

フォールト発見事象の複数回の変化を考慮した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)[†]

鳥取大学・大学院工学研究科 山田 茂 (Shigeru Yamada)[†]

[†]Graduate School of Engineering, Tottori University

1 はじめに

実際のテスト工程では、開発管理面からの要因（目標とする信頼度を納期までに達成するための活動など）や固有技術面からの要因（フォールト発見難易度、フォールトの独立性、モジュール毎のフォールト密度の違い、テスト期間中におけるフォールト検出ターゲットの変更など）の変化によって、観測されるソフトウェア信頼度成長過程の時間的挙動も著しく変化する場合がある。このような要因の変化が観測されるテスト時刻はチェンジポイント（change-point, 以下 CP と略す）[9] と呼ばれる。ソフトウェア信頼度成長モデル [5,7,8] に基づいた信頼性評価技術においても、信頼性評価精度を確保するため、ソフトウェア故障発生現象に関する上述した確率的性質の変化を考慮したモデリング枠組みの開発が必要とされており、このような背景の中、現在までに、ソフトウェア信頼度成長過程の変化を考慮した SRGM が現在までに数多く提案されている [1-3,9,10]。

しかしながら、これらの既存研究では、ソフトウェア信頼度成長過程に影響を与える要因の複数回の変化を考慮した SRGM に関する議論はなされていない。実際のテスト工程ではソフトウェア信頼度成長過程に影響を与える要因の変化が複数回観測されることは少なくなく、このような状況を考慮した SRGM の開発は、テスト環境をできるだけ忠実に表現しながら SRGM に基づいたソフトウェア信頼性評価精度を向上させるための興味ある問題の 1 つとして考えられる。本研究では、ソフトウェア故障発生現象に関する確率諸量のチェンジポイント前後における変化をテスト環境関数を用いながら表現して、上述した要因の複数回の変化を考慮した SRGM を構築する。さらに、実際のテスト工程で観測されたデータを適用しながら、今回議論したモデリング枠組みに基づいて構築した新たな SRGM を用いたソフトウェア信頼性解析例も示す。

2 NHPP モデルの構築枠組み

テスト工程において検出可能なフォールト数が有限の場合を仮定した SRGM の多くは、以下の 3 つの基本的仮定 [4] に基づいて構築することができる。

- (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て、発生時刻は、それぞれ、同一の確率分布 $F(t) \equiv \Pr\{T \leq t\} = \int_0^t f(x)dx$ に従う非負の確率変数によって与えられる。ここで、 $f(t)$ は確率密度関数を表す。
- (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） $N_0 (> 0)$ は、ある確率分布に従う確率変数とする。

いま、 $\{N(t), t \geq 0\}$ を任意のテスト時刻 t までに発見された総フォールト数を表す確率過程とする。このとき、(A3) で定義した初期潜在フォールト数 N_0 が平均 $\omega (> 0)$ のポアソン分布に従うと仮定した場合、テスト時刻 t までに m 個のフォールトが発見される確率は、次のように定式化される。

$$\begin{aligned} \Pr\{N(t) = m\} &= \sum_n \binom{n}{m} \{F(t)\}^m \{1 - F(t)\}^{n-m} \frac{\omega^n}{n!} \exp[-\omega] \\ &= \exp[-\omega] \frac{\{\omega F(t)\}^m}{m!} \sum_n \frac{\{\omega(1 - F(t))\}^{n-m}}{(n-m)!} \\ &= \frac{\{\omega F(t)\}^m}{m!} \exp[-\omega F(t)] \quad (m = 0, 1, 2, \dots) \end{aligned} \tag{1}$$

