見率 (decreasing error detection rate, 以下 DEDR と略す) 関数であるという。ここで、 $a(t)$ が IEDR (DEDR) 関数であるとは、試験が進行するにつれて、残存エラー数に対して新しくエラーの発見される可能性が増加する (減少する) ことを意味している。

さらに、式(1)および式(2)により定式化されたソフトウェア信頼度成長モデルから、ソフトウェアの信頼性評価に有用な定量的尺度を導出できる。時刻 t においてソフトウェア内に残存するエラー数を $\bar{N}(t)$ とすると、その期待値は

$$n_r(t) \equiv E[\bar{N}(t)]$$

$$= a - a(t), \quad (4)$$

により与えられる。式(4)は $\bar{N}(t)$ の分散とも一致し、残存エラー数 $\bar{N}(t)$ はポアソン分布に従うことがわかる。また、時刻 t においてソフトウェア故障が発生したときに、時間区間 $(t, t+x)$ ($x \geq 0$) で次のソフトウェア故障の発生しない確率は

$$R(x|t) = \exp[-\{a(t+x) - a(t)\}], \quad t \geq 0, \quad x \geq 0, \quad (5)$$

により与えられる。この信頼度関数は、ソフトウェア信頼度 (software reliability) と呼ばれている。

3. S 字形ソフトウェア信頼度成長モデル

第2章で導入された確率変数 $N(t)$ の実現値、すなわち実際の試験で発見された総ソフトウェアエラー数の時間的挙動は、試験時間に対するその成長曲線により把握することができる。実際の試験では、指数関数形あるいはS字曲線形を示す典型的な二つの成長曲線が観測される。Goel and Okumoto⁴⁾, Littlewood⁵⁾, および Musa⁶⁾ らのソフトウェア信頼度成長モデルでは、前者の指数形成長曲線を示すようなエラー発見事象を記述している。特に、Goel and Okumoto⁴⁾ は、平均値関数を

$$a(t) \equiv m(t) = a[1 - e^{-bt}], \quad a > 0, \quad b > 0, \quad (6)$$

とする NHPP により、この現象を記述している。ここで、パラメータ a は試験前にソフトウェア内に潜在する総期待エラー数、パラメータ b は任意の試験時刻における 1 個当りのエラー発見率である。このモデルは、NHPP に基づく指数形ソフトウェア信頼度成長モデルと呼ばれる。

これに対して、日本では従来より後者のS字形成長曲線を示すとして、人口や需要の予測などで用いられるロジスティック (logistic) 曲線やゴンペルツ (Gompertz) 曲線を観測データにあてはめる方法がとられてきた (例えば菅野⁷⁾, 当麻⁸⁾ 参照)。この方法は、各成長曲線の収束値をソフトウェア内に潜在する総エラー数として、回帰分析により推定するものである。さらに、エラー発見事象の物理的意味を踏まえた上で、発見された総エラー数がS字形成長曲線を示す場合に適切な、NHPP に基づくソフトウェア信頼度成長モデルが提案された。ソフトウェアエラーの除去過程を記述する遅延S字形ソフトウェア信頼度成長モデル (delayed S-shaped software reliability growth model) (山田・尾崎⁹⁾, Yamada et al.¹⁰⁾ およびソフトウェア故障の発生現象を記述する習熟S字形ソフトウェア信頼度成長モデル (inflection S-shaped soft-

ware reliability growth model) (大場・梶山¹¹⁾, Ohba¹²⁾) である。

以下では、これら二つの NHPP に基づく S 字形ソフトウェア信頼度成長モデルの特性について議論する。

3.1 遅延 S 字形ソフトウェア信頼度成長モデル

このモデルは、試験におけるソフトウェアエラーの除去過程を記述するために開発されたものである。本モデルによれば、試験におけるエラー発見事象が、ソフトウェア故障の発見 (failure detection) とソフトウェアエラーの認定 (error isolation) という二つの過程から成るものと考えている。

