高信頼性ソフトウェア開発のための一般化離散型信頼度成長モデル

井上　真二・山田　茂
鳥取大学大学院工学研究科社会経営工学講座

Generalized Discrete Software Reliability Growth Modeling for Developing a Highly Reliable Software

Shinji INOUE and Shigeru YAMADA
Department of Social Management Engineering,
Graduate School of Engineering, Tottori University
Tottori, 680-8552 Japan
E-mail: {ino,yamada}@sses.tottori-u.ac.jp

Abstract: Unification of software reliability growth models gives us useful information for understanding properties of a software reliability growth model theoretically. And, a modeling framework for the unification enables us to develop plausible SRGM's reflecting the software failure-occurrence phenomenon in a testing-phase of a software development process. In this paper, focusing on software reliability growth process depending on discrete-time domain, such as the number of executed test cases, we propose a generalized discrete SRGM assuming that the software failure-occurrence times distribution follows a discrete Weibull distribution by developing a modeling framework for unification of discrete-time software reliability models. And we discuss parameter estimation of our generalized discrete SRGM, which is based on a heuristic algorithm. Finally, we show numerical examples of our generalized discrete SRGM by using actual fault counting-data.

Key Words: Software reliability growth model, Discrete-time model, Program-size, Binomial process, Discrete Weibull distribution, Software reliability analysis.

1．はじめに

ソフトウェアの信頼性（software reliability）は，ソフトウェアの重要な品質特性の１つであり，それをテスト工程および運用段階において，定量的かつ可能な限り正確に計測･評価する必要がある．ソフトウェアの信頼性を定量的に評価する基盤技術の１つとして，ソフトウェア信頼性モデルがある．その中の動的モデルであるソフトウェア信頼度成長モデル（software reliability growth model, 以下 SRGM と略す）[1-4]と呼ばれる数理モデルは，テスト工程および運用段階でのソフトウェア実行過程に伴うソフトウェア故障発生時間，もしくは，ソフトウェア故障発生頻度を確率的現象として捉え，確率モデルによって記述するモデルである．これまでに，このような確率的現象に関する様々な特徴を反映しながら，数多くの SRGM が提案されている．

しかしながら，テスト工程において観測される様々なソフトウェア故障発生パターンに対しても柔軟に対応できる SRGM は数少ない．このような問題を解決するための１つの手法として，順序統計量[5]，無限サーバ待ち行列理論[6]，累積ベルヌーイ試行過程[7]などを用い，ソフトウェア故障発生現象，もしくは，フォールト発見事象を理論的かつ統一的に取り扱いながら SRGM を構築するための一般化枠組みに関する議論が行われている．また，適用性の観点から多く実用に供されている非同次ポアソン過程（nonhomogeneous Poisson process, 以下 NHPP と略す）モデルが，算術平均，幾何平均，および調和平均のような概念を用いて記述できることに着目しながら，NHPP モデルの一般化枠組みも提案されている[8]．このような一般化枠組みは，SRGM の新しい構築枠組みとしても考えられ，近年では，既存の一般化枠組みに基づいて，どのようなソフトウェア故障発生パターンにも比較的柔軟に対応できる SRGM の開発も行われている[9]．ただし，これらの議論は，連続時間モデルに関するものが大半を占める．テスト工程において実行されたテストケース数や，月お

