

## テスト環境の変化を考慮した 2 変量ソフトウェア信頼性モデル

鳥取大学・大学院工学研究科 井上 真二 (Shinji Inoue)<sup>†</sup>

鳥取大学・大学院工学研究科 山田 茂 (Shigeru Yamada)<sup>†</sup>

<sup>†</sup>Graduate School of Engineering, Tottori University

### 1 はじめに

ソフトウェア開発工程の最終工程であるテスト工程では、主に、開発されたソフトウェア製品の最終的な品質／信頼性の確認作業が行われる。したがって、ソフトウェアの最終的な品質／信頼性評価をより正確に行うことは、高い信頼性を有したソフトウェアを開発する上で重要な問題である。定量的なソフトウェア信頼性評価を支援する技術としては、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] に基づいた手法が挙げられる。この数理モデルは、ソフトウェア実行過程におけるフォールト発見事象またはソフトウェア故障発生現象を確率変数として捉え、テスト実行と共に観測される信頼度成長現象をソフトウェア信頼度成長過程としてこれを記述するモデルである。SRGM は現実な仮定の下で構築されるべきであるが、これまでに提案されている多くの SRGM は、(1) ソフトウェア信頼度成長過程は本質的にテスト時間だけに依存し、(2) ソフトウェア故障発見事象やフォールト発見事象の確率的性質はテスト期間を通じて変化しない、という 2 つの仮定に基づいて構築されている。しかしながら、現実のテスト工程を考えた場合、観測されるソフトウェア信頼度成長現象は、必ずしもテスト実行時間だけに依存することはなく、ソフトウェア信頼度成長過程に影響を与える種々の信頼度成長要因 (テスト網羅度、テスト習熟性など) にも同時に依存することが十分に考えられる [5]。また、フォールトターゲットの変更、モジュール毎のフォールト密度の違い、単位テスト期間当りの工数の変化などの要因によって、ソフトウェア故障発生現象やフォールト発見事象の確率的性質もテスト期間を通じて同じであるとは必ずしも言えない [7]。本研究では、上述の課題に対して 1 つの回答を与えるソフトウェア信頼度成長モデリング手法について議論する。すなわち、テスト工程において観測されるソフトウェア信頼度成長過程がテスト時間要因とテスト労力要因の 2 要因に依存するような 2 変量ソフトウェア信頼度成長モデリングの範ちゅうにおいて、テスト環境の変化を考慮したモデル構築を行う。また、今回提案する 2 変量モデルに関して、実測データを適用したソフトウェア信頼性解析例を示す。

### 2 2 変量ソフトウェア信頼性モデリング

2 変量ソフトウェア信頼性モデリング枠組み [5, 6] について議論する。まず、検出可能なフォールト数が有限である場合を仮定した 2 変量 SRGM の基本的モデリング枠組みは、一般的に、以下の 3 つの仮定 [8] に基づいて議論される。

- (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 2 変量確率分布  $\Pr\{S \leq s, U \leq u\} \equiv F(s, u) = \int_0^s \int_0^u f(x, y) dx dy$  に従う。ここで、 $S$  および  $U$  はそれぞれテスト時間およびテスト労力投入量を表す確率変数である。また、 $f(x, y)$  は確率密度関数であり、 $\Pr\{A\}$  は事象  $A$  の確率を表す。
- (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。

いま、 $\{N(s, u), s \geq 0, u \geq 0\}$  をテスト時刻  $s$  かつテスト労力投入量  $u$  までに発見された総フォールト数を表す 2 次元確率過程と定義する。このとき、上述の仮定：(A1)~(A3) より、 $(s, u)$  までに  $m$  個のフォー