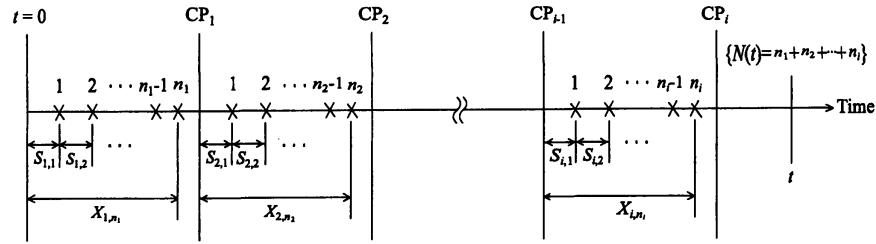


図 1：本研究で用いるソフトウェア故障発生現象に関する確率諸量。

となり，平均値関数 $E[N(t)] = \omega F(t)$ をもつ非同次ポアソン過程 (nonhomogeneous Poisson process, 以下 NHPP と略す) と本質的に等価となる．ここで， $E[\cdot]$ は期待値を表す．したがって，当該モデリング枠組みに基づいて，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が NHPP に従うような SRGM を構築したい場合，ある適切なソフトウェア故障発生時間分布 $F(t)$ を与えることで，式 (1) から具体的に NHPP モデルを構築することができる．

3 複数回のチェンジポイントを考慮した SRGM 構築枠組み

図 1 に，本研究において取り扱うソフトウェア故障発生現象に関する確率諸量を定義する．本研究では，チェンジポイント前後における上述した確率諸量について，ある関係性があるものと仮定して，それをそれぞれ次のように与える．

$$X_{i+1,j} = \alpha_i(X_{i,j}), \quad (2)$$

$$S_{i+1,j} = \alpha_i(S_{i,j}), \quad (3)$$

$$K_{i+1}(t) = K_i(\alpha_i^{-1}(t)). \quad (4)$$

式 (2)–式 (4) において， $\alpha_i(\cdot)$ は i 番目のチェンジポイント発生前後の確率諸量の変化を関係づける関数であり，これをテスト環境関数 (testing-environment function) と呼ぶことにする．特に，本研究では，次のような関数を一例として考える [6]： $\alpha_i(t) = \alpha_i t$ ($t \geq 0, \alpha_i > 0$)．ここで， $\alpha_i (> 0) (i = 1, 2, \dots)$ は i 番目のチェンジポイントに関するテスト環境係数 (testing-environmental coefficient) であり，ソフトウェア信頼度成長過程に影響を与える要因の変化がソフトウェア故障発生現象に関する確率諸量に与える相対的な影響度を表す．なお，この関数は， i 番目のチェンジポイント発生によるソフトウェア故障発生現象の変化を定量的に把握する上で単純かつ有用な関数であると考えられる．

まず，1 番目のチェンジポイント発生に関して NHPP の平均値関数を導出する．いま，テスト開始から 1 番目のチェンジポイント発生までに n_1 個のソフトウェア故障が観測され，それらのソフトウェア故障発生時刻が $0 < x_{1,1} < x_{1,2} < \dots < x_{1,n_1} \leq \tau_1$ であるとする．なお， τ_1 は 1 番目のチェンジポイントを表す．このとき，1 番目のチェンジポイント発生後 1 番目のソフトウェア故障が発生するまでの時間を表す確率変数 $X_{2,1}$ (もしくは $S_{2,1}$) の確率分布： $K_1(t) \equiv \Pr\{S_{2,1} \leq t\}$ の余関数： $\bar{K}_1(t) \equiv 1 - K_1(t)$ は次のように与えることができる．

$$\begin{aligned} \bar{K}_1(t) \equiv \Pr\{S_{2,1} > t\} &= \frac{\Pr\{S_{1,n_1+1} > \tau_1 - x_{1,n_1} + t/\alpha_1\}}{\Pr\{S_{1,n_1+1} > \tau_1 - x_{1,n_1}\}} \\ &= \frac{\exp[-\{M_1(\tau_1 + t/\alpha_1) - M_1(x_{1,n_1})\}]}{\exp[-\{M_1(\tau_1) - M_1(x_{1,n_1})\}]} \end{aligned} \quad (5)$$

ここで， $M_1(t) (\equiv \omega K_0(t))$ は，テスト開始から i 番目のチェンジポイントまでの任意のテスト時間 $t \in (0, \tau_1]$ までに発見された総期待フォールト数もしくは NHPP の平均値関数を表す．式 (5) より，1 番目のチェンジ