よび週もしくは日などのカレンダー時間の単位で推移するソフトウェア信頼度成長過程を記述する離散時間モデル[10]に対しては，それらの議論が十分であるとは言えない．

本研究では，Langberg and Singpurwalla による SRGM の一般化手法[5]に基づき，上述した離散型 SRGM の一般化モデル構築枠組みを構築する．特に今回は,信頼性評価精度のさらなる改善のため,ソフトウェア信頼性に影響を与える要因であるプログラム規模も同時に考慮した一般化枠組みを与える．また，この枠組みに基づいて，ソフトウェア故障発生時間が離散型ワイブル分布に従うものと仮定することで構築される離散ワイブル型 SRGM を構築する．本研究で構築する離散型 SRGM は，離散型ワイブル分布を用いてソフトウェア故障発生パターンを記述するため，これまでに議論されている典型的なソフトウェア信頼度成長パターンを比較的柔軟に記述できるものと考えられる．本研究では，さらに，最ゆう法および発見的手法に基づいた提案モデルのパラメータ推定手法についても議論する．最後に，実際のソフトウェア開発プロジェクトにおいて観測されたフォールト発見数データを用いて，提案モデルの適用例を示す．

２．離散ワイブル型ソフトウェア信頼度成長モデル

離散型SRGMを理論的かつ統一的に取り扱うための基本的モデリング枠組み[5,11]に基づいて，プログラム規模を考慮した離散型SRGMの新たな構築枠組みについて議論する．また，この枠組みにおいて，ソフトウェア故障発生時間分布が離散型ワイブル分布に従うような新しい離散型SRGMの構築も行う．

２．１　モデリング枠組み

プログラム規模を考慮した離散型SRGMの構築枠組みを構築するにあたり，テスト工程におけるソフトウェア故障発生現象に対して，まず，以下のような基本的仮定を与える．

(A1) ソフトウェア故障が発生した場合，その原因となるフォールトは，直ちにかつ完全に修正・除去される．

(A2) 各ソフトウェア故障は，それぞれ，独立かつ時間に関してランダムに発生し，各ソフトウェア故障発生時刻は，それぞれ，同一の離散型確率分布 $P(i)$ $(i=0,1,2,\cdots)$ に従う．

(A3) テスト開始前にソフトウェア内に潜在する総フォールト数（初期潜在フォールト数）N_0 は，ある確率分布に従う確率変数とする．

テスト開始後 i 期目までに発見される総フォールト数を表す離散型確率過程 $\{N(i), i=0,1,2,\cdots\}$ を導入する．このとき，ソフトウェア故障発生現象もしくはフォールト発見事象に関する上述の基本的仮定から，テスト開始後 i 期目までに m 個のフォールトが発見される確率関数は，

$$\Pr\{N(i)=m\} = \sum_n \binom{n}{m}\{P(i)\}^m\{1-P(i)\}^{n-m}\Pr\{N_0=n\} \quad (m=0,1,2,\cdots,n), \tag{1}$$

のように求められる．式(1)において，テスト工程におけるソフトウェア故障発生現象もしくはフォールト発見事象に関する確率的挙動は，初期潜在フォールトを表す確率変数 N_0 にある適切な確率分布を与えることで，特徴付けられることがわかる．例えば，N_0 に対してパラメータ ω のポアソン分布を仮定した場合，テスト工程におけるソフトウェア故障発生現象は，平均値関数 $\omega P(i)$ をもつNHPPに従うことが知られている[11].

２．２　プログラム規模を考慮したモデリング枠組み

本研究では，式(1)における初期潜在フォールト数を表す確率変数 N_0 に対して，以下のような，パラメータ (K,λ) をもつ二項分布を仮定することで，プログラム規模を考慮した離散型 SRGM を構築するための枠組みを与える．

$$\Pr\{N_0=n\} = \binom{K}{n}\lambda^n(1-\lambda)^{K-n} \quad (0<\lambda<1;\ n=0,1,2,\cdots,K), \tag{2}$$

