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†

岩佐 博††

ソフトウェア開発において，ソフトウェアの信頼性を定量的に評価することは重要である．この方法の一つとして，ソフトウェア信頼度成長モデルによる信頼性予測が知られている．本論文では，ソフトウェアの試験段階において発見されるソフトウェアのエラー数が試験初期には指数形成長曲線を示した後にS字形成長曲線を示す現象を考察し，その信頼度成長モデルについて議論する．このとき，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」を提案する．実際のソフトウェア・エラーデータを基に，従来のモデルと本モデルとの信頼性予測値を比較する．その結果，本モデルが良い信頼性予測値を推定することを示す．このモデルの基本となる確率過程は非同次ポアソン過程である．モデルのパラメータは非線型最小二乗法の一つである Marquardt 法により推定する．

1. まえがき

近年，ソフトウェアの品質や生産性の向上に関する研究が活発に行われている．このうち，ソフトウェアの信頼性評価に関する研究は，現在の高度情報化社会を支える高信頼度ソフトウェア製品の開発およびその工程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って得られた情報からソフトウェアの信頼性を定量的に評価することは意義のあることである．

定量的にソフトウェアの信頼性評価を行う方法として，ソフトウェア信頼度成長モデル (Software reliability growth model) (Ramamoorthy and Bastani[1] 参照) による信頼性予測が知られている．この方法は次のような考えに基づいている．ソフトウェアの試験段階では，エラーの発見および修正が行われるが，このとき新しいエラーは作り込まないと仮定すると，試験の経過と共に内在するエラー数は減少し，ソフトウェアの信頼性は向上する．このようなエラー発見過程を記述したモデルにより信頼性を予測することができる．今まで，多くのソフトウェア信頼度成長モデル[1)~9)] が提案されてきた．これらはエラーの発見された時刻またはエラーの認定された時刻とそのエラーが修正された時刻とが一致するものと仮定して成り立っていた．しかし，実際の試験では，重要度の高いエラーの修正が試験に及ぼす影響は高く，またエラー発見から修正までの期間に試験が継続して行われる場合が多い．このことは，モデルを作成する上で興味ある問題である．

本論文では，実際に観測されたソフトウェアの総発見エラー数が試験時間の経過と共に，初期においては指数形成長曲線を示し，中期以後はS字形成長曲線を示す現象を考察する．このとき，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」(Dual Software Reliability Growth Model) を提案する．実際のソフトウェア・エラーデータを基に，従来のモデルと本モデルとの信頼性予測値を比較する．その結果，本モデルが良い信頼性予測値を推定することを示す．このモデルの基本となる確率過程は非同時ポアソン過程 (nonhomogeneous Poisson process, 以下 NHPP と略す) (Ascher and Feingold[2] 参照) である．また，モデルのパラメータは非線型最小二乗法の一つである Marquardt 法 (Marquardt[10] 参照) により推定する．

2. NHPP

ソフトウェアの試験において，試験開始後，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を次のように定義する．

$$\{N(t),\ t\geqq 0\}, \tag{1}$$

この確率過程に対して，平均値関数 $a(t)$ および強度関数 $\lambda(t)$ をもつ NHPP を次のように仮定する．

$$Pr\{N(t)=n\}=\frac{\{a(t)\}^n}{n!}\exp[-a(t)]$$
$$(n=0,1,2,\cdots), \tag{2}$$

$$a(t)=\int_0^t \lambda(x)\,dx,\quad t\geqq 0, \tag{3}$$

† Proposal of Dual Software Reliability Growth Model Considered with Software Error Importance and Fixing Process by HIROSHI IWASA (Product Assurance Laboratory, IBM Japan, Ltd.).