一般に比較的複雑なソフトウェアの試験においては、ソフトウェア故障の現象を観測した後にその原因解析を十分に行わなければ、ソフトウェア故障の再現は困難となり、最終的にソフトウェアエラーの発見 (すなわちソフトウェアエラーの認定) には至らない。このうち、ソフトウェア故障の現象を観測する過程がソフトウェア故障発見過程と呼ばれ、その原因解析を行ってソフトウェアエラーの発見に至るまでの過程はソフトウェアエラー認定過程と呼ばれる。そこで、パラメータ a を試験前にソフトウェア内に潜在する総期待エラー数、パラメータ b_1 をソフトウェア故障発見率、パラメータ b_2 をソフトウェアエラー認定率とする。ソフトウェア故障発見過程に対して、 $m_f(t)$ を時刻 t までに発見される総期待ソフトウェア故障数とすると、

$$\frac{dm_f(t)}{dt} = b_1[a - m_f(t)], \quad (7)$$

が成立する。また、これに引き続くソフトウェアエラー認定過程に対して、 $M(t)$ を時刻 t までに発見される総期待エラー数とすると、

$$\frac{dM(t)}{dt} = b_2[m_f(t) - M(t)], \quad (8)$$

が成立する。したがって、近似的に $b_1 = b_2$ として式 (7) および式 (8) の微分方程式を解くと、NHPP の平均値関数

$$a(t) \equiv M(t) = a[1 - (1 + bt)e^{-bt}], \quad a > 0, b > 0, \quad (9)$$

を得る。ここで、 $b = b_1 = b_2$ とした。式 (9) は、自動制御理論における 2 次おくれ要素によっても意味付けは可能である (図 1 参照)。

式 (9) から、瞬間エラー発見率は

$$\lambda(t) \equiv \lambda_M(t) = ab^2 t \cdot e^{-bt}, \quad (10)$$

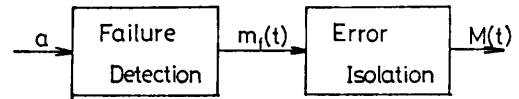
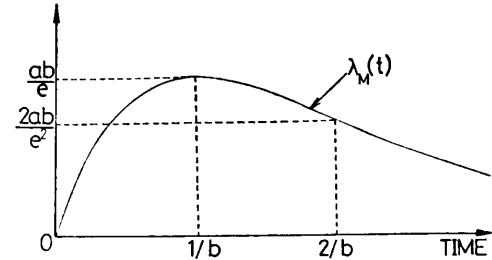
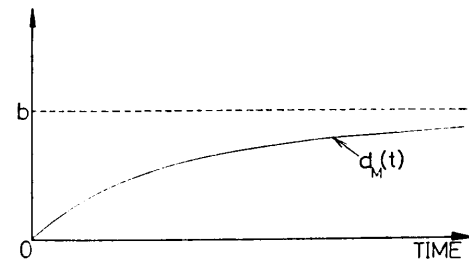


図 1 遅延 S 字形ソフトウェア信頼度成長モデル
Fig. 1 Delayed S-shaped software reliability growth model.



(a) Instantaneous error detection rate



(b) Error detection rate per error

図 2 エラー発見率 $\lambda_M(t)$ および $d_M(t)$
Fig. 2 Error detection rates $\lambda_M(t)$ and $d_M(t)$.

により与えられ、1 個当りのエラー発見率は

$$d(t) \equiv d_M(t) = b \left(1 - \frac{1}{1 + bt} \right), \quad (11)$$

となる。ここで、 $\lim_{t \rightarrow \infty} d_M(t) = b$ となることから、パラメータ b は定常状態における 1 個当りのエラー発見率を意味する。式 (10) および式 (11) を図示したのが図 2 である。瞬間エラー発見率 $\lambda_M(t)$ は単峰関数であり、 $t = 1/b$ すなわち平均値関数 $M(t)$ の変曲点において最大となる。また 1 個当りのエラー発見率 $d_M(t)$ は、試験時間 t について非減少関数であることから、 $M(t)$ は IEDR 関数であることがわかる。

3.2 習熟 S 字形ソフトウェア信頼度成長モデル

このモデルは、試験チームのソフトウェアに対する習熟度を考慮して、ソフトウェア故障の発見事象を記述するために開発されたものである。この現象をモデル化するために、試験により発見されるソフトウェアエラーには、他のエラーとは独立な発見の容易なものと、独立ではなく発見にはその前に他のエラーの発見を必要とするものとが混在すると仮定している。した

がって、ソフトウェア故障の発見能力は、新しくエラーを発見すればするほど向上することになる。このようなエラー発見に関する独立性は、ソフトウェア構造とプログラム・パスにより説明することができる (Ohba¹²⁾).

