

テストの初期習熟度を考慮したソフトウェア信頼度成長モデル

込山俊博 砂塚利彦

日本電気(株) ソフトウェア生産技術開発本部

本論文では、ソフトウェアエラーの成長曲線が指数型の場合にもS字型の場合にも適合の良いソフトウェア信頼度成長モデルを提案する。本モデルは非同時ポアソン過程(NHPP)として記述し、その平均値関数にテスト初期のエラー検出能力を表す初期習熟度パラメータを導入する。本モデルは従来から知られている指数型成長モデルと遅延S字型成長モデルの問題点を解決すると共に、両モデルを含むものである。モデルのパラメータの推定には最尤法を用いる。また、3つのモデルを指数型とS字型成長を示す2つの実データに対して適用し、偏差2乗和によりそれらの適合度を比較評価する。

A Software Reliability Growth Model
Incorporating Initial Capability for Testing

Toshihiro KOMIYAMA and Toshihiko SUNAZUKA
Software Engineering Development Laboratory, NEC Corporation
11-5, Shibaura 2-Chome, Minato-ku, Tokyo 108, Japan

This paper describes a software reliability growth model incorporating initial capability of test team. The model is assumed to be a nonhomogeneous Poisson process (NHPP). Some problems for exponential model and delayed S-shaped model are discussed. The model fits for both exponential and S-shaped type data. These models can be described as the special cases of this model. The parameter estimates can be obtained by maximum likelihood estimation. Finally, the goodness-of-fit based on the sum of the squares of the differences between the actual cumulative number of errors and the estimated value are presented. The model fits better than other two models.

1. はじめに

近年、コンピュータシステムの普及が急速に進み、社会的な役割が高まっている。例えば、銀行業務においては個人または法人口座の入出金情報の管理、医療分野では放射線医療機器の制御などに利用されており、コンピュータシステムの異常動作が財産や人命に危機を及ぼしかねない状況にある。現在このような危険性を払拭すべく、異常動作やダウンを起こさないコンピュータシステムの出現が待たれている。

コンピュータシステムをハードウェアとソフトウェアに分けて考えるとソフトウェアの障害による異常動作やダウンが圧倒的に多い。ハードウェアに関しては品質管理技術の確立、FA化技術の進歩等によって品質の高い製品が世に送り出されている。ソフトウェアに関してはこのような技術はまだ開発途上にある。

前述の問題は、一口で言うとソフトウェア品質の問題と言える。ソフトウェア品質を考えるに当たっては、その定義を明らかにし、その定義に基づいて品質を定量的に評価できるようにする必要がある。

品質の定義については、現在ISO (International Organization for Standardization)のTC97/SC7で検討され、規格原案が作成されている[8]。现阶段ではソフトウェア品質を図1に示す6つの特性でとらえることが検討部会の参加各国間でほぼ合意されており、信頼性はソフトウェアの重要な品質特性として位置付けられている(図1参照)。

このように品質の定義については国際的な場でもかなり検討が進んでおり、ISOでは図1の品質特性をさらにブレイクダウンした副特性ならびに副副特性の検討が行なわれている。

信頼性においては信頼度成長モデルが信頼性の定量的評価のための有効な尺度となっている。ソフトウェアに潜在するエラーが多いほど信頼性の定義という性能水準の維持は困難になる。信頼度成長モデルはソフトウェアのテスト時間の経過とともに増加する累積エラー数の時系列的な傾向を記述するモデルであり、確率・統計的技法を用いて任意時間の残存エラー数やソフトウェアの潜在エラー総数等を予測することができる。この予測値から信頼性を定量的に評価することができる。また、信頼度(特定システムが一定期間エラーを起こさない確率)の算出等にも利用できる。信頼性の有効な定量的評価尺度となることが分かる。

これまで本研究分野では様々な信頼度成長モデルが提案され、適用、評価されている。そこで、これまでに提案された指数型成長モデル、超指数型成長モデル(指数型成長モデルを修正したもので、S字型成長モデルを表1に示す。

