

2クラスの発見難易度をもつソフトウェアエラーを考慮した 指数-S字形信頼度成長モデルの考察†

木村 光 宏^{††} 山 田 茂^{†††} 尾 崎 俊 治^{†††}

開発過程あるいは運用段階にあるソフトウェアの信頼性を定量的に評価するために、非同次ポアソン過程に基づくソフトウェア信頼度成長モデルがこれまでに数多く提案されてきた。これらのモデルの多くは、ソフトウェア内に潜在しているエラーの発見・修正しやすさが、すべてのエラーについて同等であるものとみなしているものが多い。この仮定はモデルの単純化には都合が良いが、実際のエラー発見・修正過程に対しては必ずしも十分なものではない。例えばよく観測されるように、発見された累積エラー数の成長曲線が、テスト初期には指数形傾向を示し、それ以降S字形傾向を示す場合には、この仮定は不適切であり、新たにこの現象を記述するモデルを構築する必要がある。このことから本論文では、ソフトウェア内に潜在するエラーをその発見・修正が容易なものと、そうでないものとに区別し、それぞれのエラー発見事象を相異なる非同次ポアソン過程により記述する。また、それらの重ね合わせを用いて、実際に観測されるエラー発見事象をよりの確に記述する新しいモデルを、指数-S字形ソフトウェア信頼度成長モデルとして提案する。さらに、モデルパラメータの推定および信頼性評価の数値例を示し、既存のモデルとの適合性に関する比較を行う。

1. ま え が き

ソフトウェア開発管理者にとって、高信頼性ソフトウェアを効率良く生産することは非常に重要な課題であり、ソフトウェア工学における学問的課題でもある。したがって、開発中のソフトウェアがどのくらいの信頼性を具備しているかを定量的に知ることが、ソフトウェア製品の出荷時期のより正確な予測、開発労力の再分配や投入調整、テストにより除去できなかったソフトウェアエラーに対する保守費用の見積りなどを行う場合の一助となる。この立場から、ソフトウェア開発のテスト工程におけるソフトウェアエラー発見事象やソフトウェア故障発生現象に確率・統計論を導入し、非同次ポアソン過程 (nonhomogeneous Poisson process, 以下 NHPP と略す) により記述するソフトウェア信頼度成長モデル¹⁾⁻³⁾ が現在までに数多く提案されてきた。ここで、ソフトウェアエラー (以下エラーと略す) とはソフトウェア故障を引き起こすプログラム内部の論理的な誤りと定義し、ソフトウェア故障とはソフトウェアが期待どおりに動作しないことと定義する。これらのモデルでは、潜在エラー1個当りのエラー発見率はテスト時間の経過に伴って変

化しないという仮定の下で、あるいはエラー発見率がテスト時間に関係するようにモデル化されている。前者の仮定の場合は、ソフトウェア内に潜在するエラーはすべて同等に発見・修正される可能性をもつことと同義である。一方後者の仮定は、テスト工程におけるテスト方法、テスト要員、あるいはソフトウェア自体の特性などの要因から考えると、より現実的であろう。この仮定に対して、実際にソフトウェア内に作り込まれている潜在エラーには、発見・修正が容易なものと、そうでないものが混在していると考えてモデル化するのも1つの方法である。例えば、Yamada et al.⁴⁾は Goel and Okumoto⁵⁾の提案した指数形ソフトウェア信頼度成長モデルを拡張して、テストにより発見されるエラーを発見の容易なエラーと困難なエラーの2種類が混在するものと仮定して、それらの1個当りのエラー発見率は相異なるがテスト期間を通じて一定であるとする修正指数形ソフトウェア信頼度成長モデルを提案した。

本論文では、ソフトウェア内の潜在エラーをその発見・修正が容易にできるものと、そうでないものとに区別し、それぞれの相異なるエラー発見過程を各計数過程により記述する。これら2つの計数過程は、実際のテストにおいてもよく観測されるように⁶⁾、発見された累積エラー数の成長曲線がテスト初期段階では指数形傾向を示し、その後S字形傾向に変化していくという現象を説明するものである。さらに、それらを重ね合わせることで、そのようなエラー発見事象を記述するソフトウェア信頼度成長モデルを提案する。

† An Exponential-S-shaped Software Reliability Growth Model with Two Classes of Error Detectability by MITSUHIRO KIMURA, SHIGERU YAMADA and SHUNJI OSAKI (Department of Industrial and Systems Engineering, Faculty of Engineering, Hiroshima University).