本モデルでは、以上のことからソフトウェア故障発見率は、試験時刻 t における残存エラー数に比例し、1 回当りのソフトウェア故障の発見能力に対応する比例項は試験の進行とともに増加するものとする。そこで、 $I(t)$ を時刻 t までに発見された総期待ソフトウェア故障数とすると

$$\frac{dI(t)}{dt} = b \left\{ r + (1-r) \frac{I(t)}{a} \right\} [a - I(t)], \quad (12)$$

の微分方程式を得る。ここで、パラメータ a を試験前にソフトウェア内に潜在する総期待エラー数、パラメータ b をソフトウェア故障発見率、パラメータ r をエラー発見能力に関する習熟係数とする ($0 < r \leq 1$)。式(12)から、ソフトウェア故障の発見能力は、 $I(t)$ の関数として

$$D(I(t)) = b \left\{ r + (1-r) \frac{I(t)}{a} \right\}, \quad (13)$$

となり、これを図3に示した。 $r=1$ のときは発見の容易な独立なエラーのみが潜在し、 $r=0$ のときは潜在するエラーが木 (tree) 状に順序づけられている場合や試験への投入労力を少しずつ増加させている場合に対応している。式(12)の微分方程式を解くと、NHPP の平均値関数として

$$I(t) = \frac{a[1 - e^{-bt}]}{1 + c \cdot e^{-bt}}, \quad a > 0, b > 0, c > 0, \quad (14)$$

を得る。ここで、 $c \equiv (1-r)/r$ とした。

式(14)から、瞬間エラー発見率は

$$\lambda(t) \equiv \lambda_1(t) = \frac{ab(1+c)e^{-bt}}{(1+c \cdot e^{-bt})^2}, \quad (15)$$

により与えられ、1 個当りのエラー発見率は

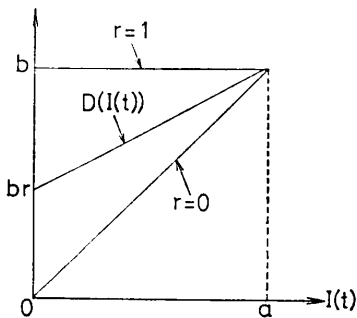
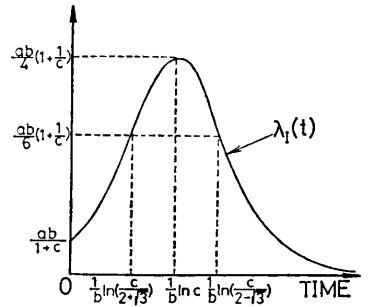
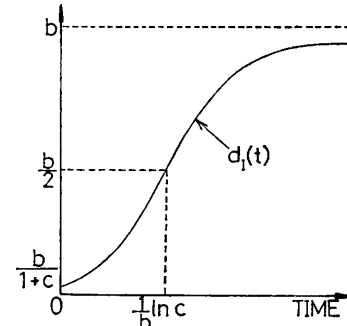


図3 ソフトウェア故障の発見能力 $D(I(t))$
Fig. 3 Detectability of software failures $D(I(t))$.



(a) Instantaneous error detection rate



(b) Error detection rate per error

図4 エラー発見率 $\lambda_1(t)$ および $d_1(t)$

Fig. 4 Error detection rates $\lambda_1(t)$ and $d_1(t)$.

$$d(t) \equiv d_1(t) = \frac{b}{(1+c \cdot e^{-bt})}, \quad (16)$$

となる。式(15)および式(16)を図示したのが図4である。瞬間エラー発見率 $\lambda_1(t)$ は単峰関数であり、 $t = \ln c/b$ すなわち平均値関数 $I(t)$ の変曲点において最大となる。さらに、 $\lambda_1(t)$ の変曲点は2個あり ($t \geq 0$)、 $t = \ln c/b$ に関して左右対称の位置にある。また、1 個当りのエラー発見率 $d_1(t)$ は、試験時間 t について非減少関数であることから、 $I(t)$ は IEDR 関数であることがわかる。