さらに表1のモデル以外にも、山田、山根、尾崎[27]による、発見されたエラーのうちそのときの修正により完全に修正される割合Pを導入したエラー修正率を考慮したモデル

$$M(t) = (a/P) \cdot (1 - e^{-Pt})$$

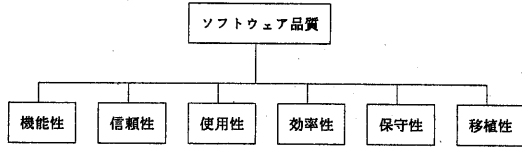
や、山田、大寺、成久[24]による、テスト労力を指数曲線 $w(t) = \alpha \beta \exp(-\beta t)$ や Rayleigh 曲線 $w(t) = \alpha \beta t \cdot \exp(-\beta t^2/2)$ により記述し、これらをモデルに組込んだモデル

$$m(t) = a(1 - \exp[-r(\alpha(1 - \exp[-\beta t]))])$$

または

$$m(t) = a(1 - \exp[-r(\alpha(1 - \exp[-\beta t^2/2]))])$$

や、山田[26]による、テスト労力をワイブル曲線 $w(t) = \alpha \beta m t^{\beta-1} \exp(-\beta t^m)$ により記述し、これを組



信頼性: 定められた期間、定められた条件下で、その性能水準を維持するための能力に関する属性

図1 ISOによるソフトウェア品質の構造と信頼性の定義

表1 ソフトウェア信頼度成長モデル一覧

区分	人名または名称	西 暦	モ デ ル 式	参 考 文 献
指 数 型 成 長	Z. Jelinski, P. B. Moranda	1972	$g(\tau_i) = \phi(N-1)$	[13]
	J. D. Musa	1975	$m(t) = N_0(1 - \exp(-\frac{Cr}{M_0 T_0}))$	[15][16][17]
	A. L. Goel, K. Okumoto	1979	$m(t) = a(1 - e^{-bt})$	[11]
	梶山昌之, 山田 茂, 尾崎俊治, 大場 光	1982	$\mu(t) = \begin{cases} 0 & 0 \leq t \leq e \\ a(1 - e^{-b(t-e)}) & t > e \end{cases}$	[5][19]
	山田 茂, 尾崎俊治	1982	$m_a(t) = a(e^{-bt} - e^{-at})$	[19][20]
	山田 茂, 尾崎俊治, 成久洋之	1983	$m_b(n) = a(1 - e^{-bn})$	[21]
	米原康弘, 大場 光	1983	$n(t) = \begin{cases} P_1 \cdot N \cdot (1 - e^{-\beta_1 t}) & : \text{機能テスト中} \\ P_2 \cdot (1 - P_1) N (1 - e^{-\beta_2 t}) & : \text{システムテスト中} \end{cases}$	[30]
	山田 茂, 尾崎俊治, 成久洋之	1983	$m(n) = a(1 - (1-b)^n)$	[22]
	米原康弘, 梶山昌之, 大場 光	1982	$n(t) = \sum_{i=1}^n N_i (1 - e^{-\beta_i t})$	[4][29]
	山田 茂, 大寺浩志, 成久洋之	1984	$m_p(t) = \sum_{i=1}^n P_i \cdot a(1 - e^{-\beta_i t})$	[23][24]
モ ス デ ル 型 成 長	北岡武司, 山田 茂, 尾崎俊治	1984	$C_p(n) = \sum_{i=1}^n P_i \cdot a(1 - (1-b_i)^n)$	[10]
	ロジスティックモデル	年代不明	$Y = K / (1 + m e^{-\alpha t})$	[9][12]
	ゴンベルツモデル	1825	$Y = K a^{-bt}$	[9][12]
成 長	山田 茂, 尾崎俊治	1981	$M(t) = a(1 - (1 + bt)e^{-bt})$	[18]
	梶山昌之, 大場 光	1982	$n(t) = N(1 - e^{-\beta t}) / (1 + \beta e^{-\beta t})$	[14][15]

込んだモデル

$$M(t) = a(1 - \exp[-r\alpha(1 - \exp[-\beta t^m])])$$

などが提案されている。

本論文では、まず、ソフトウェア開発の現場でよく用いられる代表的なGoel-Okumotoの指数型成長モデル[11]と山田、尾崎の遅延S字型成長モデル[18]の特徴と問題点を説明する。次に、パラメータの値によってS字型にも指数型にもなるモデルを提案し、最尤法を用いたパラメータの推定方法について述べる。さらに、本モデルを指数型、S字型傾向を持つそれぞれの実データに当てはめて、その適合度を指数型成長モデルおよび遅延S字型成長モデルと比較する。