†† 日本アイ・ビー・エム(株)製品保証

これにより，ソフトウェア・エラー発見過程を定式化することができる．ここで，$a(t)$ は試験時刻 t までに発見される総期待エラー数を表す「NHPP の平均値関数」(mean value function) と呼ばれる．また，$\lambda(t)$は，試験時刻 t における瞬間エラー発見率を表す「NHPP の強度関数」(intensity function) と呼ばれる．

3. 従来のモデル

試験時間に対する総発見エラー数の成長曲線として図1(b)のような指数形成長曲線および図1(c)のようなS字形成長曲線が知られている．

Goel and Okumoto[3], Littlewood[4] および Musa[5] らのソフトウェア信頼度成長モデルは前者の指数形成長曲線を示す次のようなエラー発見事象を記述している．

1．ソフトウェア・エラーの発見は互いに独立な事象である．
2．任意の試験時刻において，発見されるエラー数はソフトウェア内に残存するエラー数に比例する．
3．エラー1個当りの発見率は一定である．
4．発見されたエラーは次のテストが行われる前に修正される．

特に，Goel and Okumoto[3] は次に示すような平均値関数を持つ NHPP によって上記の事象を記述している．

$$m(t)=N\{1-\exp(-b\cdot t)\}, \qquad (4)$$

ここで，b は任意の試験時刻におけるエラー1個当りの発見率を表す．また，N 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．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ている．このモデルは，すべてのエラーが他のエラーとは独立に発見可能である場合に適合する．

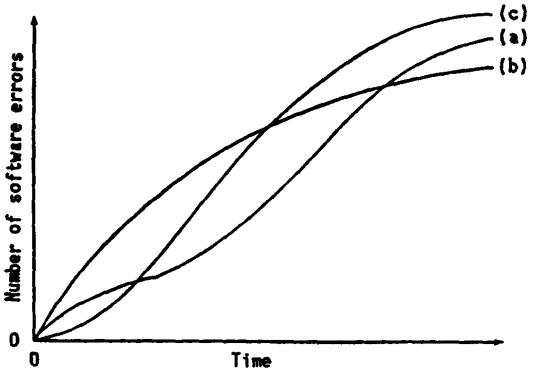

図1 ソフトウェア信頼度成長曲線
Fig. 1 Software reliability growth curves.

大場・梶山[8),9)] は試験により発見されるソフトウェア・エラーを，他のエラーとは独立な発見の容易なもの（以後，独立なエラーと呼ぶ）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（以後，従属なエラーと呼ぶ）とに分類できることに注目した．そこで，大場・梶山[8),9)] はS字形成長曲線を示す次のようなエラー発見事象を記述したソフトウェア信頼度成長モデルを提案した．

1．ソフトウェア内には互いに従属なエラーが混在する．
2．任意の試験時刻において，発見されるエラー数はソフトウェア内に残存する発見可能なエラー数に比例する．
3．エラー1個当りの発見率は一定である．
4．発見されたエラーは完全に修正される．

このエラー発見事象を平均値関数

$$h(t)=N\frac{1-\exp(-b\cdot t)}{1+c\cdot\exp(-b\cdot t)}, \qquad (5)$$

とする NHPP により記述した．ここで，c は

$$c(r)=(1-r)/r, \quad r>0, \qquad (6)$$

と表される．r は加速係数と呼ばれ，試験開始時におけるソフトウェア内に潜在する発見可能なエラーの比率を表している．このモデルは，加速S字形ソフトウェア信頼度成長モデルと呼ばれている．このモデルの特長は，従属なエラーが混在する場合においても適合することにある．

4. 複合ソフトウェア信頼度成長モデル

実際の試験におけるエラー成長曲線では，図1(a)の曲線が表すように，試験初期においては指数形成長曲線を示した後にS字形成長曲線を示す（以後，指数-S字形成長曲線と呼ぶ）場合が良く観測される．この現象は，修正されるエラーの重要度の違いと，エラー発見から修正までの期間に試験が継続して行われることに起因するものと考えられる．

ここで，エラーの修正がエラー発見現象に及ぼす影響度によりエラーを分類する．独立なエラーと従属なエラー（大場・梶山[8),9)] 参照）については先に述べたが，見方を変えて従属するエラーが存在することにより従属されるエラーの修正が終了しなければ従属するエラーが発見可能とならない重要度の高いエラーと，従属するエラーがなく他のエラーの発見には影響を及ぼさないエラーとに分類する．つまり，エラー修正に

より発見可能なエラーの割合を増加させるエラー（以後，高重要度エラーと呼ぶ）と発見可能なエラーの割合には影響を及ぼさないエラー（以後，単独エラーと呼ぶ）とに分類する．