4. パラメータの推定

本章では、NHPP に基づくソフトウェア信頼度成長モデルを実際データに適用してデータ解析を行うときに、基本となるモデル・パラメータの最尤推定法について議論する。

試験時間区間 $(0, t_k]$ ($k=1, 2, \dots, n$; $0 < t_1 < t_2 < \dots < t_n$) において発見された総エラー数が y_k であるような n 組のソフトウェアエラーデータが観測されたものとする。このとき、式(1)および式(2)により定義される平均値関数 $a(t)$ をもつ NHPP モデルに対して、 $\{N(t_1)=y_1, N(t_2)=y_2, \dots, N(t_n)=y_n\}$ の同時確率関数すなわち尤度関数 (likelihood function) は、

$$L \equiv P, \{N(t_1)=y_1, N(t_2)=y_2, \dots, N(t_n)=y_n\}$$

$$= \prod_{k=1}^n \frac{\{a(t_k)-a(t_{k-1})\}^{y_k-y_{k-1}}}{(y_k-y_{k-1})!}$$

$$\times \exp[-\{a(t_k)-a(t_{k-1})\}], \quad (17)$$

により与えられる。ここで、 $t_0 \equiv 0$ および $y_0 \equiv 0$ とする。式(17)の尤度関数の自然対数をとると

$$\ln L = \sum_{k=1}^n (y_k - y_{k-1}) \ln [a(t_k) - a(t_{k-1})]$$

$$- a(t_n) - \sum_{k=1}^n \ln [(y_k - y_{k-1})!], \quad (18)$$

を得る。

したがって、遅延S字形ソフトウェア信頼度成長モデルでは $a(t) \equiv M(t)$ とおいて、モデル・パラメータ (a, b) に関する偏微分を $\partial \ln L / \partial a = \partial \ln L / \partial b = 0$ とし、習熟S字形ソフトウェア信頼度成長モデルでは $a(t) \equiv I(t)$ とおいて、モデル・パラメータ (a, b, c) に関する偏微分を $\partial \ln L / \partial a = \partial \ln L / \partial b = \partial \ln L / \partial c = 0$ として、各モデルに対する尤度方程式を導出することができる。これらの同時尤度方程式を数値的に解けば、各モデルのパラメータの最尤推定値 (\hat{a}, \hat{b}) あるいは $(\hat{a}, \hat{b}, \hat{c})$ を求めることができる。

5. 実際の適用例における考察

今までに議論してきた NHPP に基づくS字形ソフトウェア信頼度成長モデルのエラー発見率に基づく特性を、実際の試験で観測されたソフトウェアエラーデータに対する各モデルの適用例において考察する。本章で用いたデータは、次の DS1 および DS2 という二つのデータセットである：

DS1: オンライン・データエントリ・ソフトウェアパッケージの試験データ (Ohba¹²⁾),

DS2: RADC プロジェクト2の試験データ (Brooks and Motley¹³⁾).

まず、オンライン・データエントリ・ソフトウェアパッケージの試験において観測された、21組のソフトウェアエラーデータ DS1 (t_k, y_k) ($k=1, 2, \dots, 21$) を解析する。試験対象のプログラムは、アセンブリ言語により書かれた約 40,000 LOC (コード行) から成り、試験時間はソフトウェアエラーの原因解析の時間を含んでいる。そこで、遅延S字形ソフトウェア信頼度成長モデルを適用すると、第4章で述べた最尤法によりモデル・パラメータの最尤推定値 $\hat{a}=71.7$ および $\hat{b}=0.104$ を得る。したがって、平均値関数 $M(t)$ の最尤推定値は