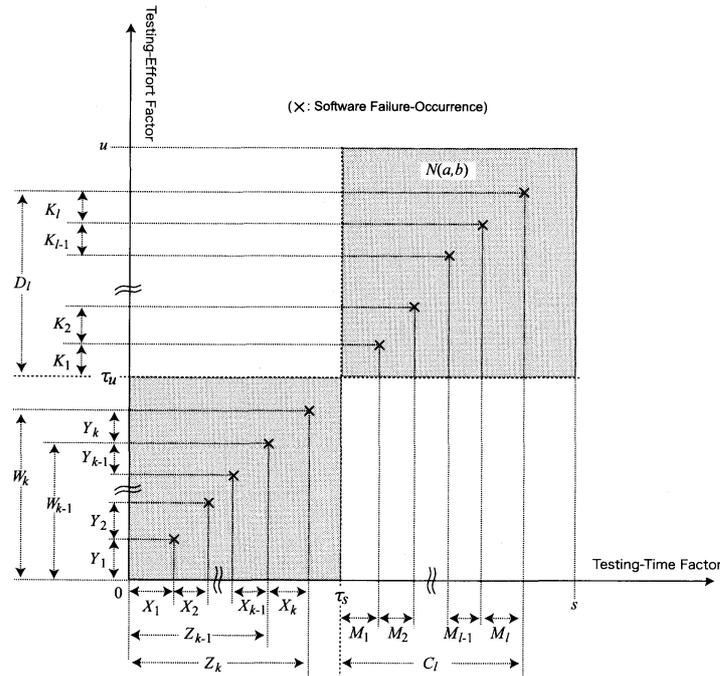


図 1: チェンジポイントを考慮した場合のソフトウェア故障発生現象に関する確率諸量.

ルトが発見される 2次元確率関数は,

$$\Pr\{N(s, u) = m\} = \sum_n \binom{n}{m} \{F(s, u)\}^m \{1 - P(s, u)\}^{n-m} \times \Pr\{N_0 = n\} \quad (m = 0, 1, 2, \dots), \quad (1)$$

と求められる. 式(1)より,  $N_0$  がパラメータ  $\omega (> 0)$  のポアソン分布に従うと仮定した場合, 式(1)は,

$$\begin{aligned} \Pr\{N(s, u) = m\} &= \sum_n \binom{n}{m} \{F(s, u)\}^m \{1 - P(s, u)\}^{n-m} \frac{\omega^n}{n!} \exp[-\omega] \\ &= \frac{\{\omega F(s, u)\}^m}{m!} \exp[-\omega F(s, u)], \end{aligned} \quad (2)$$

となる. 式(2)は, 2次元平均値関数  $E[N(s, u)] = \omega F(s, u)$  をもつ 2次元非同次ポアソン過程と本質的に等価となる. ここで,  $E[\cdot]$  は期待値を表す.

### 3 2変量チェンジポイントモデル

まず, 図1に, 今回のモデリングに必要な確率諸量を定義する. 次に, 定義した確率諸量について, チェンジポイント前後における関係性を以下のように与える.

$$\begin{cases} M_i = \alpha_s(X_i), & C_i = \alpha_s(Z_i), \\ K_i = \alpha_u(Y_i), & D_i = \alpha_u(W_i). \end{cases} \quad (3)$$

ここで,  $\alpha_s(\cdot)$  および  $\alpha_u(\cdot)$  は, それぞれテスト時間要因およびテスト労力要因に対するテスト環境関数 [9] を表し, チェンジポイント前後における各要因の変化を特徴づける関数である. ただし, テスト期間を通じて 2次元空間上にチェンジポイントは 1 回のみしか発生しないものと仮定して, そのチェンジポイントを