2. Goel-Okumotoの指数型成長モデル

1972年、JelinskiとMorandaは"Software Reliability Research"[13]においてソフトウェアの信頼性についての議論を展開している。そこで提案されたエラー発見過程のモデルはエラー発見率がソフトウェアの残存エラー数に比例するという仮定から導き出されたものである。

$$g(\tau_i) = \phi(N-i)$$

i: 累積発見エラー個数

N: 潜在エラー総数

ϕ : 比例定数

τ_i : i個目のエラー発見からi+1個目発見までの時間

g(): エラー発見率

彼等はソフトウェアエラーの発見がポアソン過程に従うと仮定すると最尤法によりNと ϕ を推定できることを示している。

その後、1979年にGoelとOkumoto[11]はJelinski-Morandaのモデルを非同次ポアソン過程(NHPP: NonHomogeneous Poisson Process)として解析した指数型成長モデルを提案している。

このモデルではソフトウェアのエラー発見過程が平均値関数m(t)を持つNHPPであると仮定している。

ここで{N(t): t ≥ 0}を時刻tまでに発見される累積エラー数を表す計数過程であるとすると時刻tにおいてn個のエラーが発見される確率は

$$\Pr\{N(t)=n\} = \{m(t)\}^n \cdot \exp(-m(t))/n! \quad (1)$$

となる。

また、モデルを特徴付ける平均値関数m(t)はエラー発見率が残存エラー数に比例するという仮定から次式が導かれる。

$$m(t) = a(1 - e^{-bt}) \quad (2)$$

a: 期待潜在エラー総数

b: エラー1個あたりの発見率

このようにGoel-Okumotoの指数型成長モデルは上記(1)、(2)によって記述される。

また、(2)式からこのモデルのエラー発見率m'(t)とエラー1個あたりの発見率m'(t)/(a-m(t))を算出すると次のようになる。

$$m'(t) = abe^{-bt} \quad (3)$$

$$m'(t)/(a-m(t)) = b \quad (4)$$

さらに、(2)、(3)、(4)式を図示すると図2、3、4のようになる。

このモデルは次のような場合に適合がよいとされている。

- 1) エラーを現象発見として捉える
- 2) 現象と原因であるエラーが1対1に対応している
- 3) テスト投入労力が常に一定である
- 4) エラー1個あたりの発見率が一定である

また、一般にこのモデルは大規模システムの開発におけるエラーの成長曲線に当てはまることがよく知られている。

このモデルではエラー1個あたりの発見率がテストの最初から最後まで一定(図4参照)である。テスト開始時点ですでにエラー検出能力が定常状態になり、これが常に維持されるという仮定には無理がある。よほど経験を積み、管理されたテストチームでないと適合しづらいモデルである。

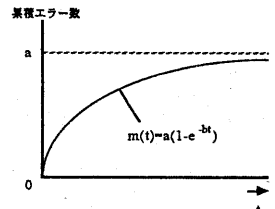


図2 指数型成長モデルの平均値関数

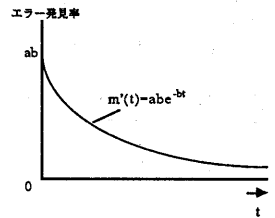


図3 指数型成長モデルのエラー発見率

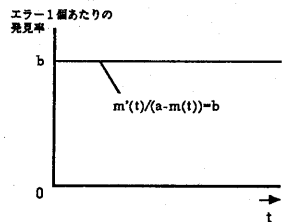


図4 指数型成長モデルのエラー1個あたりの発見率

3. 遅延S字型成長モデル

1981年、山田と尾崎[18]は中小規模システム開発における累積エラーの成長曲線が指数型よりもS字型の傾向を示すことが多いことから次の(1)'、(2)'で表される遅延S字型成長モデルを提案した。

このモデルにおいてもソフトウェアのエラー発見過程が平均値関数 $M(t)$ を持つNHPPであると仮定している。よって、 $\{N_s(t): t \geq 0\}$ を時刻 t までに発見される累積エラー数を表す計数過程であるとするとき時刻 t において n 個のエラーが発見される確率は

$$\Pr\{N_s(t)=n\} = \{M(t)\}^n \cdot \exp(-M(t)) / n! \quad (1)'$$

となる。

また、平均値関数 $M(t)$ は

$$M(t) = a(1 - (1+bt)e^{-bt}) \quad (2)'$$

a : 期待潜在エラー総数

b : 定常状態($t \rightarrow \infty$)におけるエラー1個あたりの発見率

である。

また、(2)'式からこのモデルのエラー発見率 $M'(t)$ とエラー1個あたりの発見率 $M'(t)/(a-M(t))$ を算出すると次のようになる。

$$M'(t) = ab^2te^{-bt} \quad (3)'$$

$$M'(t)/(a-M(t)) = b^2t/(1+bt) \quad (4)'$$

さらに、(2)'、(3)'、(4)'式を図示すると図5、6、7のようになる。

前述したように、中小規模システム開発における累積エラーの成長曲線はS字型を示すことが多いとされている。その原因として以下のような理由があげられる。

- 1) テスト開始直後はテスト担当者がプログラムやテスト環境に習熟していないため有効なテストを実施できない。
- 2) テスト中の投入労力が一定でない。
- 3) テスト担当者が障害らしき現象を発見してからプログラムに含まれるエラーを確認するまでに再試行等、現象を再現させるための努力をするためテスト全体に遅れを生じる(現象を引き起こした個々の原因をエラーとして捉えている)。