†† 広島大学大学院工学研究科システム工学専攻

††† 広島大学工学部第二類 (電気系)

また、実際のテスト工程から得られたエラーデータを用いて数値例を示し、従来のモデルとの適合性に関する比較を行う。

2. モデルの記述

ソフトウェアの信頼性をテスト工程（あるいは運用段階）において定量的に評価するために、これまで数多くのソフトウェア信頼度成長モデルが提案されてきた。これらのモデルの多くは基本的に、エラー発見事象あるいはソフトウェア故障発生現象の観測期間に対して、そのテスト環境を説明する仮定を与え、1つの信頼度成長過程を大局的に記述するものであった。例えば、指数形ソフトウェア信頼度成長モデルまたはS字形ソフトウェア信頼度成長モデルは、テスト期間中の信頼度成長過程を、テストにより発見された累積エラー数の成長曲線により、すなわち1つの信頼度成長曲線（指数形またはS字形信頼度成長曲線）により表現するものである。しかし、実際のテスト工程から得られたテスト経過時間に対する累積発見エラー数の挙動は、図1に示すように信頼度成長曲線の傾向がテスト期間中に変化する場合（例えば岩佐⁶⁾参照）、すなわちテスト初期は指数形信頼度成長曲線、中期以降はS字形信頼度成長曲線とみなせることも多く、これらは大局的に見て相異なる信頼度成長過程と考えるほうが自然であろう。したがって、このような挙動を示すエラー発見・修正過程に対して、テスト初期で発見されるエラーと中期以降に発見されるエラーとは本質的に相異なる性質をもつものであるという解釈をする必要がある。このことは、テスト初期において発見されるエラーはその多くがソフトウェア故障の観測が直ちに

エラーの発見・除去につながるような比較的単純なエラーであり、一方テスト中期以降で多く発見されるエラーは、十分原因解析を行わなければエラーの再現ができず修正・除去できないような比較的複雑なエラーであるという意味づけが行えるであろう⁷⁾。以上のことから本論文では、テスト初期に発見されやすいエラー（以下、クラス1エラーと呼ぶ）と中期以降に多く発見されるエラー（以下、クラス2エラーと呼ぶ）の発見過程を、ソフトウェア信頼度成長モデルの立場からモデル化する。中川、竹中⁸⁾は、テスト工程で発見されるエラーの複雑度を3クラスに分類し、これらを考慮したソフトウェア信頼性モデルを提案している。

テスト時刻 t ($t \geq 0$) までに発見されるクラス1エラーの累積数を $M_1(t)$ 、クラス2エラーの累積数を $M_2(t)$ とする。テスト工程におけるエラー発見過程のもつ不確定要因に対して、 $M_i(t)$ ($i=1, 2$) は非負整数値を取る確率過程とみなし、以下のような性質を持つNHPPを仮定する⁹⁾。

$$(1) \quad M_i(0) = 0 \quad (i=1, 2),$$

$$(2) \quad \{M_i(t), t \geq 0\} \quad (i=1, 2) \text{ はそれぞれ独立増分 (independent increments) をもつ。}$$

$$(3) \quad \Pr[M_i(t+\Delta t) - M_i(t) = 1] = h_i(t)\Delta t + o(\Delta t) \quad (i=1, 2),$$

$$(4) \quad \Pr[M_i(t+\Delta t) - M_i(t) \geq 2] = o(\Delta t) \quad (i=1, 2).$$

ここで、 $\Pr[A]$ は事象Aの生起する確率を表す。これらの仮定のもとで、クラス1エラーおよびクラス2エラーの発見事象は、それぞれ

$$\left. \begin{aligned} \Pr[M_i(t) = m] &= \frac{\{H_i(t)\}^m}{m!} \exp[-H_i(t)] \\ (m=0, 1, 2, \dots) \end{aligned} \right\}, \quad (1)$$

$$H_i(t) = \int_0^t h_i(t') dt' \quad (i=1, 2)$$

となる確率分布により記述される。式(1)の平均値関数 (mean value function) と呼ばれる $H_i(t)$ ($i=1, 2$) は、 $M_i(t)$ ($i=1, 2$) の期待値であり、それぞれテスト時刻 t までに発見されるクラス1エラーおよびクラス2エラーの総期待数を表す。また、 $h_i(t)$ ($i=1, 2$) は強度関数 (intensity function) と呼ばれ、瞬間エラー発見率を表す。