ポイント後の任意のテスト時刻 $t \in (\tau_1, \tau_2]$ までに発見される総期待フォールト数は,

$$\begin{aligned} M_2(t) &= -\log \Pr\{S_{2,1} > t - \tau_1\} = -\log \bar{K}_1(t - \tau_1) \\ &= M_1\left(\tau_1 + \frac{t - \tau_1}{\alpha_1}\right) - M_1(\tau_1). \end{aligned} \quad (6)$$

のように導出できる. 同様に, $i (i \geq 2)$ 番目のチェンジポイント後から 1 番目のソフトウェア故障発生時間 $S_{i,1}$ の確率分布関数 $K_i(t)$ は, 最終的に次のように導出することができる.

$$\begin{aligned} &\Pr\{S_{i,1} > t\} \\ &= \exp\left[-\left\{M_1\left(\frac{\sum_{k=1}^i A_{k,i} + t}{A_i}\right) - M_1\left(\frac{\sum_{k=1}^{i-1} A_{k,i} + (x_{i,n} - \tau_{i-1})}{A_{i-1}}\right)\right\}\right] \\ &\quad / \exp\left[-\left\{M_1\left(\frac{\sum_{k=1}^{i-1} A_{k,i-1} + (\tau_i - \tau_{i-1})}{A_{i-1}}\right) - M_1\left(\frac{\sum_{k=1}^{i-1} A_{k,i-1} + (x_{i,n} - \tau_{i-1})}{A_{i-1}}\right)\right\}\right]. \end{aligned} \quad (7)$$

ここで,

$$\begin{cases} A_{k,i} &= \alpha_k \cdot \alpha_{k+1} \cdots \alpha_i (\tau_k - \tau_{k-1}), \\ A_{k,i-1} &= \alpha_k \cdot \alpha_{k+1} \cdots \alpha_{i-1} (\tau_k - \tau_{k-1}), \\ A_i &= \alpha_1 \cdot \alpha_2 \cdots \alpha_i, \\ A_{i-1} &= \alpha_1 \cdot \alpha_2 \cdots \alpha_{i-1}. \end{cases} \quad (8)$$

である. これより, $i (i \geq 2)$ 番目のチェンジポイント後の任意のテスト時刻 t までに発見される総期待フォールト数 $M_i(t) (i = 2, 3, \dots)$ は,

$$M_i(t) = M_1\left(\frac{\sum_{k=1}^i A_{k,i-1} + (t - \tau_{i-1})}{A_{i-1}}\right) - M_1\left(\frac{\sum_{k=1}^{i-2} A_{k,i-2} + (\tau_{i-1} - \tau_{i-2})}{A_{i-2}}\right), \quad (9)$$

のように導出できる. ここで,

$$\begin{cases} A_{k,i-2} &= \alpha_k \cdot \alpha_{k+1} \cdots \alpha_{i-2} (\tau_k - \tau_{k-1}), \\ A_{i-2} &= \alpha_1 \cdot \alpha_2 \cdots \alpha_{i-2}. \end{cases} \quad (10)$$

である.

以上の議論から, テスト時刻 t までにチェンジポイントが $i (i = 1, 2, \dots)$ 回発生した場合を考慮した総期待発見フォールト数 $\Lambda(t)$ は最終的に,

$$\Lambda(t) = \begin{cases} \Lambda_1(t) = M_1(t) & (0 \leq t \leq \tau_1), \\ \Lambda_2(t) = M_1(\tau_1) + M_2(t) & (\tau_1 < t \leq \tau_2), \\ \vdots \\ \Lambda_i(t) = M_1(\tau_1) + M_2(\tau_2) + \cdots + M_{i-1}(\tau_{i-1}) + M_i(t) \\ \quad = M_1\left(\frac{\sum_{k=1}^i A_{k,i-1} + (t - \tau_{i-1})}{A_{i-1}}\right) & (t > \tau_{i-1}). \end{cases} \quad (11)$$

式 (11) から, 複数回のチェンジポイントを考慮した NHPP モデルの平均値関数は, 1 番目の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 $K_0(t)$ もしくは NHPP の平均値関数 $M_1(t)$ を適切に与えることによって構築できることがわかる.