指数-S字形成長曲線を示す「複合ソフトウェア信頼度成長モデル」は次のようなエラー発見事象を基本とする．

1．ソフトウェア内には互いに従属なエラーが混在する．
2．エラーの発見から，そのエラーの修正までの期間に試験は継続して行われる．
3．従属なエラーが発見可能なエラーに変り始めるのは，高重要度エラーが最初に修正される時刻からである．
4．任意の試験時刻において，発見されるエラー数はソフトウェア内に残存する発見可能なエラー数に比例する．
5．エラー1個当りの発見率は一定である．
6．発見されたエラーは完全に修正される．

これらのエラー発見事象を次に示す平均値関数を持った NHPP により記述する．

$$I(t)=N\frac{1-\exp(-b\cdot t)}{1+c\cdot\exp(-b\cdot w)}, \tag{7}$$

w は最初の高重要度エラー修正時刻 d を起点とする試験時刻を示し，

$$w(t,d)=\begin{cases}t-d, & t-d\geqq 0,\\ 0, & t-d<0,\end{cases} \tag{8}$$

で表される．この d を加速開始時刻と呼ぶ．また，c は

$$c(r)=(1-r)/r,\quad r>0, \tag{9}$$

で表される．r は試験開始時および加速開始時刻におけるソフトウェア内に潜在する発見可能なエラーの割合を表す．これを複合ソフトウェア信頼度成長モデルの加速係数と呼ぶ．

ここで，「複合ソフトウェア信頼度成長モデル」(Dual Software Reliability Growth Model) を提案する．

本モデルをエラー発見事象の確率密度関数（probability density function）により考察する．$i-1$ 番目のエラー発見から i 番目のエラー発見区間におけるエラー発見事象の確率密度関数は

$$H(\tau_i)=b\cdot U(i)\cdot\{N-(i-1)\}, \tag{10}$$

と表すことができる．$U(i)$ は $N-i$ 個の残存エラーのうち発見可能なエラーの割合を表す．ここで，高重要度エラーの修正が i_f 番目のエラー発見と同時に行われると仮定すると，本モデルの $U(i)$ は，

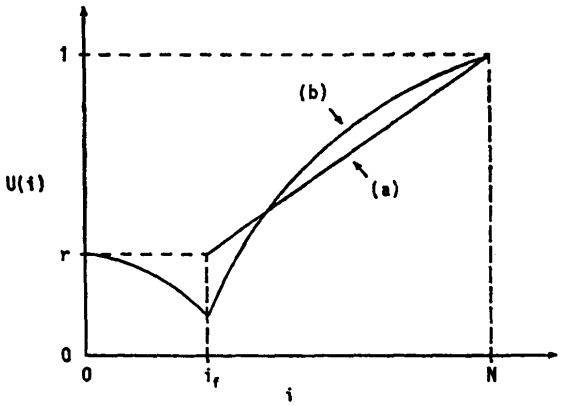

図 2 発見されたエラー数 i と発見可能エラー率関数 $U(i)$ との関係
Fig. 2 Number of detected failures i and detectable failure rate function $U(i)$.

・$0\leqq i\leqq i_f$ のとき

$$U(i)=(r\cdot N-i)/(N-i), \tag{11}$$

・$i_f<i\leqq N$ のとき

$$U(i)=\frac{(1-r)\cdot N}{N-i_f}\cdot\frac{i}{N}+\frac{r\cdot N-i_f}{N-i_f}, \tag{12}$$

となる（図2参照）．つまり，試験が開始されてから高重要度エラーが最初に修正されるまでの期間においては，試験開始前に潜在する独立なエラーのみが発見可能であるため，試験の経過と共に発見可能なエラーの割合は式 (11) で示すように減少する．また，それ以後の試験においては高重要度エラーの修正が行われるため，それまで発見不可能であった従属なエラーが徐々に発見可能となり，発見可能なエラーの割合は上昇する．実際の試験における上昇曲線は図2(b)に示すような曲線を描くと考えられることから，最初の高重要度エラーの修正時点では発見可能なエラーの割合が r であり，それ以後は一定の割合で上昇するような直線により近似する．式(12)がその近似式である．