2.1 クラス1エラーの発見過程

前述のように、クラス1エラーはソフトウェア故障の発生現象の観測が直ちにエラーの発見および修正に結びつくものである。したがって、式(1)の平均値関数 $H_1(t)$ に関して、残存エラー数に対するエラー発見

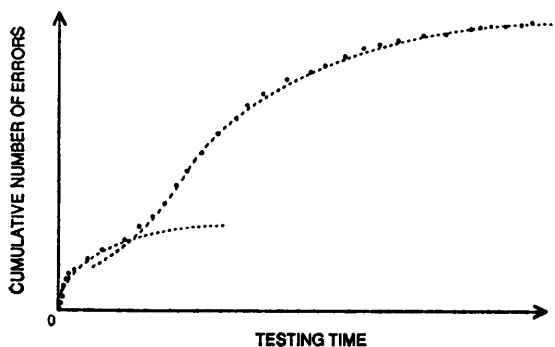


図1 実際に観測されるソフトウェア信頼度成長過程 (指数-S字形信頼度成長曲線)

Fig. 1 Actual software reliability growth process (exponential-S-shaped reliability growth curve).

率がテスト期間を通じて一定であるものと仮定すれば、

$$\frac{dH_1(t)}{dt} = b_1[a_1 - H_1(t)] \quad (a_1 > 0, b_1 > 0), \quad (2)$$

の関係式を得る。ここで、各パラメータは次の意味をもつ。

a_1 : テスト開始前におけるソフトウェア内に潜在するクラス1エラーの総期待数、

b_1 : 残存するエラー1個当りのエラー発見率、

式(2)を $H_1(t)$ について解くと、

$$H_1(t) = a_1(1 - \exp[-b_1 t]), \quad (3)$$

となる。これは、クラス1エラーの発見過程に対し、Goel and Okumoto⁵⁾の指数形ソフトウェア信頼度成長モデルを適用したことを意味する。

2.2 クラス2エラーの発見過程

比較的複雑なソフトウェアをテストする場合、ソフトウェア故障発生現象を観測した後、その原因解析を十分に行わなければソフトウェア故障現象の再現ができず、結果としてエラーの認知・修正が行えない場合がある。例えば、テスト入力だけでなくコンピュータの内部状態にも依存してソフトウェア故障が発生する場合などであり、中川、竹中⁶⁾によるエラー複雑度の分類におけるクラス2および3に相当するものと考えられる。そこで、クラス2エラーをそのような認知・修正が困難なエラーであると仮定する。山田、大場⁷⁾によれば、そのソフトウェア故障発生現象を観測する過程はソフトウェア故障発見 (failure detection) 過程と呼び、その原因解析を行ってエラーの発見・修正に至るまでの過程はエラー認知 (error isolation) 過程と呼んでいる。また木村⁸⁾は、ホスト適用業務システムのテスト工程におけるエラー発見事象が、このような2つの過程で説明でき、実際に以下で述べる遅延S字形ソフトウェア信頼度成長モデルにより信頼性評価を行っている。ここで、次のパラメータを導入する。

a_2 : テスト開始前にソフトウェア内に潜在するクラス2エラーの総期待数、

b_f : ソフトウェア故障発見率、

b_e : クラス2エラー認知率。

前述した仮定のもとで、テスト時刻 t までに発見される総期待ソフトウェア故障数を $H_f(t)$ とすると、

$$\frac{dH_f(t)}{dt} = b_f[a_2 - H_f(t)] \quad (a_2 > 0, b_f > 0), \quad (4)$$

が成立する。すなわち、テスト時刻 t におけるクラス2エラーに起因するソフトウェア故障の発生しやすさが、その時点でソフトウェア内に残存するクラス2エ

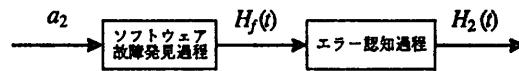


図2 クラス2エラーの発見過程
Fig. 2 A class 2 error detection process.