式(2)には，初期潜在フォールト数に関する以下のような物理的意味が含まれている．

(B1) テスト開始時点におけるプログラムは，K コード行(LOC)で構成される．

(B2) 各コードは，それぞれ一定の確率 λ で 1 個のフォールトを含む．

(B3) プログラム内のコード中に潜在するフォールトにより引き起こされるソフトウェア故障は，それぞれ独立かつランダムに発生する．

初期潜在フォールト数を表す確率変数N_0に，式(2)のような二項分布を仮定することによって，プログラム規模がソフトウェア信頼度成長過程に与える影響を，SRGMに反映することができる[12]．一般的に，プログラム規模は，ソフトウェアの信頼度成長過程に影響を与える要因として考えられるソフトウェアの複雑性メトリクス[13]の1つとして知られている．また，ソフトウェア複雑性メトリクスの中でも最も単純で一般によく使われるいるメトリクスの1つでもある．従って，式(2)のような手法を用いてこのような複雑性メトリクスをSRGMに反映することは，ソフトウェアの信頼性評価を行う上で有効な手法であると考えることができる．

式(2)を式(1)に代入して整理すると，テスト開始後i期目までにm個のフォールトが発見される確率関数は，次式のように求められる．

$$\Pr\{N_B(i)=m\}=\binom{K}{m}\{\lambda P(i)\}^m\{1-\lambda P(i)\}^{K-m} \quad (m=0,1,2,\cdots,K). \tag{3}$$

式(1)における初期潜在フォールト数N_0が式(2)のような二項分布に従う場合，テスト工程におけるソフトウェア故障発生現象もしくはフォールト発見事象は，離散型二項過程に従うことがわかる．また，ソフトウェア故障発生時刻に関するある適切な離散型確率分布$P(i)$を，式(3)に適用することで，プログラム規模を考慮した離散型SRGMを容易に構築できる．

２．３　ソフトウェア故障発生時間分布

式(3)に基づいて具体的な離散型SRGMを構築するためには，式(3)に含まれるソフトウェア故障発生時間分布$P(i)$に対して，ある適切な離散型確率分布関数を与える必要がある．この場合，ソフトウェア故障発生時間分布としては，その特徴を比較的柔軟に捉えることができる離散型確率分布を適用することが，ソフトウェア故障発生現象もしくはフォールト発見事象を一般的に記述する意味で重要となる．本研究では，離散型確率分布の中でも，上述した性質を有する離散型ワイブル分布（discrete Weibull distribution）[14]を，ソフトウェア故障発生時間分布として適用することにする．

離散型ワイブル分布の分布関数は，次のように与えられる．

$$P(i)=1-(1-p)^{i^\beta} \quad (i=0,1,2,\cdots;\ \beta>0,\ \ 0<p<1). \tag{4}$$

ここで，pは単位期間当りに1つのソフトウェア故障が発生する確率を表し，βは形状パラメータを表す．ところで，1つのソフトウェア故障に対するハザードレートは，式(4)から，

$$z(i)=1-(1-p)^{(i+1)^\beta-i^\beta},$$

と導出できる．すなわち，式(4)の離散型ワイブル分布は，形状パラメータβによって，ソフトウェア故障発生時間分布の確率的挙動を柔軟に記述することができる．すなわち，形状パラメータが$\beta<1$，$\beta=1$，または$\beta>1$のような値を取るとき，ソフトウェア故障発生時間分布は，それぞれ，DFR（decreasing software failure rate），CFR（constant software failure rate），IFR（increasing software failure rate）の性質をもつ離散型確率分布となる．

３．ソフトウェア信頼性評価尺度

SRGMを用いてテスト工程におけるソフトウェア故障発生現象の確率的特徴を記述した後，ソフトウェア信頼性評価に有用な尺度に基づきながら，定量的なソフトウェア信頼性評価を行うことは，ソフトウェアの品質管理やプロジェクトマネジメントの観点から重要である．本研究では，議論した一般化枠組みに基づいて，種々のソフトウェア信頼性評価尺度を一般的に導出する．その後に，初期潜在フォールト数を表す確率変数が，式(2)のような二項分布に従う場合に対するソフトウェア信頼性評価尺度を，それぞれ，具体的に導出する．