$$\hat{M}(t) = 71.7 [1 - (1 + 0.104 t)e^{-0.104 t}], \quad (19)$$

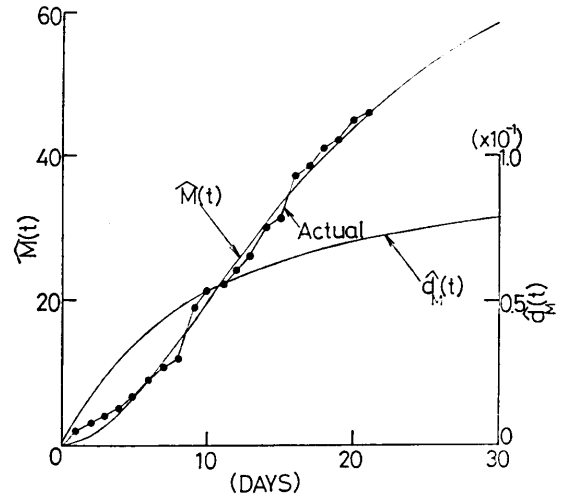


図5 ソフトウェアエラーデータ DS1 に対する推定値 $\hat{M}(t)$ および $\hat{d}_M(t)$
 Fig. 5 Estimated $\hat{M}(t)$ and $\hat{d}_M(t)$ for software error data DS1.

となり、これを実際データとともに図5に示した。式(19)の平均値関数をもつ NHPP モデルの観測データに対する適合性は、Kolmogorov-Smirnov 適合度検定 (Yamada and Osaki¹⁴⁾ 参照) により確認した。さらに、この試験以降のエラー発生状況から、ソフトウェア内に潜在的総エラー数の真値は約 69 であると報告されており、上記の推定結果は非常に精度の高いものとなっている。

上記の推定結果を用いて、図5には式(11)の1個当りのエラー発見率 $d_M(t)$ の推定値

$$\hat{d}_M(t) = (0.104)^2 t / (1 + 0.104 t), \quad (20)$$

も示した。遅延S字形ソフトウェア信頼度成長モデルは、試験の進行とともに、ソフトウェア内に残存するエラーの発見される可能性が増加するという、IEDR 関数の特性をもつことがわかる。また図6には、式(10)の瞬間エラー発見率 $\lambda_M(t)$ の推定値

$$\hat{\lambda}_M(t) = (71.7)(0.104)^2 t e^{-0.104 t}, \quad (21)$$

と、その実現値 $(y_k - y_{k-1}) / (t_k - t_{k-1})$ ($k=1, 2, \dots, 21$) を示した。推定値によれば、試験時刻 $t=1/\hat{b}=9.62$ (日)において瞬間エラー発見率は最大となっているが、これはほぼ実現値と一致している。

次に、Brooks and Motley¹³⁾ により引用された、35組のソフトウェアエラーデータ DS2 (t_k, y_k) ($k=1, 2, \dots, 35$) を解析する。試験対象のプログラムは、高水準言語 JOVIAL/J3 により書かれた約 124,000 LOC から成り、データはソフトウェア故障の発生現象の観測により得られたものである。そこで、習熟S字形ソフ

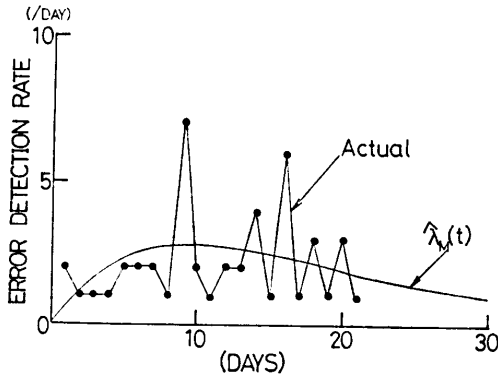


図 6 瞬間エラー発見率の推定値 $\lambda_M(t)$
Fig. 6 Estimated instantaneous error detection rate $\lambda_M(t)$.

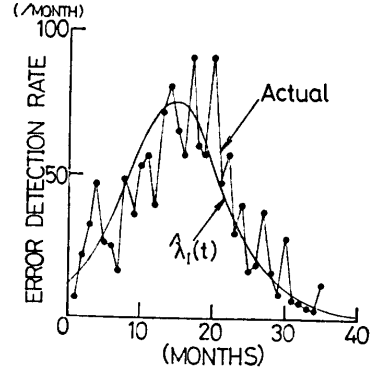


図 8 瞬間エラー発見率の推定値 $\lambda_I(t)$
Fig. 8 Estimated instantaneous error detection rate $\lambda_I(t)$.