遅延S字型成長モデルは上記3)の理由によってS字型を示す場合を考慮して提案されたものである。

しかし、このモデルにおけるエラー発見率がテスト開始時点で0(図6参照)という仮定には無理がある。テスト担当者には多少なりともソフトウェア(テスト)に関する知識があることやテスト開始時点では発見が容易なエラーが多いことを考えるとテスト開始時点においてもエラーは発見できると考えるのが妥当であり、この仮定ではテスト担当者の能力を過小評価している。同様に、残存エラーの発見難易性を示すエラー1個あたりの発見率もテスト開始時点で0(図7参照)という仮定には無理があり、開始時点である値を持ち、その後徐々に大きくなると考えるのが妥当である。

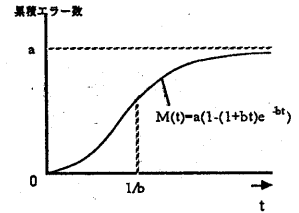


図5 S字型成長モデルの平均値関数

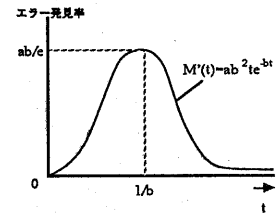


図6 S字型成長モデルのエラー発見率

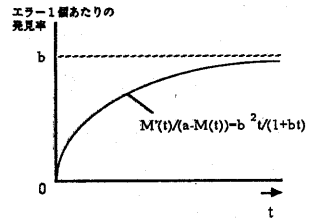


図7 S字型成長モデルのエラー1個あたりの発見率

4. 提案するモデル

図2、3、4および図5、6、7からも分るように指数型成長モデルと遅延S字型成長モデルの特徴は大きく異なっている。多くの場合、エラー発見過程はこれらのモデルを用いてある程度記述できる。しかし、これらのモデルの仮定を現実のテスト作業と比較した場合にそれぞれいくつかの問題点があることは2章、3章で述べた通りである。

本論文では従来のモデルの長所を生かすと共に、これらモデルの問題点を解決するモデルを提案する。

テストチームのエラー検出能力という点で、指数型では高水準の維持が現実的には難しいし、遅延S字型では初期時点で甘く見積もりすぎていると考えられる。そこでこの両極端の中間をカバーするモデルの特徴を検討する。

1) エラー発見率

テスト開始時点で正の実数を取り、単調増加後に極大値を取り単調減少を示すか、あるいは初めから単調減少を示すようにする(図8、9参照)。前者はより遅延S字型に近い特性であり、後者はS字型に近い特性である。

2) エラー1個あたりの発見率

テスト開始時点で正の実数を取り、単調増加を示すようにする(図10参照)。初期値と収束値との差が大きいと遅延S字型に近く、差が小さいと指数型に近くなる。

このようなモデルは遅延S字型成長モデルに初期値補正パラメータ t_0 を組入れることによって実現できる。 t_0 は遅延S字型成長モデルの原点を時間 t_0 だけ平行移動するパラメータであり、提案するモデルは t_0 を原点とする図11のようなモデルとなる。ここで t_0 を遅延S字型成長モデルの変曲点($1/b$:後述)より小さくした場合にはエラー発見率とエラー1個あたりの発見率が、それぞれ図8、図10のようなモデルとなり、変曲点より大きくした場合には図9、図10のようなモデルとなる。

次に、本モデルのモデル式を求める。

本モデルにおいてもソフトウェアのエラー発見過程が平均値関数 $Y(t)$ を持つNHPPであると仮定する。よって、 $\{N_v(t): t \geq 0\}$ を時刻 t までに発見される累積エラー数を表す計数過程であるとすると時刻 t において n 個のエラーが発見される確率は

$$\Pr\{N_v(t)=n\} = \{Y(t)\}^n \cdot \exp(-Y(t))/n! \quad (1)''$$

となる。

平均値関数 $Y(t)$ を求める。

移動した原点の時刻を t_0 とし期待潜在エラー総数が a となるように補正すると、遅延S字型成長モデル式 $M(t)$ を用いて以下のような関係式が導かれる。

$$Y(t) = \{a/(a-M(t_0))\} \cdot \{M(t+t_0)-M(t_0)\}$$

この式に $M(t)$ を代入し、 $l=t_0$ とすると

$$Y(t) = a \{1 - (1 + (bt/(1+bl))) \cdot e^{-bt}\} \quad (2)''$$

- a: 期待潜在エラー総数
- b: 定常状態($t=\infty$)におけるエラー1個あたりの発見率
- l: テストの初期習熟度パラメータ
($=t_0$: 初期値補正パラメータ)