３．１　発見フォールト数の期待値および分散

本研究では，テスト開始後i期目までに発見される総フォールト数$N(i)$を確率変数として取り扱っているため，その期待値や分散から，ソフトウェア信頼度達成状況などの情報を得ることは極めて重要である．

式(1)より，テスト開始後i期目までに発見される総フォールト数$N(i)$の期待値$\mathrm{E}[N(i)]$は，次のように導出される．

$$\mathrm{E}[N(i)] = \sum_{z=0}^{n} z \sum_{n} \binom{n}{z} \{P(i)\}^{z} \{1-P(i)\}^{n-z} \Pr\{N_0 = n\}$$
$$= \mathrm{E}[N_0]P(i).$$

また，同様に，$N(i)$の分散は，

$$Var[N(i)] = \mathrm{E}[N(i)^2] - (\mathrm{E}[N(i)])^2$$
$$= Var[N_0]\{P(i)\}^2 + \mathrm{E}[N_0]P(i)\{1-P(i)\},$$

のように導出できる．したがって，初期潜在フォールト数を表す確率変数N_0が，式(2)のような二項分布に従う場合，テスト開始後i期目までに発見される総フォールト数$N_B(i)$の期待値および分散は，それぞれ，

$$\mathrm{E}[N_B(i)] = K\lambda P(i),$$
$$Var[N_B(i)] = K\lambda P(i)\{1-P(i)\},$$

のように求められる．このとき，$K\lambda$は，テスト開始前にソフトウェア内に潜在する総期待フォールト数を表すことがわかる．

３．２　ソフトウェア信頼度関数

ソフトウェア信頼度関数は，有用なソフトウェア信頼性評価尺度の１つとして知られている．今回議論しているような離散時間に依存するソフトウェア信頼度成長過程を取り扱う場合，ソフトウェア信頼度関数は，テスト開始後i期目までテストが進行しているとき，時間区間$(i, i+h](i, h = 0, 1, \cdots)$においてソフトウェア故障が発生しない確率として定義される[10]．これより，離散型ソフトウェア信頼度関数$R(i,h)$は，式(1)から，

$$\begin{aligned} &R(i,h) \\ &= \sum_{k} \Pr\{N(i+h) = k \mid N(i) = k\} \Pr\{N(i) = k\} \\ &= \sum_{k} \ [\{P(i)\}^{k} \{1-P(i+h)\}^{-k} \\ &\qquad \sum_{k} \binom{n}{k} \{1-P(i+h)\}^{n} \Pr\{N_0 = n\}], \end{aligned} \tag{5}$$