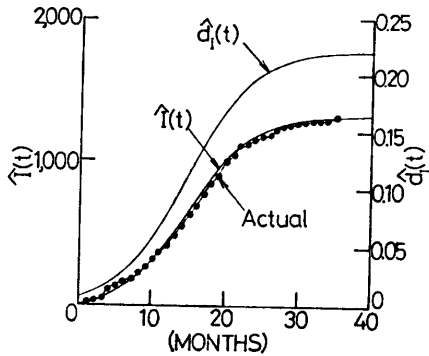


図 7 ソフトウェアエラーデータ DS 2 に対する推定値 $\hat{I}(t)$ および $\hat{d}_I(t)$
Fig. 7 Estimated $\hat{I}(t)$ and $\hat{d}_I(t)$ for software error data DS 2.

ソフトウェア信頼度成長モデルを適用して、最尤法によりモデル・パラメータの推定値 $\hat{a}=1315.9$, $\hat{b}=0.221$, および $\hat{c}=25.3$ を得る。したがって、平均値関数 $I(t)$ の最尤推定値は $\hat{I}(t)=1315.9(1-e^{-0.221t})/[1+(25.3)e^{-0.221t}]$, (22)

となり、これを実際データとともに図 7 に示した。ソフトウェアエラーデータ DS 1 と同様に、式 (22) の平均値関数をもつ NHPP モデルの適合性は Kolmogorov-Smirnov 適合度検定により確認した。式 (16) の 1 個当りのエラー発見率 $d_I(t)$ の推定値

$$\hat{d}_I(t)=0.221/[1+(25.3)e^{-0.221t}], \quad (23)$$

も図 7 に示した。習熟 S 字形ソフトウェア信頼度成長モデルも、IEDR 関数の特性をもつことがわかる。また図 8 には、式 (15) の瞬間エラー発見率 $\lambda_I(t)$ の推定値

$$\hat{\lambda}_I(t)=(1315.9)(0.221)(1+25.3)e^{-0.221t}/$$

$$[1+(25.3)e^{-0.221t}]^2, \quad (24)$$

とその実現値 $(y_k - y_{k-1})/(t_k - t_{k-1})$ ($k=1, 2, \dots, 35$) を示した。推定値によると、試験時刻 $t = \ln \hat{c}/\hat{b} = 14.6$ (月) において最大となっているが、実現値でも同傾向を示している。

ソフトウェアエラーデータ DS 1 および DS 2 に対して、各 S 字形ソフトウェア信頼度成長モデルの適用結果を上述べた。表 1 には、従来からよく用いられている式 (6) の平均値関数をもつ Goel and Okumoto⁴⁾ の指数形ソフトウェア信頼度成長モデルを含めて、三つのモデルによる適用結果をすべて示した。ここで、モデルの試験期間内の観測データに対する適合性を比較するために、観測値と推定値との偏差 2 乗和

表 1 ソフトウェアエラーデータ解析結果
Table 1 The results of software error data analysis.

Data	Model	Estimated model parameters	Sum of squares (SSE)
DS 1	NHPP model with $M(t)$	$\hat{a}=71.7$ $\hat{b}=0.104$	33.89
	NHPP model with $I(t)$	$\hat{a}=52.2$ $\hat{b}=0.206$ $\hat{c}=9.0$	81.88
	NHPP model with $m(t)$	$\hat{a} \rightarrow \infty$ $\hat{b}=0$	∞
DS 2	NHPP model with $M(t)$	$\hat{a}=1478.1$ $\hat{b}=0.105$	1.48×10^5
	NHPP model with $I(t)$	$\hat{a}=1315.9$ $\hat{b}=0.221$ $\hat{c}=25.3$	2.25×10^4
	NHPP model with $m(t)$	$\hat{a}=2499.5$ $\hat{b}=0.021$	4.53×10^4

$$(SSE) = \sum_{k=1}^n (y_k - \hat{y}_k)^2$$

$$(y_k = \hat{M}(t_k), \hat{y}_k = \hat{I}(t_k), \text{または } \hat{y}_k = \hat{m}(t_k)),$$

$$(25)$$

も示した. この比較基準によれば, ソフトウェアエラーの認定過程をも反映した DS1 に対しては遅延S字形ソフトウェア信頼度成長モデルが, またソフトウェア故障の発見事象を観測して得られた DS2 に対しては習熟S字形ソフトウェア信頼度成長モデルが良く適合している.