$\tau = \{\tau_s, \tau_u\} (0 < \tau_s < s_e, 0 < \tau_u < u_e)$  と標記することにする。ここで、 $s_e$  および  $u_e$  はそれぞれ、テスト終了時刻およびテスト終了時刻まで投入された総テスト労力量を表す。今回のモデリングでは、上述した環境関数が具体的に以下のように与えられる場合を考える：

$$R = \alpha Q. \quad (4)$$

式(4)において、 $R$  および  $Q$  は、それぞれ、チェンジポイント前後のテスト時間要因およびテスト労力要因に関する実測値を要素とする列ベクトル： $R = (m \ k)^T (m > 0, k > 0)$   $Q = (x \ y)^T (x > 0, y > 0)$ 、 $\alpha$  はテスト環境係数から成る定数ベクトル： $\alpha = (\alpha_s \ \alpha_u)$  である。ここで、 $\alpha$  はチェンジポイントにおけるテスト環境要因の変化がチェンジポイント前後のソフトウェア信頼度成長現象の変化に与える影響の相対的な大きさを表すパラメータであり、 $T$  は転置行列を意味する。

いま、チェンジポイントまでに発生した  $n (> 1)$  個のソフトウェア故障に関して、そのときのテスト時間要因データとテスト労力要因データが、それぞれ、 $(x_0, y_0), (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  のように観測されたものとする。ただし、 $0 < x_1 < x_2 < \dots < x_n \leq \tau_s$  および  $0 < y_1 < y_2 < \dots < y_n \leq \tau_u$  である。いま、チェンジポイント前までのテスト領域  $(0, s] \times (0, u] (s \leq \tau_s, u \leq \tau_u)$  において発見される総フォールト数が平均値関数  $\Lambda_B(s, u)$  をもつ 2次元 NHPP に従う場合を考える。このとき、チェンジポイント  $(\tau_s, \tau_u)$  から  $n+1$  番目のフォールトが発見されるまでの時間間隔  $(M_1, K_1)$  の確率分布関数は、以下の条件付き確率によって表すことができる。

$$\Pr\{M_1 > s, K_1 > u\} = \frac{\Pr\{X_{n+1} > \tau_s - x_n + s/\alpha_s, Y_{n+1} > \tau_u - y_n + u/\alpha_u\}}{\Pr\{X_{n+1} > \tau_s - x_n, Y_{n+1} > \tau_u - y_n\}}. \quad (5)$$

上式は最終的に、

$$\Pr\{M_1 > s, K_1 > u\} = \exp[-\Lambda_B(\tau_s + s/\alpha_s, \tau_u + u/\alpha_u) + \Lambda_B(\tau_s, \tau_u + u/\alpha_u) + \Lambda_B(\tau_s + s/\alpha_s, \tau_u) - \Lambda_B(\tau_s, \tau_u)] \quad (6)$$

のように導出できる。したがって、チェンジポイント以降に発見される総期待フォールト数  $\Lambda_A(s, u)$  は、次のように求められる。

$$\begin{aligned} \Lambda_A(s, u) &= -\log[\Pr\{M_1 > s - \tau_s, K_1 > u - \tau_u\}] \\ &= \Lambda_B\left(\tau_s + \frac{s - \tau_s}{\alpha_s}, \tau_u + \frac{u - \tau_u}{\alpha_u}\right) - \Lambda_B\left(\tau_s, \tau_u + \frac{u - \tau_u}{\alpha_u}\right) \\ &\quad - \Lambda_B\left(\tau_s + \frac{s - \tau_s}{\alpha_s}, \tau_u\right) + \Lambda_B(\tau_s, \tau_u). \end{aligned} \quad (7)$$

以上の結果から、チェンジポイントを考慮した 2次元平均値関数は、

$$\Lambda(s, u) = \begin{cases} \Lambda_1(s, u) = \Lambda_B(s, u) \\ \quad (\text{for } 0 \leq s \leq \tau_s, 0 \leq u \leq \tau_u) \\ \Lambda_2(s, u) = \Lambda_B(\tau_s, \tau_u) + \Lambda_A(s, u) \\ \quad = \Lambda_B\left(\tau_s + \frac{s - \tau_s}{\alpha_s}, \tau_u + \frac{u - \tau_u}{\alpha_u}\right) - \Lambda_B\left(\tau_s, \tau_u + \frac{u - \tau_u}{\alpha_u}\right) \\ \quad \quad - \Lambda_B\left(\tau_s + \frac{s - \tau_s}{\alpha_s}, \tau_u\right) + 2\Lambda_B(\tau_s, \tau_u) \\ \quad (\text{for } s > \tau_s, u > \tau_u) \end{cases} \quad (8)$$