のように導出される．したがって，初期潜在フォールト数N_0が，式(2)の二項分布に従う場合，離散型ソフトウェア信頼度関数$R_B(i,h)$は，式(9)から，

$$R_B(i,h) = [1 - \lambda\{P(i+h) - P(i)\}]^{K},$$

のように導出できる．

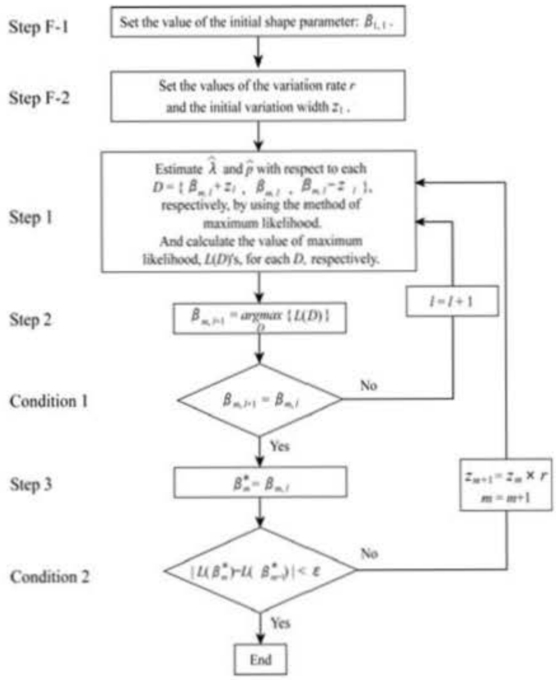

図１：パラメータ推定アルゴリズム．

４．パラメータ推定

SRGMに含まれるパラメータの推定では，最ゆう法やEMアルゴリズム[15]などの手法が多く適用されている．しかしながら，本研究で提案する離散型SRGMに関しては，これらの手法を単純に用いるだけでは，推定したいパラメータλ，p，およびβを同時に推定することが極めて困難である．そのため，本研究では，ゆう度に基づいた発見的アルゴリズムに従って，提案した離散型SRGMのパラメータ推定を行う．

図１に，提案した離散型SRGMのパラメータ推定アルゴリズムを示す．本研究において用いるパラメータ推定アルゴリズムは，これまでに提案されている一般化ガンマソフトウェア信頼性モデルの発見的パラメータ推定アルゴリズム[9]の基本的概念を，今回提案した離散ワイブル型SRGMのパラメータ推定に応用したものである．提案モデルのパラメータ推定アルゴリズムは，ます最初に，提案モデルに含まれる形状パラメータβを，事前に設定した変化幅に応じて固定した上で，パラメータpおよびλを，最ゆう法により推定を行った後，その結果に基づいて，パラメータβの推定を行うものである．

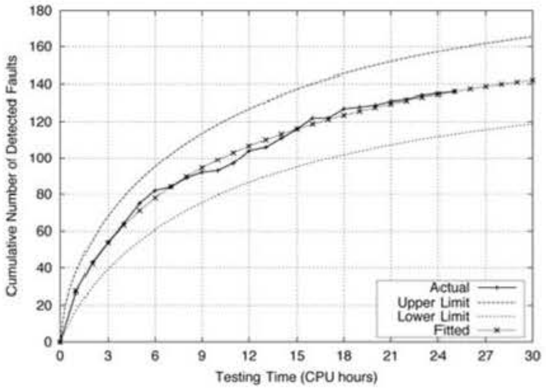

図 2　推定された発見フォールト数の期待値およびその 95%信頼限界．(DS1)

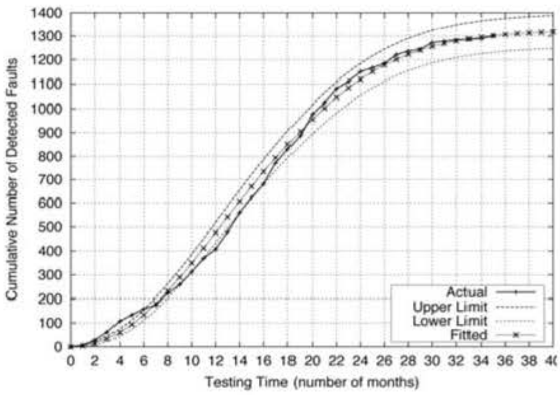

図 3　推定された発見フォールト数の期待値およびその 95%信頼限界．(DS2)

ただし，ある固定された形状パラメータ β に対して，パラメータ p および λ を推定するための対数ゆう度関数は，次のように求められる．はじめに，一定のテスト時間間隔 $(0,t_k]$ において発見された総フォールト数 y_k に関する N 組のフォールト発見数データ $(t_k,y_k)(k=0,1,2,\cdots,N)$ が観測されたものとする．まず，離散型二項過程 $\{N_B(i),i=0,1,2,\cdots,N)$ に関するゆう度関数 l は，ベイズの定理および確率過程 $\{N_B(i),i=0,1,2,\cdots,N)$ が有するマルコフ性を用いて，

$$
\begin{aligned}
l &\equiv \Pr\{N_B(t_1)=y_1,N_B(t_2)=t_2,\cdots,N_B(t_N)=y_N\}\\
&=\prod_{i=2}^{N}\Pr\{N_B(t_i)=y_i\mid N_B(t_{i-1})=y_{i-1}\}\\
&\qquad\qquad \Pr\{N_B(t_1)=y_1\},
\end{aligned}
\tag{6}
$$