以上のことから, 発見された総エラー数がS字形成長曲線を示すときには, S字形ソフトウェア信頼度成長モデルは適合性の良いことがわかる.

6. むすび

本論文では, ソフトウェアの信頼度成長を把握する指標の一つであるエラー発見率に基づき, NHPP により記述されるS字形ソフトウェア信頼度成長モデルの特性について考察した. エラー発見率としては, 単位試験時間当りに発見される期待エラー数を表す瞬間エラー発見率と, 任意の試験時刻における(残存エラー)1個当りのエラー発見率を考えた. その結果, 遅延S字形ソフトウェア信頼度成長モデルと習熟S字形ソフトウェア信頼度成長モデルの両モデルには, 次のような共通する特性があることがわかった.

- (1) 瞬間エラー発見率は, 平均値関数の変曲点において最大となる単峰関数である. すなわちこの時点で, 遅延S字形ソフトウェア信頼度成長モデルではソフトウェアエラーの認定作業に対して, また習熟S字形ソフトウェア信頼度成長モデルではソフトウェア故障の発見作業に対して, 効果的に試験労力が投入されることによりソフトウェアの信頼度成長が促進されることを意味している.
- (2) 1個当りのエラー発見率は, 試験の進行とともにソフトウェア内に残存するエラーに対する発見の可能性が増加するという, 増加型エラー発見率 (IEDR) 関数の性質をもつ. これは, 遅延S字形ソフトウェア信頼度成長モデルでは, 試験の進行とともにソフトウェア内におけるエラーの分布状態が明確化してくるので, ソフトウェアエラーの認定がしやすくなることを意味する. 一方, 習熟S字形ソフトウェア信頼度成長モデルでは, ソフトウェア故障の発見がはかどると発見可能なエラー数の割合が増加し, 試験労力の投入量が増加

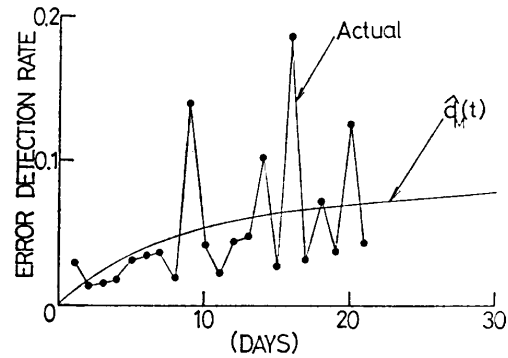


図9 1個当りのエラー発見率 $d_M(t)$ の推定値とその実現値

Fig. 9 Estimated and actual values of error detection rate per error, $d_M(t)$.

するにつれて試験チームのエラー発見能力が向上することを意味する.

これらは, 従来の研究⁹⁾⁻¹²⁾をさらに前進させる成果である. 特に(2)の1個当りのエラー発見率は, ソフトウェアの信頼度成長過程を観察する場合に有用である. これを, 第5章で用いたソフトウェアエラーデータ DS1 に対する遅延S字形ソフトウェア信頼度成長モデルの適用例において考察する.

このデータについては, ソフトウェア内に潜在する総エラー数の真値がほぼ 69 ($\equiv A$) であることがわかっていて, したがって, 1個当りのエラー発見率 $d_M(t)$ の実現値に相当する $[(y_k - y_{k-1}) / (t_k - t_{k-1})] / (A - y_k)$ を試験時間に対してプロットすると図9を得る. 図9には, 図5に示した $d_M(t)$ の推定値も示した. 1個当りのエラー発見率の実現値も試験時間の経過に対して増加傾向を示していることから, これはIEDR関数の性質により把握できることがわかる.

本論文で考察した, 各S字形ソフトウェア信頼度成長モデルを特徴づけるエラー発見率は, ハードウェアの信頼性理論における故障分布に対する故障率 (failure rate) に対応するものであると考えられる (Barlow and Proschan¹⁵⁾). この指標は, システムの信頼性解析, 信頼性試験, 各種の保全性問題など, 広範囲にわたって構成要素やシステムの故障特性を把握するのに役立っている. したがって, この故障率と同様にソフトウェアエラー数の確率分布に対するエラー発見率が, ソフトウェアの信頼性評価の実用に供するように, さらに考察を深めていかなければならない.