ところで，本モデルは試験開始時刻と加速開始時刻が一致 ($i_f=0$) したとき

$$U(i)=(1-r)\cdot i/N+r, \tag{13}$$

となり，$H(\tau_i)$ は習熟S字形ソフトウェア信頼度成長モデルの確率密度関数を表し，また加速係数 r が1のとき

$$U(i)=1, \tag{14}$$

となり，$H(\tau_i)$ は指数形ソフトウェア信頼度成長モデルの確率密度関数を表す特長を持っていることがわかる．

ここで，本モデルで仮定した基本要因の一つであるエラー発見事象2の「エラーの発見から，そのエラーの修正までの期間に試験は継続して行われる.」を「発見されたエラーは次のテストが行われる前に修正される.」と仮定しなおしたとすると，エラーの発見時刻と修正時刻は一致することになる．また，任意の試験時刻における発見可能な高重要度エラーおよび発見可能な単独エラー1個当りの発見率は一定であることから，全試験期間において常に高重要度エラーの発見および修正が行われる確率があり，常に発見可能エラー率 $U(i)$ が増加するようなモデルを考えることができる．これは，本モデルの確率密度関数において高重要度エラーの修正が $i_f=0$ 番目に起こったときの状態にあたると考えられる．つまり，実試験において高重要度エラーの発見時刻と修正時刻との差が非常に小さい場合には，高重要度エラーが最初に修正される時刻 d を0とおけることを意味している．

5. パラメータの推定

試験時刻 $(t_1, t_2, \cdots, t_n)$ においてエラーが発見された場合，それらの時刻において発見された総エラー数を $(y_1, y_2, \cdots, y_n)$ とする．このとき，式(7)の平均値関数の時刻 $(t_1, t_2, \cdots, t_n)$ における総期待エラー数を $(g_1, g_2, \cdots, g_n)$ とすると，$\sum_{i=1}^{n}[y_i-g_i]^2$ を最小にするときのパラメータ値が求める値であるとする．そこで，非線型最小二乗法の一つである Marquardt 法 (Marquardt[10] 参照) により式(7)，(8)および(9)のパラメータ N, b および r を求める．また，パラメータ d は実測された値を使用する．

6. モデルの適合性比較

実際に観測された指数-S字形成長曲線を示す2例のソフトウェアデータおよびS字形成長曲線を示す1例のソフトウェアデータをそれぞれ指数形ソフトウェア信頼度成長モデル，加速S字形ソフトウェア信頼度成長モデルおよび複合ソフトウェア信頼度成長モデルに適用し，モデルの適合性を比較する．これらの3例は約 200 Kloc. からなるホスト通信端末プログラムの試験で観測されたものである．**図3**，**図6**および**図8**に観測された総発見エラーの成長曲線を示す．便宜上，最終的に発見された総エラー数および試験終了時刻をそれぞれ100とおいた．推定精度の比較基準としては，最終的に発見される実際の総エラー数 N_a と推定された総期待エラー数 N_e との誤差

$$A=(N_a-N_e)/N_a, \tag{15}$$

および，各観測値 y_k と推定値 $\hat{y}_k$ との誤差分散

$$V=\frac{1}{n}\sum_{k=1}^{n}(y_k-\hat{y}_k)^2, \tag{16}$$

を用いる．

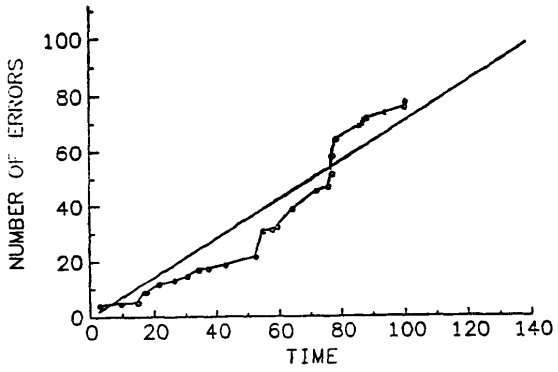

図 3 指数形ソフトウェア信頼度成長モデルによる事例1の解析結果
Fig. 3 Case 1 deta analysis by the exponential model.