のように導出できる．ところで，式(6)に含まれる条件付確率は，次のように書き換えることができる．

$$
\begin{aligned}
&\Pr\{N_B(t_i)=y_i\mid N_B(t_{i-1})=y_{i-1}\}\\
&=\binom{K-y_{i-1}}{y_i-y_{i-1}}\{z(t_{i-1},t_i)\}^{y_i-y_{i-1}}\{1-z(t_{i-1},t_i)\}^{K-y_i}.
\end{aligned}
$$

ここで，

$$
z(t_{i-1},t_i)=\frac{\lambda\{P(t_i)-P(t_{i-1})\}}{1-\lambda P(t_{i-1})}
$$

である．以上より，式(6)は，

$$
l=\prod_{i=1}^{N}\binom{K-y_{i-1}}{y_i-y_{i-1}}\{z(t_{i-1},t_i)\}^{y_i-y_{i-1}}\{1-z(t_{i-1},t_i)\}^{K-y_i},
\tag{7}
$$

のように求められる．ただし，$t_0=0,\ y_0=0,$ および $P(t_0)=0$ である．したがって，対数ゆう度関数 L は，式(7)の両辺に自然対数をとることで，

$$
\begin{aligned}
\log l &\equiv L\\
&=\log K!-\log\{(K-y_i)!\}\\
&\quad-\sum_{i=1}^{N}\log\{(y_i-y_{i-1})!\}+y_N\log\lambda\\
&\quad+\sum_{i=1}^{N}(y_i-y_{i-1})\log\{P(t_i)-P(t_{i-1})\}\\
&\quad+(K-y_N)\log\{1-\lambda P(t_N)\},
\end{aligned}
\tag{8}
$$

のように導出できる．形状パラメータ β は所与のため，式(8)から，パラメータ p および λ に関する同時対数ゆう度方程式を数値的に解くことで，パラメータ p および λ の推定値 $\hat{p}$ および $\hat{\lambda}$ を，それぞれ得ることができる．

5．　モデルの適用例

今回提案した離散ワイブル型 SRGM が，典型的なソフトウェア信頼度成長パターンである指数形および S 字形信頼度成長曲線の両方に対しても柔軟に適応可能であることを実証するため，実測データを用いた提案モデルの適用例を示す．適用する実測データは，それぞれ，指数形信頼度成長曲

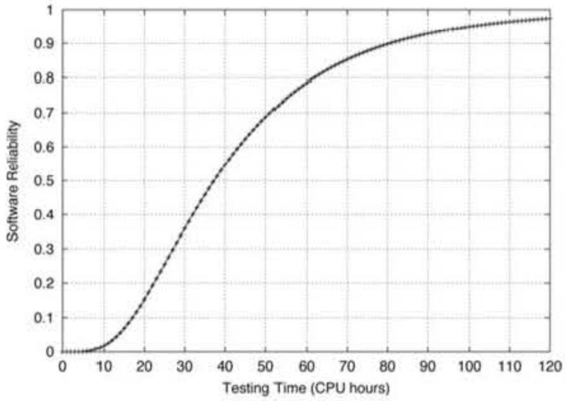

図 4 推定されたソフトウェア信頼度関数．(DS1)

線を示す 25 組のフォールト発見数データ（DS1）$(t_k, y_k)(k=0,1,2,\cdots,25;\ t_{25}=25,\ y_{25}=136)$ [16]，および，S 字形信頼度成長曲線を示す 35 組のフォールト発見数データ（DS2）$(t_k, y_k)(k=0,1,2,\cdots,35;\ t_{35}=35,\ y_{35}=1301)$ [17]である．また，DS1 および DS2 は，それぞれ，プログラム規模が $K=2.170\times10^4$ (LOC) および $K=1.240\times10^5$ (LOC) であるソフトウェアシステムのテスト工程において，得られた実測データである．