ラー数に比例することを表す。また、これに引き続きエラー認知過程において、時刻 t までに発見されるクラス2エラーの総期待数を $H_2(t)$ とおくと、

$$\frac{dH_2(t)}{dt} = b_e[H_f(t) - H_2(t)] \quad (b_e > 0), \quad (5)$$

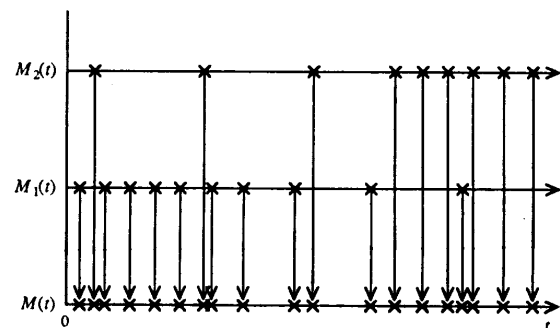
が成立する。したがって、近似的に $b_2 = b_f = b_e$ とし、式(4)および式(5)の連立微分方程式を $H_2(t)$ に関して解くと、

$$H_2(t) = a_2 \{1 - (1 + b_2 t) \exp[-b_2 t]\}, \quad (6)$$

を得る。これは、クラス2エラーの発見過程に対し、山田ら^{9), 7)}により提案された遅延S字形ソフトウェア信頼度成長モデルを適用したことを意味し、図2にクラス2エラーによるソフトウェア故障発生過程およびエラー認知過程を示す。また、式(4)および式(5)において $H_f(t) \equiv H_2(t)$ の場合、すなわちエラーの発見と認知が同時に行われる、あるいはエラーの認知時間が発見時間に比べて十分に短く無視できると仮定できる場合、クラス2エラーの平均値関数 $H_2(t)$ は式(3)と同じになる。このことは、文献7)においてもすでに示されているものである。

2.3 指数-S字形ソフトウェア信頼度成長モデル

2.1 および 2.2 節において、クラス1エラーとクラス2エラーの発見過程を、それぞれ相異なる平均値関数をもつ NHPP として記述した。2つのクラスのエラー発見事象を1つの確率過程で取り扱うために、これらを混合した新たな計数過程 $\{M(t), t \geq 0\}$ を考える。図3に示すように、 $\{M(t), t \geq 0\}$ は2つの NHPP



(X; error detection or software failure occurrence)

図3 計数過程 $\{M(t), t \geq 0\}$
Fig. 3 Counting process $\{M(t), t \geq 0\}$.

である $\{M_1(t), t \geq 0\}$ および $\{M_2(t), t \geq 0\}$ の重ね合わせ (superposition)¹⁰⁾ を形成することがわかる. NHPP の性質により, $\{M(t), t \geq 0\}$ は NHPP となり, その平均値関数を $H(t)$ とすると,

$$\Pr[M(t)=m] = \frac{\{H(t)\}^m}{m!} \exp[-H(t)]$$

$$(m=0, 1, 2, \dots), \quad (7)$$

$$H(t) = H_1(t) + H_2(t)$$

$$= a_1(1 - \exp[-b_1 t])$$

$$+ a_2\{1 - (1 + b_2 t)\exp[-b_2 t]\}, \quad (8)$$

を得る. また, テスト開始時点における潜在エラーの総期待数を a とすると, $a = a_1 + a_2$ である. そこで, $p = a_2/a$ とし, これらを式(8)に代入して整理すると,

$$H(t) = a\{1 - (1-p)\exp[-b_1 t]$$

$$- p(1 + b_2 t)\exp[-b_2 t]\}, \quad (9)$$

となる. ここで, パラメータ p はクラス2エラーの含有率を表すことになる. 式(9)の平均値関数をもつ NHPP モデルを指数-S字形ソフトウェア信頼度成長モデル (exponential-S-shaped software reliability growth model) と呼ぶことにする. 式(9)から明らかのように, $p=0$ のときに指数形ソフトウェア信頼度成長モデル, $p=1$ のときに遅延S字形ソフトウェア信頼度成長モデルにそれぞれ帰着する.