のように求められる。式(8)より、今回議論したモデリング枠組みにおいてチェンジポイントを考慮した 2次元 NHPP モデルを構築するためには、チェンジポイント前における 2次元平均値関数、強いてはチェンジポイント前におけるソフトウェア故障発生時間分布を適切に与えることが必要である。

式(8)のように導出されたモデルに基づいて定量的な信頼性評価を行うために有用な信頼性評価尺度についても議論する。ソフトウェア運用信頼度 [5] とは、テストが終了時刻  $s_e$  まで進行しており、テスト開始時刻からその時点までの総テスト労力投入量が  $u_e$  であるとき、以後の(運用)時間区間  $(s_e, s_e + \eta]$

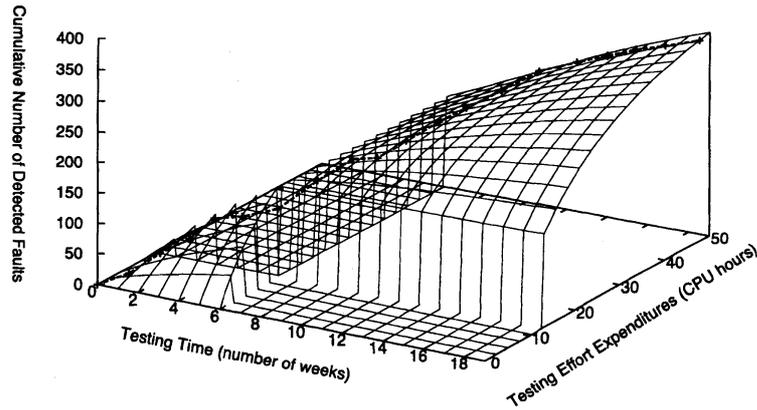


図 2: 推定されたチェンジポイントを考慮した 2 次元平均値関数. ( $\tau_s = 6, \tau_u = 12.89$ )

( $s_e \geq 0, \eta \geq 0$ ) においてソフトウェア故障が発生しない確率を表す. これより, ソフトウェア運用信頼度は, 式 (1) における基本的仮定より,

$$\begin{aligned}
 R(\eta | s_e, u_e) &= \sum_k \Pr\{N(s_e + \eta, u_e) = k | N(s_e, u_e) = k\} \cdot \Pr\{N(s_e, u_e) = k\} \\
 &= \sum_k \left[ \{F(s_e, u_e)\}^k \{1 - F(s_e + \eta, u_e)\}^{-k} \cdot \sum_n \binom{n}{k} \{1 - F(s_e + \eta, u_e)\}^n \cdot \Pr\{N_0 = n\} \right], \quad (9)
 \end{aligned}$$

のように導出される. したがって, 式 (9) において,  $N_0$  がパラメータ  $\omega (> 0)$  のポアソン分布に従うと仮定した場合, ソフトウェア運用信頼度は, 次のように導出できる.

$$R(\eta | s_e, u_e) = \exp[-\{\Lambda(s_e + \eta, u_e) - \Lambda(s_e, u_e)\}]. \quad (10)$$

期待残存フォールト数は, 任意の時刻においてソフトウェア内に残存している総期待フォールト数を表し, 以下のように導出される.

$$\begin{aligned}
 M(s, u) &\equiv E[\bar{N}(s, u)] \\
 &= E[N(\infty, \infty) - N(s, u)] \\
 &= \Lambda(\infty, \infty) - \Lambda(s, u). \quad (11)
 \end{aligned}$$