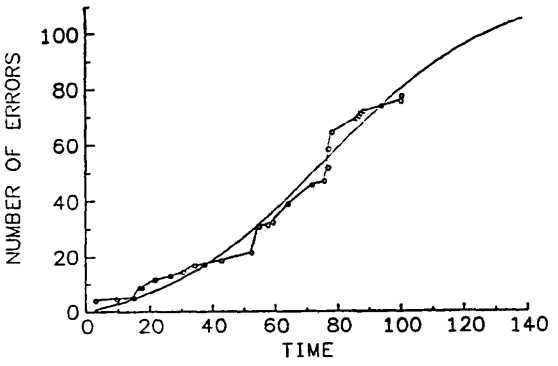

図 4 加速S字形ソフトウェア信頼度成長モデルによる事例1の解析結果
Fig. 4 Case 1 data analysis by the inflection S-shaped model.

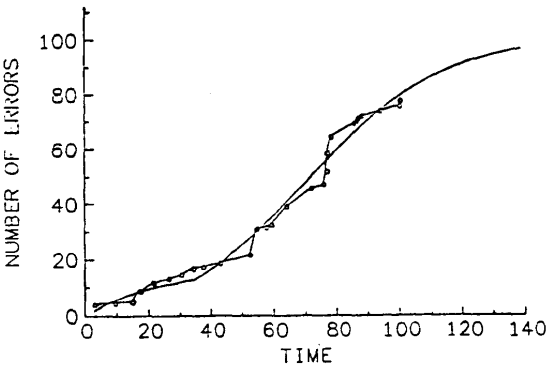

図 5 複合ソフトウェア信頼度成長モデルによる事例1の解析結果
Fig. 5 Case 1 data analysis by the dual model.

指数-S字形成長曲線を示す事例1のソフトウェアエラーデータを各モデルで解析する．まず，平均値関数 $m(t)$ をもつ指数形ソフトウェア信頼度成長モデルを適用すると $m(t)$ における総期待エラー数 N は ∞ となる．したがって，事例1に指数形ソフトウェア信頼度成長モデルは適合しないことがわかる．次に，平均値関数 $h(t)$ をもつ加速S字形ソフトウェア信頼度成長モデルを適用すると

$$h(t)=117.19\frac{1-\exp(-0.036141\cdot t)}{1+16.073\cdot\exp(-0.036141\cdot t)}, \tag{17}$$

が得られる．事例1の加速開始時刻が 34 と観測されたことから $d=34$ とおき，平均値関数 $I(t)$ をもつ複合ソフトウェア信頼度成長モデルを適用すると

$$I(t)=100.31\frac{1-\exp(-0.046028\cdot t)}{1+5.4037\cdot\exp(-0.046028\cdot w)},$$

$$w(t)=\begin{cases}t-34, & t-34\geqq 0,\\ 0, & t-34<0,\end{cases} \tag{18}$$

が得られる．ここで，比較基準 A を $h(t)$ と $I(t)$ に適用すると

$$A_h=17.19(\%), \tag{19}$$

$$A_I=0.31(\%), \tag{20}$$

となる．この比較基準によるると複合ソフトウェア信頼度成長モデルの方が良く適合することがわかる．次に，比較基準 V を適用すると

$$V_h=13.767, \tag{21}$$

$$V_I=10.423, \tag{22}$$

となる．この比較基準によっても複合ソフトウェア信頼度成長モデルの方が良く適合することがわかる．これら2つの比較基準から指数-S字形成長曲線を示す事例1には複合ソフトウェア信頼度成長モデルが良く適合するといえる．

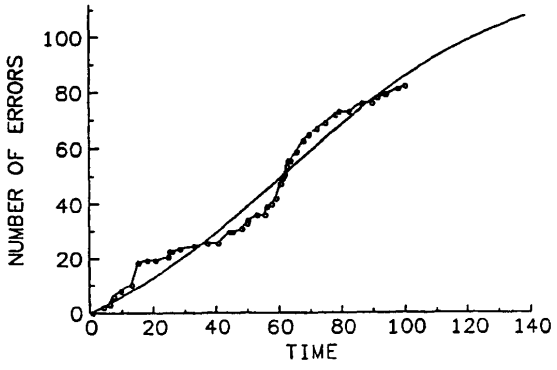

図 6 加速S字形ソフトウェア信頼度成長モデルによる事例2の解析結果
Fig. 6 Case 2 data analysis by the inflection S-shaped model.