を得る。

本モデルのパラメータ $l=0(t_0=0)$ とすると遅延S字型成長モデルになることは(2)''式より明らかである。

また、本モデルのエラー発見率 $Y'(t)$ とエラー1個あたりの発見率 $Y'(t)/(a-Y(t))$ を算出すると次のようになる。

$$Y'(t) = ab^2 e^{-bt} \cdot (t+1)/(1+bl) \quad (3)''$$

$$l=0 \text{ のとき } Y'(t) = ab^2 t e^{-bt} = M'(t)$$

$$l=\infty \text{ のとき } Y'(t) = abe^{-bt} = m'(t)$$

$$Y'(t)/(a-Y(t)) = b^2(t+1)/(1+bt+bl) \quad (4)''$$

$$l=0 \text{ のとき } Y'(t)/(a-Y(t)) = b^2 t/(1+bt) = M'(t)/(a-M(t))$$

$$l=\infty \text{ のとき } Y'(t)/(a-Y(t)) = b = m'(t)/(a-m(t))$$

以上より遅延S字型($l=0$)および指数型成長モデル($l=\infty$)を包含しているときとみなせる。

また、本モデルがS字型成長曲線となり変曲点を持つ条件を求める。(2)''式を2階微分することより $Y''(t) = ab^2((1-bt-bl)/(1+bl)) \exp(-bt)$ を得る。 $Y''(t)=0$ より $t=(1/b)-1$ となり、このような正値 t が存在することが変曲点を持つ条件であるから $l < 1/b$ がその条件である。よって、 $l < 1/b$ であれば、遅延S字型成長モデルに近いモデルとなり、 $l > 1/b$ であれば指数型成長モデルに近いモデルとなる。

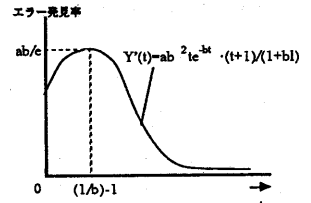


図8 新モデルのエラー発見率(1)

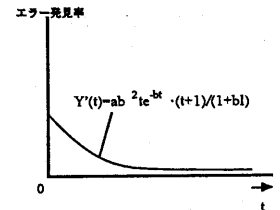


図9 新モデルのエラー発見率(2)

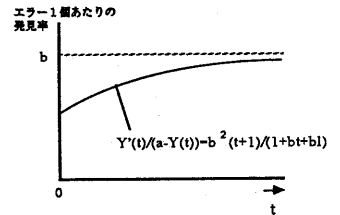


図10 新モデルのエラー1個あたりの発見率

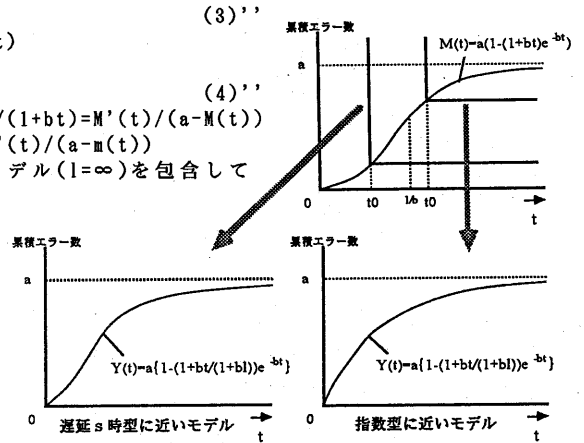


図11 遅延S字型成長モデルから新モデルへの変換

5. パラメータの推定

本論文で提案したモデルの3つのパラメータ(a, b, l)の推定方法を示す。

ここで、あるソフトウェアに対してテストを実施し、時刻 t_i までに発見されたエラーが n_i 個であるような一連のデータ $(t_i, n_i) (i=0, 1, 2, \dots, k)$ が得られたものとする。ソフトウェアエラーの発見過程が $Y(t)$ を平均値関数とするNHPPに従うと仮定すると、時刻 t_i において n_i 個のエラーが発見される確率は