#### 4 ソフトウェア信頼性解析例

実際のテスト工程において観測されたフォールト発見数データを用いて, 今回提案したモデルに基づいたソフトウェア信頼性解析例を示す. 本研究において用いる実測データは, 19 組のフォールト発見数データ  $(s_k, u_k, y_k) (k = 0, 1, 2, \dots, 19; t_{19} = 19(\text{weeks}), s_{19} = 47.65(\text{CPU hours}), y_{19} = 328)$  [10] である. また, 今回の研究では, 一例として, ソフトウェア故障発生時間分布がパラメータ  $a, b$  および  $z$  の Gumbel 分布 [11]:

$$F(s, u) = (1 - e^{-as})(1 - e^{-bs})(1 + ze^{-as-bu}) \quad (a > 0, b > 0, -1 \leq z \leq 1), \quad (12)$$

に従う場合を考える. また, 本研究では, 簡単のため  $\alpha_s = \alpha_u = \alpha$  として, モデル内部に含まれるパラメータ  $\omega, a, b, z$ , および  $\alpha$  の推定値:  $\hat{\omega}, \hat{a}, \hat{b}, \hat{z}$ , および  $\hat{\alpha}$  を, それぞれ, 最尤法を用いて推定を行う. また,

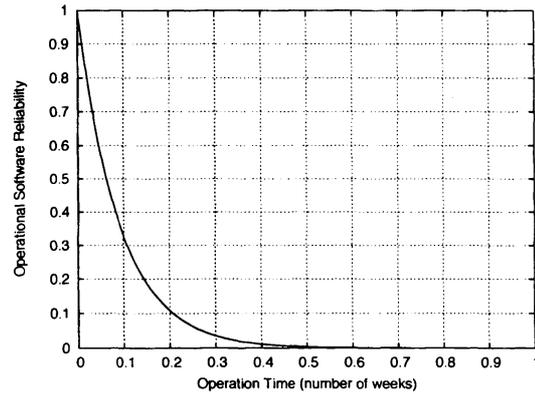


図 3： 推定されたソフトウェア運用信頼度関数 ( $\tau_s = 6, \tau_u = 12.89$ )

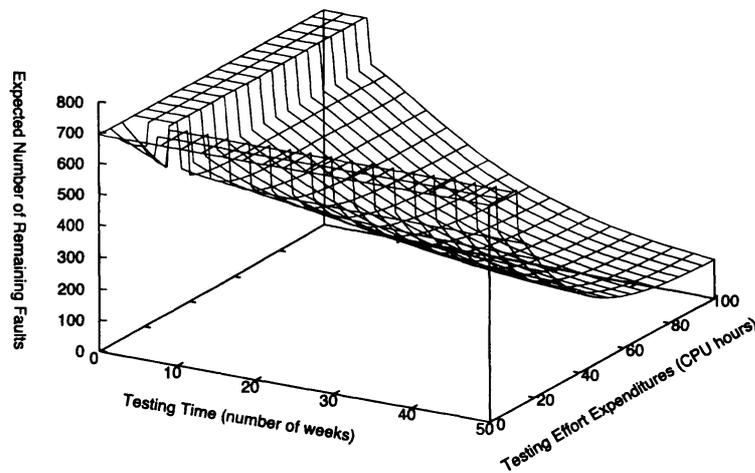


図 4： 推定された期待残存フォールト数  $\widehat{M}(s, u)$

チェンジポイントについては、今回、観測データの挙動から推測して与えることとする。ただし、フォールトターゲット変更時など明確にチェンジポイントを与えることができる場合などは、それに該当しない。

図 2 に、式 (2) の 2 次元 NHPP に対する推定された平均値関数（総期待発見フォールト数）の挙動を示す。図 2 から、2 次元空間上におけるソフトウェア故障発生領域が制限されていることがわかる。すなわち、図 2 において、総期待発見フォールト数がゼロである領域ではソフトウェア故障が発生しない。これは、今回議論したモデリング枠組みに基づいて構築した SRGM の特徴と言える。なお、推定されたテスト環境係数は  $\hat{\alpha}_s = \hat{\alpha}_u = 0.5797$  であり、チェンジポイント前と比較してチェンジポイント後の方が厳しい環境でテストが実行されていることがわかる。