上記のフォールト発見数データを用い，本研究において議論したアルゴリズムに基づいて，パラメータ推定を行う．パラメータ推定の結果，DS1 から，パラメータ推定値 $\hat{\lambda}=7.589\times10^{-3}$，$\hat{p}=1.666\times10^{-1}$，$\hat{\beta}=0.7023$ を得ることができた．また，DS2 からは，$\hat{\lambda}=1.068\times10^{-2}$，$\hat{p}=2.639\times10^{-3}$，$\hat{\beta}=2.064$ を得た．ただし，本研究では，$\beta_1=1.0$, $z_1=0.3$, $r=0.25$, および $\varepsilon=1.0\times10^{-3}$ と設定した．

図 2 および図 3 に，それぞれ，DS1 および DS2 を用いて推定された発見されたフォールト数の期待値およびその 95%信頼限界[18]を示す．図 2 および図 3 から，今回提案した離散ワイブル型 SRGM は，指数形または S 字形信頼度成長曲線を描くフォールト発見数データの両方に対しても，モデルに含まれる形状パラメータ β によって，それらの曲線が示す特徴を柔軟に表現できていることがわかる．

また，図 4 および図 5 に，それぞれ，DS1 および DS2 を用いて推定されたソフトウェア信頼度関数 $\hat{R}_B(i,1)$ を示す．図 4 から，DS1 を用いて推定されたテスト開始後 60 期目におけるソフトウェア信頼度 $\hat{R}_B(60,1)$ は，約 0.786 推定される．同様に，図 5 から，DS2 を用いて推定されたテスト開始後 45 期目におけるソフトウェア信頼度 $\hat{R}_B(45,1)$ は，約 0.678 と推定される．

6．おわりに

本研究では，まず，離散型 SRGM に対する既存の一般化枠組みに基づき，プログラム規模を考慮した離散型 SRGM を構築するためのモデリング枠組みについて議論した．さらに，本研究において議論したモデリング枠組みに基づいて，ソフトウェア故障発生時間の確率的挙動が離散型ワイブル分布に従うような，離散ワイブル型 SRGM の構築を行った．また，最ゆう法に基づいた発見的手法による提案モデルのパラメータ推定法についても議論した．今回提案した離散ワイブル型 SRGM の特長としては，モデルに含まれる形状パラメータによって，テスト工程において観測されるフォールト発見数データが示す信頼度成長曲線の形状を，柔軟に捉えながら記述できることが挙げられる．

ただし，今回議論した離散型 SRGM 構築枠組みは，式(1)に含まれる初期潜在フォールト数を表す確率変数 N_0 に対して，式(2)の二項分布を適用することで導出された枠組みである．したがって，$K\lambda=\omega$ を保ったまま，$K\to\infty$，$\lambda\to0$ に近づくような場合，式(2)は平均値 ω のポアソン分布となり，離散型確率過程 $\{N_B(i), i=0,1,2,\cdots,N)$ は $\omega P(i)$ の離散型ポアソン過程に従うことに注意すべきである．このことから，今回議論したような

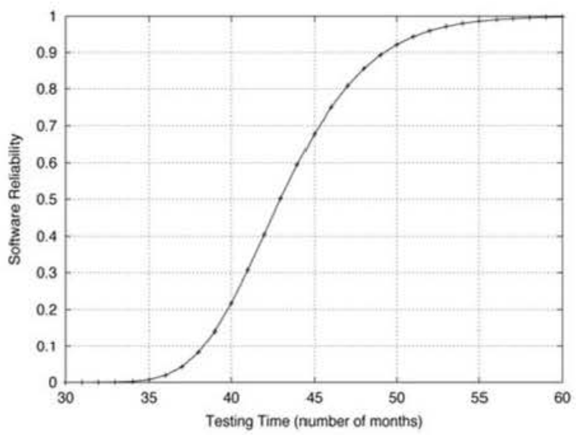

図 5 推定されたソフトウェア信頼度関数．(DS2)