$$\Pr\{Y(t_i)=n_i\} = \{Y(t_i)\}^{n_i} \cdot \exp(-Y(t_i))/n_i! \quad (5)$$

となる。

(5)式は任意の時間間隔 $(t_i - t_{i-1})$ における発見エラー数 $(n_i - n_{i-1})$ についても成り立ち

$$\Pr\{Y(t_i) - Y(t_{i-1}) = n_i - n_{i-1}\} = \frac{\{Y(t_i) - Y(t_{i-1})\}^{n_i - n_{i-1}}}{\{n_i - n_{i-1}\}!} \exp[-(Y(t_i) - Y(t_{i-1}))] \quad (6)$$

となる。

よって、 $\{Y(t_1) = n_1, Y(t_2) = n_2, \dots, Y(t_k) = n_k\}$ の同時確率は

$$\Pr\{Y(t_1) = n_1, Y(t_2) = n_2, \dots, Y(t_k) = n_k\} \\ = \exp(-Y(t_k)) \prod_{i=1}^k \frac{\{Y(t_i) - Y(t_{i-1})\}^{n_i - n_{i-1}}}{\{n_i - n_{i-1}\}!} \quad (7)$$

となる。

式(7)より対数尤度関数 $L(t, y; a, b, l)$ は

$$L(t, y; a, b, l) \\ = \sum_{i=1}^k (n_i - n_{i-1}) \cdot \ln \left\{ a \left(1 - \left(1 + \frac{bt_i}{1+bl} \right) \cdot \exp(-bt_i) \right) - a \left(1 - \left(1 + \frac{bt_{i-1}}{1+bl} \right) \cdot \exp(-bt_{i-1}) \right) \right\} \\ - \sum_{i=1}^k \{ (n_i - n_{i-1})! \} - a \left(1 - \left(1 + \frac{bt_k}{1+bl} \right) \cdot \exp(-bt_k) \right) \quad (8)$$

となる。

最尤推定値は、本式をそれぞれのパラメータ a, b, l について微分した $\delta L / \delta a, \delta L / \delta b, \delta L / \delta l$ を0とした以下の連立方程式を数値的に解くことで得られる。
 $\delta L / \delta a = 0$ より

$$a = \frac{n_k}{1 - \left(1 + \frac{bt_k}{1+bl} \right) \cdot \exp(-bt_k)} \quad (9)$$

$\delta L / \delta b = 0$ より

$$a \left(t_k - \frac{t_k}{(1+bl)^2} + \frac{bt_k^2}{1+bl} \right) \exp(-bt_k) \\ = \sum_{i=1}^k (n_i - n_{i-1}) \frac{\left(t_i - \frac{t_i}{(1+bl)^2} + \frac{bt_i^2}{1+bl} \right) \exp(-bt_i) - \left(t_{i-1} - \frac{t_{i-1}}{(1+bl)^2} + \frac{bt_{i-1}^2}{1+bl} \right) \exp(-bt_{i-1})}{- \left(1 + \frac{bt_i}{1+bl} \right) \exp(-bt_i) + \left(1 + \frac{bt_{i-1}}{1+bl} \right) \exp(-bt_{i-1})} \quad (10)$$

$\delta L / \delta l = 0$ より

$$at_k \exp(-bt_k) = \sum_{i=1}^k (n_i - n_{i-1}) \frac{t_i \exp(-bt_i) - t_{i-1} \exp(-bt_{i-1})}{- \left(1 + \frac{bt_i}{1+bl} \right) \exp(-bt_i) + \left(1 + \frac{bt_{i-1}}{1+bl} \right) \exp(-bt_{i-1})} \quad (11)$$

6. モデルの適合度の評価

提案したモデル、指数型成長モデル、遅延S字型成長モデルを指数型傾向を持つデータとS字型傾向を持つデータにそれぞれ適用し、その偏差2乗和を比較することで適合度を評価する。

指数型傾向を持つデータ(D1)として、社内のOA系のアプリケーション開発部門の29組のソフトウェアエラーデータ (t_i, n_i) を用いる。本データではテスト実施日数を時間軸にとっている。

また、S字型傾向を持つデータ(D2)として、大場[4]の"Software Reliability Analysis Models"のPL/I Application Program Test Dataを用いる。本データは以下19組のソフトウェアエラーデータ (t_i, n_i) である。本データでは、観測した週を時間軸にとっている。