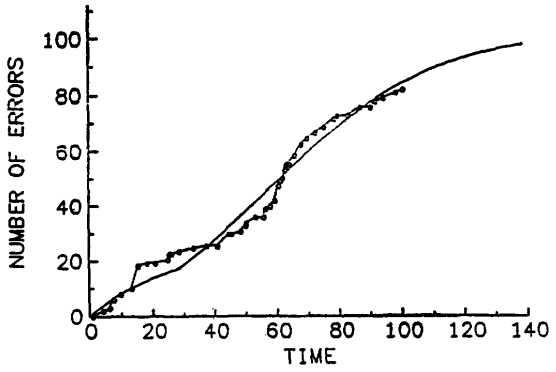

図 7 複合ソフトウェア信頼度成長モデルによる事例2の解析結果
Fig. 7 Case 2 data analysis by the dual model.

次に，事例1と同じく指数-S字形成長曲線を示す事例2のソフトウェアエラーデータを各モデルで解析する．指数形ソフトウェア信頼度成長モデルを適用すると $m(t)$ における総期待エラー数 N はやはり ∞ となる．したがって，事例2の場合にも指数形ソフトウェア信頼度成長モデルは適合しないことがわかる．次に，平均値関数 $h(t)$ をもつ加速S字形ソフトウェア信頼度成長モデルおよび平均値関数 $I(t)$ をもつ複合ソフトウェア信頼度成長モデルを適用する．ただし，事例2の加速開始時刻が 28.24 と観測されたことから $I(t)$ のパラメータ d を 28.24 とおいた．推定された平均値関数は

$$h(t)=123.15\frac{1-\exp(-0.026972\cdot t)}{1+5.2143\cdot\exp(-0.026972\cdot t)}, \tag{23}$$

$$I(t)=102.34\frac{1-\exp(-0.038066\cdot t)}{1+2.9607\cdot\exp(-0.038066\cdot w)},$$

$$w(t)=\begin{cases}t-28.24, & t-28.24\geqq 0,\\ 0, & t-28.24<0,\end{cases} \tag{24}$$

となる．ここで，比較基準 A および V を $h(t)$ と $I(t)$ に適用すると

$$A_h=23.15(\%),\quad V_h=20.424, \tag{25}$$

$$A_I=2.34(\%),\quad V_I=16.529, \tag{26}$$

が得られる．これら2つの比較基準から指数-S字形成長曲線を示す事例2についても複合ソフトウェア信頼度成長モデルが良く適合するといえる．

次に，S字形成長曲線を示す事例3のソフトウェアエラーデータを平均値関数 $m(t)$，$h(t)$ および $I(t)$ をもつ各モデルで解析する．ただし，事例3の加速開始時刻が 20.024 と観測されたことから $I(t)$ のパラメータ d を 20.024 とおいた．推定された平均値関数は

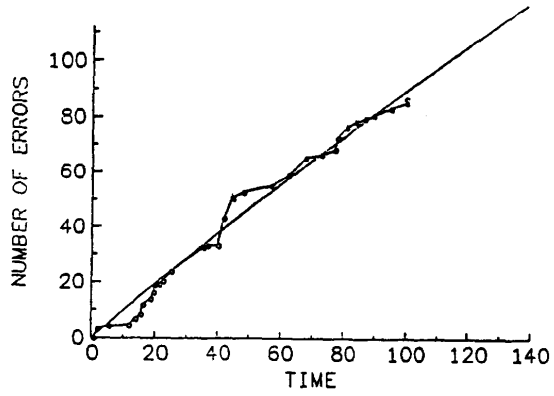

図 8 指数形ソフトウェア信頼度成長モデルによる事例3の解析結果
Fig. 8 Case 3 data analysis by the exponential model.

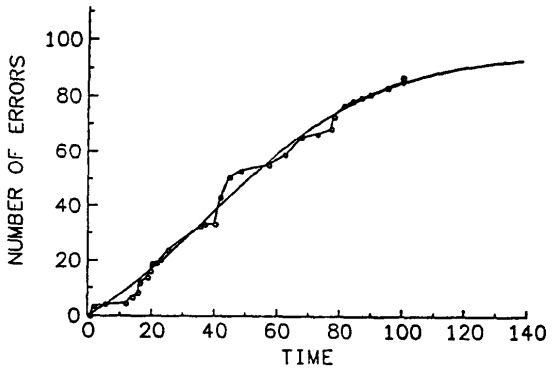

図 9 加速S字形ソフトウェア信頼度成長モデルによる事例3の解析結果
Fig. 9 Case 3 data analysis by the inflection S-shaped model.