離散型二項過程に従うモデリング枠組みは，小・中規模かつ各コード行当りのフォールト混入率が高いソフトウェアシステムに対して，その効果が得られるものと予想される．今後の課題として，より多くの実測データを用いて，既存の離散型SRGMとの適合性比較を行い，今回提案した離散ワイブル型 SRGM の有効性について議論する必要がある．

参考文献

[1] 山田　茂：ソフトウェア信頼性モデル—基礎と応用—，日科技連出版社，東京，1994 年．

[2] Yamada, S. : Software reliability models, in Stochastic Models in Reliability and Maintenance (S. Osaki, Ed.), Springer-Verlag, Berlin, pp.253-280, 2002.

[3] 山田　茂：ソフトウェア信頼性の基礎－モデリングアプローチ，共立出版, 東京，2011 年．.

[4] Pham, H. : Software Reliability, Springer-Verlag, Singapore, 2000.

[5] Langberg, N. and Singpurwalla : A unification of some software reliability models, SIAM J. Stat. Comput., Vol. 6, No. 3, pp. 781-790, 1985.

[6] Dohi, T., Matsuoka, T., and Osaki, S. : An infinite server queueing model for assessment of the software reliability, Elect. Comm. In Japan, Part 3, Vol. 85, No. 3, pp. 536-544, 2000.

[7] Dohi, T., Yasui, K., and Osaki, S. : Software reliability assessment models based on cumulative Bernoulli trial process, Mathematic. and Comput. Modeli., Vol. 38, No. 11-13, pp. 1177-1184, 2003.

[8] Huang, C.Y., Lyu, M.R., and Kuo, S.Y. : A unified scheme of some nonhomogeneous Poisson process models for software reliability estimation, IEEE Tras. Softw. Eng., Vol. 29, No. 3, pp. 261-269, 2003.

[9] 岡村寛之，安藤光昭，土肥正：一般化ガンマソフトウェア信頼性モデル，電子情報通信学会論文誌，Vol. J87-D-I, No. 8, pp. 805-814, 2004 年 8 月．

[10] Yamada, S. and Osaki, S.　: Discrete software reliability growth models, Appli. Stoc. Mod. Data Ana., Vol. 1, No. 1, pp. 65-77, 1985.

[11] Okamura, H. Murayama, A., and Dohi, T. : EM algorithm for discrete software reliability models: a unified parameter estimation method, Proc. 8th IEEE Intern. Symp. High Assura. Syst. Eng. (HASE'04), 2004, pp. 219-228.

[12] Kimura, M., Yamada, S., Tanaka, H., and Osaki, S. : Software reliability measurement with prior-information on initial fault content, Trans. IPS Japan, Vol. 34, No. 7, pp. 1601-1609, 1993.

[13] 山田　茂，高橋宗雄：ソフトウェアマネジメントモデル入門—ソフトウェア品質の可視化と評価法—，共立出版，東京，1993 年．

[14] Nakagawa, T. and Osaki, S. : The discrete Weibull distribution, IEEE Trans. Reliab., Vol. R-24, No. 5, pp. 300-301, 1975.

[15] 岡村寛之，渡部保博，土肥正，尾崎俊治：EM アルゴリズムに基づいたソフトウェア信頼性モデルの推定，電子情報通信学会論文誌，Vol. J85-A, No. 4, pp. 442-450, 2002.

[16] Goel, A.L. : Software reliability models: Assumptions, limitations, and applicability, IEEE Trans. Softw. Eng., Vol. SE-11, No. 12, pp. 1411-1423, 1985.

[17] 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.

[18] 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.

（受理　平成 24 年 7 月 10 日）