上記の指数-S字形ソフトウェア信頼度成長モデルから, 次のようなソフトウェア信頼性評価のための定量的尺度を導出することができる^{3), 11)}. 任意のテスト時刻 t におけるソフトウェア内の期待残存エラー数は

$$n(t) = a\{(1-p)\exp[-b_1 t]$$

$$+ p(1 + b_2 t)\exp[-b_2 t]\}, \quad (10)$$

により与えられる. また, テストが時刻 t まで進行しているときに, 時間区間 $(t, t+x]$ ($x \geq 0$) においてエラーの発見されない確率は

$$R(x|t) = \exp[-a\{(1-p)e^{-b_1 t} + p(1 + b_2 t)e^{-b_2 t}$$

$$- (1-p)e^{-b_1(t+x)}$$

$$- p(1 + b_2(t+x))e^{-b_2(t+x)}\}], \quad (11)$$

となり, ソフトウェア信頼度 (software reliability) と呼ばれている. さらに, 信頼性工学の分野でも典型的な信頼性評価尺度である平均故障間隔 (mean time between failures, 以下 MTBF と略す) も導出でき, テスト時刻 t における瞬間 MTBF³⁾は

$$MTBF(t) = \frac{1}{dH(t)/dt}$$

$$= \frac{1}{ab_1(1-p)e^{-b_1 t} + ab_2 p t e^{-b_2 t}}, \quad (12)$$

により与えられる.

3. パラメータの推定

実際のテスト工程において, 式(7)および式(9)により記述される指数-S字形ソフトウェア信頼度成長モデルに基づいてソフトウェアの信頼性評価を行う場合, テスト工程で得られたエラーデータを用いて, モデルパラメータ a , p , b_1 , および b_2 を推定しなければならない. 本論文では, 最尤法^{2), 3), 11)}に基づくパラメータ推定法を採用し, エラーデータの収集法に応じた2つの方法を考える.

いま, y_k ($k=1, 2, \dots, n$) を一定のテスト時刻 t_k ($k=1, 2, \dots, n$; $0 < t_1 < t_2 < \dots < t_n$) までに発見された累積エラー数の実測値とする. このような観測データに対する尤度関数 l は

$$l = \Pr[M(t_1)=y_1, M(t_2)=y_2, \dots, M(t_n)=y_n],$$

$$(13)$$

により与えられる. NHPP の性質から, 式(13)は

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

$$\cdot \exp[-\{H(t_k) - H(t_{k-1})\}], \quad (14)$$

となる³⁾. ここで, $t_0=0$ および $y_0=0$ とする. また, 平均値関数 $H(t)$ に含まれるクラス2エラーの含有率 p の値はエラーの複雑さを解析することにより得られているものとする³⁾. 式(14)の両辺の自然対数を取り, $L = \ln l$ とすると,

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

$$- H(t_k) - \sum_{k=1}^n \ln[(y_k - y_{k-1})!], \quad (15)$$

となる.

パラメータ a , b_1 および b_2 の最尤推定値 \hat{a} , \hat{b}_1 および \hat{b}_2 を求めるために式(15)を a , b_1 および b_2 について偏微分して,

$$\frac{\partial L}{\partial a} = \frac{\partial L}{\partial b_1} = \frac{\partial L}{\partial b_2} = 0, \quad (16)$$

とおけば同時尤度方程式を得る. 式(16)を数値的に解けば, 最尤推定値 \hat{a} , \hat{b}_1 および \hat{b}_2 が求められる. これに対して, クラス1およびクラス2エラーが区別されたエラーデータが利用できない場合 (例えば, テスト工程においてクラス1およびクラス2エラーを区別することが不可能であったり, 過去の実績データに本モデルを適用して同種の新プロジェクトの初期の信頼性評価に反映させる場合) は, クラス2エラーの含有率 p の値を開発管理者の経験などにより既知として与え, 式(16)を数値的に解いて同様にモデルパラメー