新モデルでは、3つのパラメータについての尤度方程式を解く必要があり、計算がかなり繁雑となる。そこで本論文では、1を与えたもとで a, b を算出し、そのときの偏差2乗和が極小となる近傍の1を最適解の代用とした(図12、13参照)。

表2、3に、D1、D2それぞれについて3つのモデルのパラメータの最適解および偏差2乗和を比較した表を掲載する。

図12、13において $l=0$ および ∞ のときの偏差2乗和は、それぞれ遅延S字型および指数型成長モデルのそれと一致する。従って、本モデルは遅延S字型および指数型成長モデル以上の性能があると考えられる。

表2 指数型データ (D1) に対する適合度の比較

	a	b	l	偏差2乗和
新モデル	53.84	0.1154	10	<u>82.60</u>
指数型成長モデル	57.15	0.06717	-	87.44
遅延S字型成長モデル	50.53	0.1844	-	188.11

偏差2乗和

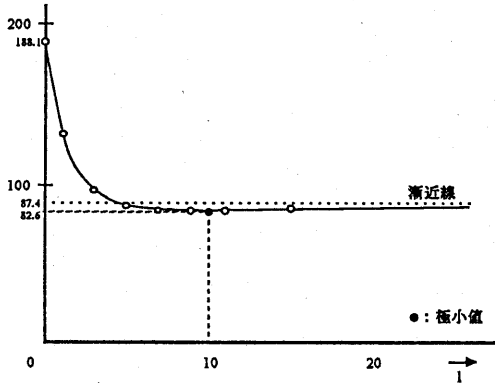


図12 l vs 偏差2乗和 (D1)

表3 S字型データ (D2) に対する適合度の比較

	a	b	l	偏差2乗和
新モデル	384.09	0.1650	2	<u>2492.74</u>
指数型成長モデル	513.15	0.05365	-	4219.79
遅延S字型成長モデル	359.92	0.2126	-	3588.19

偏差2乗和

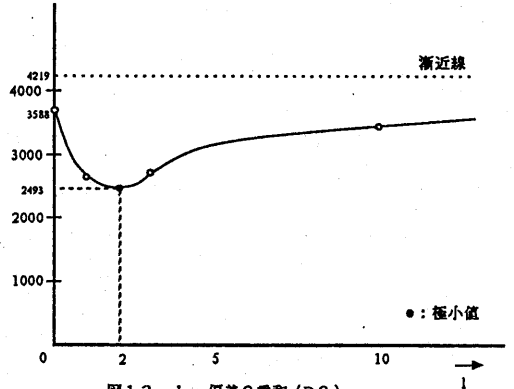


図13 l vs 偏差2乗和 (D2)

7. おわりに

本論文では、テスト初期習熟度パラメータ l を導入することにより、従来からよく用いられている Goel-Okumotoの指数型成長モデルと山田-尾崎の遅延S字型成長モデルの持つ問題点を解決し、両モデルを包含するモデルを実現し、その尤度方程式を示した。モデル式は平均値関数 $Y(t)=a\{1-(1+(bt/(1+bt))\cdot \exp(-bt))\}$ を持つNHPPモデルである。さらに、指数型傾向を持つデータとS字型傾向を持つそれぞれのデータについて、上記3つのモデルの適合度を偏差2乗和を用いて比較し、本モデルが双方のデータに対して従来のモデルより適合度の良いことを示した。

《参考文献》

- [1] M. Ohba, S. Yamada, K. Takeda and S. Ozaki, "S-shaped Software Reliability Growth Curve: How Good Is It?," Proc. of IEEE COMPSAC 82, pp.38-44, 1982
- [2] 大場充, "ソフトウェア信頼度成長モデル-ソフトウェア品質評価の道具として," 第13回日科技連 信頼性・保守性シンポジウム 8-L-2, pp.353-358, 1983
- [3] M. Ohba and S. Yamada, "S-shaped Software Reliability Growth Models," 4th International Conference on Reliability and Maintainability, pp.430-436, 1984
- [4] M. Ohba, "Software Reliability Analysis Models," IBM J. Res. Develop., Vol.28, No.4, pp.428-443, 1984
- [5] 梶山昌之, 山田茂, 尾崎俊治, 大場充, "ソフトウェア-エラー発見過程のモデル-ソフトウェアの信頼度成長曲線: その比較と問題点," 情処学会第24回全国大会講演論文集 7P-5, pp.435-436, 1982
- [6] 梶山昌之, 大場充, "テストの習熟を考慮したソフトウェア信頼度成長モデル," 情処学会第25回全国大会講演論文集 1E-3, pp.401-402, 1982
- [7] 梶山昌之, "習熟型ソフトウェア信頼度成長モデル," 情処学会第27回全国大会講演論文集 7B-6, pp.611-612, 1983
- [8] 菅忠義他編, "ソフトウェア開発の実際-標準とその活用," (財)日本規格協会, 1988
- [9] 菅野文友, "ソフトウェア品質予測における成長曲線のあてはめについて," 品質管理, Vol.26, 5月臨時増刊号, pp.190-194, 1975
- [10] 北岡武司, 山田茂, 尾崎俊治, "2種類のエラーを考慮した離散のソフトウェア信頼度成長モデル," 電子通信学会論文誌, Vol. J68-D, No. 6, pp.1242-1247, 1985