謝辞 いつも有益なご指導をいただく広島大学工学部 教授 尾崎俊治博士に感謝の意を表します.

参 考 文 献

- 1) Ramamoorthy, C. V. and Bastani, F. B.: Software Reliability—Status and Perspectives, *IEEE Trans. Softw. Eng.*, Vol. SE-8, No. 4, pp. 354-371 (1982).
- 2) Ascher, H. and Feingold, H.: *Repairable Systems Reliability: Modeling, Inference, Misconceptions and Their Causes*, Marcel Dekker, Inc., New York (1984).
- 3) Yamada, S. and Osaki, S.: Software Reliability Growth Modeling: Models and Applications, *IEEE Trans. Softw. Eng.*, Vol. SE-11, No. 12, pp. 1431-1437 (1985).
- 4) Goel, A. L. and Okumoto, K.: Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures, *IEEE Trans. Reliab.*, Vol. SE-28, No. 3, pp. 206-211 (1979).
- 5) Littlewood, B.: Theories of Software Reliability: How Good Are They and How Can They Be Improved?, *IEEE Trans. Softw. Eng.*, Vol. SE-6, No. 5, pp. 489-500 (1980).
- 6) Musa, J. D.: The Measurement and Management of Software Reliability, *Proc. IEEE*, Vol. 68, No. 9, pp. 1131-1143 (1980).
- 7) 菅野文友: ソフトウェア・エンジニアリング, 日科技連出版社, 東京(1979).
- 8) 当麻喜弘: フォールトトレラント技術の新フロンティア, 信学技報, FTS 85-1 (1985).
- 9) 山田 茂, 尾崎俊治: ソフトウェアの信頼度成長モデルとその比較, 信学論 (D), Vol. J 65-D, No. 7, pp. 906-912 (1982).
- 10) Yamada, S., Ohba, M. and Osaki, S.: S-Shaped Reliability Growth Modeling for Software Error Detection, *IEEE Trans. Reliab.*, Vol. R-32, No. 5, pp. 475-478, 484 (1983).
- 11) 大場 充, 梶山昌之: 習熟S字形ソフトウェア信頼度成長モデル, 情処学ソフトウェア工学研資, 28-6 (1983).
- 12) Ohba, M.: Software Reliability Analysis Models, *IBM J. Res. Dev.*, Vol. 28, No. 4, pp. 428-443 (1984).
- 13) Brooks, W. D. and Motley, R. W.: Analysis of Discrete Software Reliability Models, Technical Report RADC-TR-80-84, Rome Air Development Center, New York (1980).
- 14) Yamada, S. and Osaki, S.: Reliability Growth Models for Hardware and Software Systems Based on Nonhomogeneous Poisson Processes: A Survey, *Microelectron. Reliab.*, Vol. 23, No. 1, pp. 91-112 (1983).
- 15) Barlow, R. E. and Proschan, F.: *Statistical Theory of Reliability and Life Testing: Probability Models*, Holt, Rinehart and Winston, Inc., New York (1975).

(昭和60年7月26日受付)

(昭和61年6月19日採録)



山田 茂 (正会員)

昭和27年生。昭和50年広島大学工学部経営工学科卒業。昭和52年同大学院修士課程修了。昭和52~55年、日本電装(株)品質保証部勤務。

昭和58年広島大学大学院博士課程修了。同年より岡山理科大学勤務、現在工学部電子工学科専任講師。工学博士。信頼性工学、ソフトウェアの信頼性などの研究に従事。著書「ソフトウェアの信頼性評価法」など。電子通信学会、日本OR学会各会員。



大場 充 (正会員)

昭和24年生。昭和46年青山学院大学理工学部経営工学科卒業。昭和48年同大学院修士課程修了。昭和49年より日本アイ・ビー・エム(株)に勤務、現在同社サイエンス・イン

スティテュート、ソフトウェア・エンジニアリング担当。ソフトウェア工学およびソフトウェア品質評価法の研究に興味をもつ。