タの最尤推定値 \hat{a} , \hat{b}_1 , および \hat{b}_2 を求めることができる。

4. 数値例

本章では、式(7)および式(9)により表される指数-S字形ソフトウェア信頼度成長モデルを用いた信頼性評価例を示す。ここで用いたエラー発見データは、文献8)に引用された累積発見エラー数に関する14組のデータ (t_k, y_k) ($k=1, 2, \dots, 14$) である。このデータのエラー発見期間は42か月であり ($t_{14}=42$ (months)), その期間中に4,140個のエラーが発見されている(テスト対象のプログラムの特徴およびデータの詳細は文献8)を参照)。まず、文献8)より、エラー発見の観測終了時点でのクラス2エラーの含有率 p を $p=0.28$ とした(本論文では、文献8)で述べられているエラー複雑度の分類基準のうち、静止型をクラス1エラーとし、条件型および複合型をクラス2エラーとして取り扱った)。前章で述べた最尤法によりモデルパラメータ \hat{a} , \hat{b}_1 , および \hat{b}_2 を推定すると、それぞれ、 $\hat{a}=4583.1$, $\hat{b}_1=0.097440$ および $\hat{b}_2=0.057816$ となった。これらの推定結果から、当該ソフトウェアにはエラー発見の観測終了時点において約443個のエラーが残存しているものと推測される。図4に発見された累積エラー数の実測値の経時変化と、式(9)によって表される平均値関数 $H(t)$ の推定値を示した。

さらに、上記の推定結果に基づき、既存のモデルとのデータに対する適合性に関する比較を行う。ここで取り上げたモデルは、前述した指数形ソフトウェア信頼度成長モデルである。このモデルを上記の実測データに適用し、式(3)の平均値関数を最尤法により推定すると、

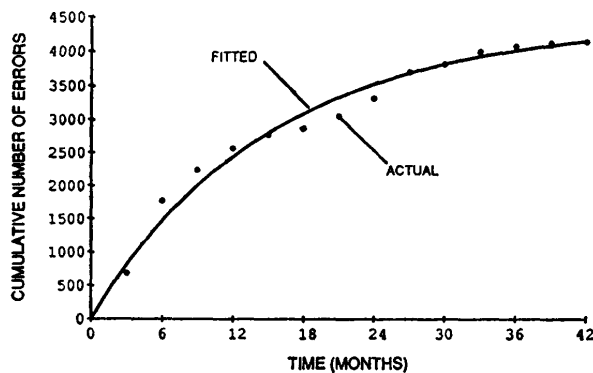


図4 実測データに対して推定された平均値関数 $\hat{H}(t)$
 Fig. 4 Estimated mean value function $\hat{H}(t)$.

$$\hat{H}_1(t) = 4403.3(1 - \exp[-0.067064 \times t]). \quad (16)$$

を得る。モデルの観測データに対する適合性を比較するための評価基準として、テストにより発見される累積エラー数の実測値 y_j と、その推定値 \hat{y}_j との偏差2乗和

$$SSE = \sum_{j=1}^{14} (y_j - \hat{y}_j)^2, \quad (17)$$

およびモデルパラメータの最尤推定値が得られたときの対数尤度関数値 L (log-likelihood function) を用いる。また、 \hat{y}_j は観測時刻 t_j のときのNHPPモデルの平均値関数の最尤推定値、すなわち式(3)の推定値 $\hat{H}_1(t_j)$ または式(9)の推定値 $\hat{H}(t_j)$ である。後者の評価基準である対数尤度関数値 L は、式(13)および式(15)からも明らかのように、NHPPを仮定したときに上記の観測データが得られる確率の対数値であり、適合性評価のためによく用いられるものである(例えば、Keiller and Miller¹²⁾参照)。表1にそれらの比較結果を示した。表1から、データに対する偏差2乗和および対数尤度関数値の結果から、この観測データに関しては指数-S字形ソフトウェア信頼度成長モデルのほうが指数形ソフトウェア信頼度成長モデルより適合性が優れていることがわかる。

データに対する適合性の良好であった本モデルを用いて、ソフトウェアの信頼性評価尺度を推定する。式(10)の期待残存エラー数 $n(t)$ のエラー発見の観測時間の経過に伴う推移を図5に示す。図5を用いて、例えばエラー発見の観測をさらに継続したものと仮定し、観測時刻 $t=50$ における期待残存エラー数は302.5個であると推定できる。また、式(11)のソフトウェア信頼度 $R(x|t)$ を $x=0.01, 0.05$, および $x=0.1$ の場合について、エラー発見の観測時間の経過に伴う変化を図6に示す。これによって、たとえば当該ソフトウェアの将来における観測時間 $x=0.1$ での信頼度の目標値を式(11)において $R(0.1|t)=0.80$ と設定した場合、これを満たすエラー発見の観測時刻を求めると $t=89.4$ となる。

表1 実測データに対するモデルの適合性比較
 Table 1 The comparison results in terms of goodness-of-fit.