- [11] A.L. Goel and K. Okumoto, "Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures," IEEE Trans. Reliab., Vol. R-28, No. 3, pp. 206-211, 1979
- [12] 坂田一志, "ソフトウェアの生産管理における予測技法の定式化—静的な予測および故障率推移モデル," 電子通信学会論文誌, Vol. 57-D, No. 5, pp. 277-283, 1974
- [13] Z. Jelinski and P. Moranda, "Software Reliability Research," (Statistical Computer Performance Evaluation), W. Freiberger, Ed., Academic Press, Inc., NY, 1972
- [14] "マンマシンインタフェース及びソフトウェア評価の標準化に関する調査研究報告書," (社)日本機械工業連合(財)日本規格協会, 1988
- [15] J.D. Musa, "A Theory of Software Reliability and Its Application," IEEE Trans. Soft. Eng., Vol. SE-1, No. 3, pp. 312-327, 1975
- [16] J.D. Musa, "Validity of Execution-Time Theory of Software Reliability," IEEE Trans. Reliab., Vol. R-28, No. 3, pp. 181-191, 1979
- [17] J.D. Musa, "The Measurement and Management of Software Reliability," Proc. of IEEE, Vol. 68, No. 9, pp. 1131-1143, 1980
- [18] 山田茂, 尾崎俊治, "ソフトウェアエラー発見過程の確率モデルとその応用," ソフトウェア工学研究会報告 21-3, 1981
- [19] 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," 電子通信学会論文誌, Vol. J65-D, No. 7, pp. 906-912, 1982
- [20] 山田茂, 尾崎俊治, "ソフトウェア信頼性評価のための定量的方法," 電子通信学会論文誌, Vol. J65-D, No. 12, pp. 1483-1490, 1982
- [21] 山田茂, 尾崎俊治, 成久洋之, "離散的ソフトウェア信頼度成長モデルの比較," 日本OR学会'83秋季研究発表会77'ストラク集 D-3, pp. 145-146, 1983
- [22] 山田茂, 尾崎俊治, 成久洋之, "テストラン試行回数を考慮したソフトウェア信頼度成長モデルに関する考察," ソフトウェア工学研究会報告 31-1, 1983
- [23] S. Yamada, H. Narihisa and H. Ohtera, "Software Reliability Analysis Based on Nonhomogeneous Error Detection Rate Model," Microelectron. Reliab., Vol. R-24, No. 5, pp. 915-920, 1984
- [24] 山田茂, 大寺活志, 成久洋之, "エラー発見の難易度を考慮したソフトウェア信頼度成長モデルとその最適リリース政策に関する考察," 日本OR学会'85春季研究発表会77'ストラク集 1-C-2, pp. 46-47, 1985
- [25] 山田茂, 大寺活志, 成久洋之, "テスト労力を考慮したソフトウェア信頼度成長モデルとその応用," 電子通信学会技術研究報告(信頼性) R85-9, pp. 17-22, 1985
- [26] 山田茂, "ソフトウェアの信頼性評価法," ソフトウェア・リサーチ・センター, 1985
- [27] 山田茂, "テスト投入労力の記述とソフトウェア信頼度成長モデル," ソフトウェア工学研究会報告 50-5, 1986
- [28] 山田茂, 山根勉, 尾崎俊治, "エラー修正率を考慮したソフトウェア信頼度成長モデル," 情報処理学会論文誌, Vol. 27, No. 1, pp. 64-71, 1986
- [29] 米原康弘, 梶山昌之, 大場充, "モデル構造ソフトウェアの信頼度成長曲線," 情処学会第25回全国大会講演論文集 1E-2, pp. 399-400, 1982
- [30] 米原康弘, 大場充, "テストフェーズを考慮したソフトウェアの信頼度予測," 情処学会第27回全国大会講演論文集 7B-5, pp. 609-610, 1983