次に、推定された 2 次元平均値関数に基づいて推定される種々のソフトウェア信頼性評価尺度を示す。まず、図 3 に推定されたソフトウェア運用信頼度関数  $\widehat{R}(\eta | 19, 47.65)$  の挙動を示す。図 3 より、テスト終了時と同等の環境で運用した場合、テスト終了後 0.3 週目におけるソフトウェア運用信頼度は、 $\widehat{R}(0.3 | 19, 47.65) \approx 0.036$  と推定される。次に、図 4 に推定された期待残存フォールト数  $\widehat{M}(s, u)$  の挙動を示す。図 4 から、テスト終了時においてソフトウェア内に残存する総期待フォールト数は  $\widehat{M}(19, 47.65) \approx 366$  と推定される。

## 5 おわりに

本研究では、チェンジポイントを考慮した2次元SRGMのモデリング枠組みを与えると共に、議論したモデリング枠組みから構築された2次元SRGMに基づいて、実測データを用いたソフトウェア信頼性解析例を示した。今回議論したモデリング枠組みは、実際のソフトウェア開発現場において観測されるチェンジポイントがソフトウェア信頼度成長過程に与える影響を考慮する点、および、2つのソフトウェア信頼度成長要因に基づいてソフトウェア信頼度成長過程を記述する点から、SRGMに基づいたソフトウェア信頼性評価精度の向上が期待される。今後は、多くの実測データを適用しながら、チェンジポイントが考慮されていない従来モデルとの比較を多角的に行い、今回議論したモデリング枠組みの有効性およびモデルの適合性を検証する必要がある。

### 謝辞

本研究の一部は、文部科学省科学研究費補助金 基盤研究(C) (課題番号 22510150) の援助を受けたことを付記する。

### 参考文献

- [1] J.D. Musa, "A theory of software reliability and its application," *IEEE Trans. Soft. Eng.*, Vol. SE-1, No. 3, pp. 312-327, 1975.
- [2] J.D. Musa, D. Iannio, and K. Okumoto, *Software Reliability: Measurement, Prediction, Application*. McGraw-Hill, New York, 1987.
- [3] 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.
- [4] H. Pham, *Software Reliability*. Springer-Verlag, Singapore, 2000.
- [5] 石井智隆, 土肥正, 「二次元NHPPに基づいたテスト労力依存型ソフトウェア信頼性モデル」, 電子情報通信学会論文誌, vol. J89-D, no. 8, pp. 1684-1694, 2006.
- [6] S. Inoue and S. Yamada, "A framework for two-dimensional software reliability modeling with program size," Proc. 14th ISSAT Intern. Conf. Reliability and Quality in Design, 2008, pp. 198-202.
- [7] M. Zhao, "Change-point problems in software and hardware reliability," *Commun. Statist. — Theory Meth.*, Vol. 22, No. 3, pp. 757-768, 1993.
- [8] N. Langberg and N.D. Singpurwalla, "A unification of some software reliability models," *SIAM J. Scien. Comput.*, Vol. 6, No. 3, pp. 781-790, 1985.
- [9] 岡村寛之, 土肥正, 尾崎俊治, 「運用段階におけるソフトウェア製品の信頼性評価手法 —加速寿命試験モデルの提案—」, 電子情報通信学会論文誌, vol. J83-A, no. 3, pp. 294-301, 2000.
- [10] M. Ohba, "Software reliability analysis models," *IBM J. Res. Dev.*, vol. 28, no. 4, pp. 428-443, 1984.
- [11] E.J. Gumbel, "Bivariate exponential distributions," *J. Amer. Statist. Assoc.*, no. 55, pp. 698-707, 1960.