	Exponential-S-shaped SRGM	Exponential SRGM
SSE	308023	366471
L	-482.232	-490.724

(SRGM: Software Reliability Growth Model)

さらに、図7に式(12)の瞬間 MTBF を示す。これによりエラー発見の観測時間の経過につれて信頼性が向上していく様相が読み取れる。

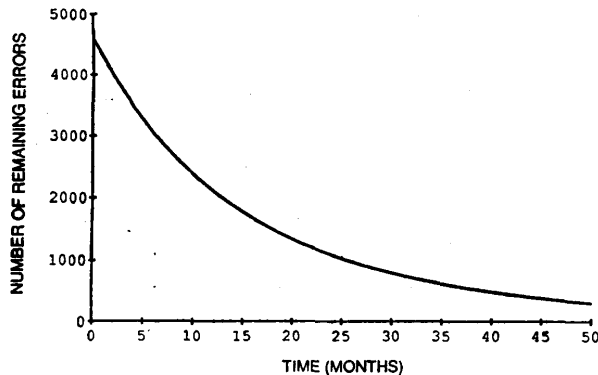


図5 実測データに対して推定された期待残存エラー数 $\hat{n}(t)$
Fig. 5 Estimated number of errors remaining at time t , $\hat{n}(t)$.

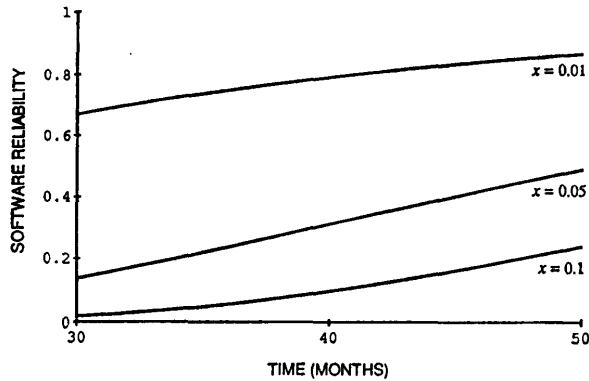


図6 推定されたソフトウェア信頼度 $\hat{R}(x|t)$
($x=0.01, 0.05, 0.1$)
Fig. 6 Estimated software reliability functions $\hat{R}(x|t)$'s
($x=0.01, 0.05, 0.1$).

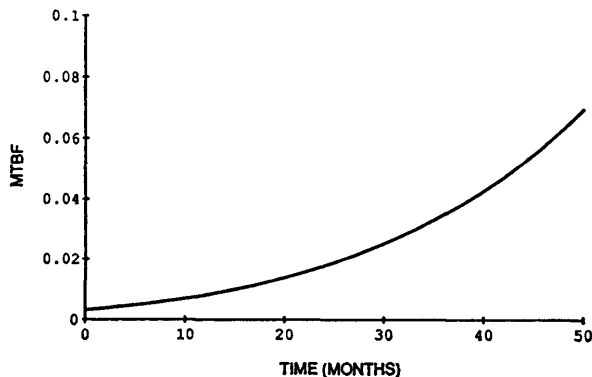


図7 推定された平均ソフトウェア故障間隔 $\widehat{MTBF}(t)$
Fig. 7 Estimated MTBF, $\widehat{MTBF}(t)$.

5. むすび

本論文では、テスト工程におけるエラー発見過程を、エラーの発見難易度をクラス1エラーとクラス2エラーとに分類して捉え、それぞれを相異なるNHPPにより記述し、これらを重ね合わせることでモデル化を行った。このモデルは、クラス2エラーの含有率を、中川、竹中⁸⁾による各クラスのエラーの相対検出率を用いたり、またはソフトウェア開発管理者の経験から与えることにより、従来よりも精密な信頼性評価を可能にするものである。

また、実際のエラー発見データに対して本モデルを適用し、偏差2乗和および対数尤度関数値による適合性比較の結果より、このモデルが既存の指数形ソフトウェア信頼度成長モデルに比べて良好なことがわかった。

今後はさらに研究を進め、実際のテスト工程においてエラーの発見難易度により区別してデータを収集するための合理的な方法や、エラーのクラスを区別した収集データが利用できない場合に必要となるパラメータの決定方法、さらにより実的な信頼性評価尺度についても考察を深めていかねばならない。