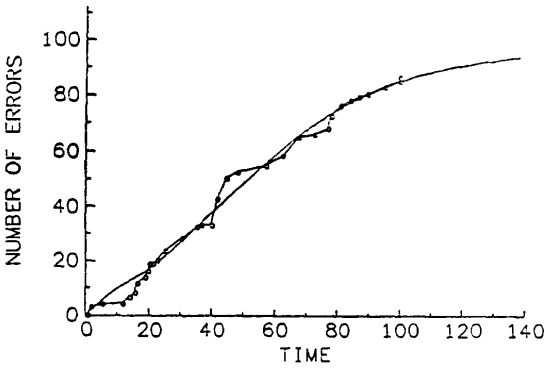

図 10 複合ソフトウェア信頼度成長モデルによる事例3の解析結果
Fig. 10 Case 3 data analysis by the dual model.

$$m(t)=541.13\{1-\exp(-0.0018008\cdot t)\}, \quad (27)$$

$$h(t)=95.287\frac{1-\exp(-0.037447\cdot t)}{1+4.2449\cdot\exp(-0.037447\cdot t)}, \quad (28)$$

$$I(t)=96.444\frac{1-\exp(-0.036826\cdot t)}{1+2.0462\cdot\exp(-0.036826\cdot w)},$$

$$w(t)=\begin{cases} t-20.024, & t-20.024\geqq 0, \\ 0, & t-20.024<0, \end{cases} \quad (29)$$

となる．ここで，比較基準 A および V を $m(t)$, $h(t)$ および $I(t)$ に適用すると

$$A_m=441.13(\%), \quad V_m=12.84, \quad (30)$$

$$A_h=-4.713(\%), \quad V_h=7.6548, \quad (31)$$

$$A_I=-3.556(\%), \quad V_I=9.6706, \quad (32)$$

が得られる．これら2つの比較基準からS字形成長曲線を示す事例3については加速S字形ソフトウェア信頼度成長モデルおよび複合ソフトウェア信頼度成長モデルが良く適合するといえる．

以上のことから，発見された総エラー数が指数-S字形成長曲線を示す時には，複合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．

7. むすび

本論では，一般的にエラー発見時刻と修正時刻との間には時間のずれがあり，エラー修正がエラー発見事象に与える影響度により単独エラーと高重要度エラーとにエラーを分類できることに注目して，高重要度エラーが最初に修正される時刻を考慮した「複合ソフトウェア信頼度成長モデル」を提案した．本モデルはモデル比較において，今までのモデルでは良い推定値を得られなかった指数-S字形成長曲線を示す実試験データにおいても良い適合性を示すことがわかった．

今後，多くの実試験データによる本モデルの妥当性評価と，エラー属性およびエラー修正がエラー発見事象に与える影響を考察していかなければならない．

謝辞 貴重な時間をさいて御指導をいただいた日本アイ・ビー・エム(株) 大場 充氏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) 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).
4) 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).
5) Musa, J. D.: The Measurement and Management of Software Reliability, *Proc. IEEE*, Vol. 68, No. 9, pp. 1131-1143 (1980).
6) 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).
7) 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).
8) 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, 28-9 (1983).
9) Ohba, M.: Software Reliability Analysis Models, *IBM J. Res. Dev.*, Vol. 28, No. 4, pp. 428-443 (1984).
10) Marquardt, D. W.: An Algorithm for Least Squares Estimation of Nonlinear Parameters, *J. Soc. Indust. Appl. Math.*, Vol. 11, pp. 431-441 (1963).

(昭和62年12月7日受付)
(昭和63年3月9日採録)

岩佐 博 (正会員)

昭和28年生. 昭和53年東京理科大学理工学部数学科卒業. 昭和53年同大学情報科学科研究生. 昭和54～59年, 三菱スペース・ソフトウェア(株)勤務. 昭和59年より日本アイ・ビー・エム(株)に勤務, 現在同社製品保証部門所属. 信頼性工学, ソフトウェア品質評価法に興味を持つ. 電子情報通信学会会員.