謝辞 本論文の著者のうち山田は、本研究の一部に対して文部省科学研究費（一般研究（C）, 課題番号04650316）の補助を受けたことを付記する。

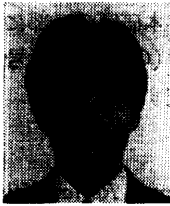
参 考 文 献

- 1) Bittanti, S. (ed.): *Software Reliability Modeling and Identification*, Springer-Verlag, Berlin (1987).
- 2) Musa, J.D., Iannino, A. and Okumoto, K.: *Software Reliability: Measurement, Prediction, Application*, McGraw-Hill, New York (1987).
- 3) 山田 茂: ソフトウェア信頼性評価技術, HBJ 出版局, 東京 (1989).
- 4) Yamada, S., Osaki, S. and Narihisa, H.: A Software Reliability Growth Model with Two Types of Errors, *R. A. I. O.-Operations Research*, Vol. 19, No. 1, pp. 87-104 (1985).
- 5) Goel, A.L. and Okumoto, K.: Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures, *IEEE Trans. Reliability*, Vol. R-28, No. 3, pp. 206-211 (1979).
- 6) 岩佐 博: エラーの重要度およびエラー修正過程を考慮した複合ソフトウェア信頼度成長モデルの提案, 情報処理学会論文誌, Vol. 29, No. 5, pp. 505-512 (1988).

- 7) 山田 茂, 大場 充: 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, 情報処理学会論文誌, Vol. 27, No. 8, pp. 821-828 (1986).
- 8) 中川 豊, 竹中市朗: エラー複雑度に基づくソフトウェア信頼性モデル, 電子情報通信学会論文誌, Vol. J74-D-1, No. 6, pp. 379-386 (1991).
- 9) 木村茂和: ホスト適用業務プログラムにおける信頼性評価, ソフトウェアシンポジウム '91 論文集, pp. B-26-B-34 (1991).
- 10) Cox, D. R. and Lewis, P. A. W.: *The Statistical Analysis of Series of Events*, Methuen, London (1966).
- 11) 山田 茂, 大寺浩志: ソフトウェアの信頼性—理論と実践的応用—, ソフト・リサーチ・センター東京 (1990).
- 12) Keiller, P. A. and Miller, D. R.: On the Use and the Performance of Software Reliability Growth Models, *Reliability Engineering and System Safety*, Vol. 32, Nos. 1 & 2, pp. 95-117 (1991).

(平成 4 年 1 月 13 日受付)

(平成 4 年 9 月 10 日採録)



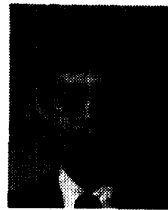
木村 光宏

平成元年広島大学工学部第二類卒業。平成 3 年同大大学院博士課程前期修了。現在同大大学院博士課程後期在学中。ソフトウェアの信頼性に関する研究に従事。日本 OR 学会, 電子情報通信学会各会員。



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 58~63 年岡山理科大学勤務。昭和 63 年広島大学工学部第二類 (電気系) 助教授, 現在に至る。工学博士。ソフトウェア信頼性モデル, ソフトウェアマネジメントモデル, 信頼性工学, 品質管理などの研究に従事。平成 3 年度情報処理学会 Best Author 賞受賞。著書「ソフトウェア信頼性評価技術」(HBJ 出版局), 「ソフトウェアの信頼性~理論と実践的応用~」(ソフト・リサーチ・センター) など。電子情報通信学会, 日本 OR 学会, 日本経営工学会, IEEE 各会員。



尾崎 俊治 (正会員)

昭和 45 年京都大学大学院博士課程修了。同年広島大学工学部講師, 同年助教授, 昭和 61 年教授。工学博士。信頼性工学およびシステム工学の研究に従事。昭和 45~47 年南カルフォルニア大学研究員。昭和 51~52 年マンチェスター大学研究員。日本 OR 学会第 4 回大西文献賞受賞。著書「応用確率論」, 「Stochastic System Reliability Modeling」など。編「Stochastic Models in Reliability Theory」, 「Reliability Theory and Applications」, 電子情報通信学会, 日本 OR 学会各会員。