4 ソフトウェア信頼性解析例

本研究では、実際に観測されたデータ： $(t_i, y_i) (i = 1, 2, \dots, 26; t_{26} = 26 \text{ (週)}, y_{26} = 34; \tau_1 = 17, \tau_2 = 21)$ を用いながら、今回のモデリング枠組みに基づいた SRGM の数値例を示す。ここで、 y_i はテスト時刻 t_i までに観測された総フォールト数を表す。また、このデータは、ある Windows バージョンソフトウェアのテスト工程において得られたフォールト発見数データであり、観測された 2 回のチェンジポイントはテスト人員の変更とテスト技術者の変更を行った時刻である。

式 (11) のように与えられた今回のモデリング枠組みに基づいて、具体的な SRGM を構築する。今回のモデリング枠組みを用いて SRGM を構築するためには、1 番目のチェンジポイント発生前におけるソフトウェア故障発生時間分布を適切に仮定する必要がある。例えば、1 番目のチェンジポイント発生前にソフトウェア故障発生時間分布がパラメータ b の指数分布に従うと仮定したとき、1 番目のチェンジポイント発生前における NHPP の平均値関数は、指数形 SRGM： $\Lambda_1(t) = \omega \{1 - \exp[-bt]\}$ ($a > 0, b > 0$) [5] に従うと仮定され、最終的に 2 回のチェンジポイントの影響を考慮した場合の平均値関数は、

$$\Lambda(t) = \begin{cases} \Lambda_1(t) = \omega \{1 - \exp[-bt]\} & (0 < t \leq \tau_1), \\ \Lambda_2(t) = \omega \left\{1 - \exp\left[-b\left(\tau_1 + \frac{t-\tau_1}{\alpha_1}\right)\right]\right\} & (\tau_1 < t \leq \tau_2), \\ \Lambda_3(t) = \omega \left\{1 - \exp\left[-b\left(\tau_1 + \frac{\alpha_2(\tau_2-\tau_1)+(t-\tau_2)}{\alpha_1\alpha_2}\right)\right]\right\} & (\tau_2 < t), \end{cases} \quad (12)$$

のように求めることができる。他にも、1 番目のチェンジポイント発生前にソフトウェア故障発生時間分布がパラメータ $(2, b)$ のガンマ分布に従うと仮定したとき、1 番目のチェンジポイント発生前における NHPP の平均値関数は遅延 S 字形 SRGM： $\Lambda_1(t) = \omega \{1 - (1 + bt) \exp[-bt]\}$ ($a > 0, b > 0$) [8] に従うもの仮定され、最終的に 2 回のチェンジポイントの影響を考慮した場合の平均値関数は、同様にして、

$$\Lambda(t) = \begin{cases} \Lambda_1(t) = \omega \{1 + (1 + bt) \exp[-bt]\} & (0 < t \leq \tau_1), \\ \Lambda_2(t) = \omega \left\{1 - (1 + b(\tau_1 + \frac{t-\tau_1}{\alpha_1})) \exp[-b(\tau_1 + \frac{t-\tau_1}{\alpha_1})]\right\} & (\tau_1 < t \leq \tau_2), \\ \Lambda_3(t) = \omega \left\{1 - (1 + b(\tau_1 + \frac{\alpha_2(\tau_2-\tau_1)+(t-\tau_2)}{\alpha_1\alpha_2})) \exp[-b(\tau_1 + \frac{\alpha_2(\tau_2-\tau_1)+(t-\tau_2)}{\alpha_1\alpha_2})]\right\} & (\tau_2 < t). \end{cases} \quad (13)$$

と導出される。

本研究では、前述したデータを用いて、式 (13) に含まれるパラメータを最尤法に基づいて推定し、ソフトウェア信頼性解析例を示す。まず、図 2 に、推定された式 (13) の平均値関数を示す。ここで、パラメータ推定値は、それぞれ、 $\hat{\omega} = 34.038$, $\hat{b} = 0.172$, $\hat{\alpha}_1 = 0.669$, および $\hat{\alpha}_2 = 0.249$ であった。図 2 より、推定された総期待発見フォールト数の時間的挙動が実測データの挙動に沿いながらチェンジポイントにおいて変化していることがわかる。また、推定されたテスト環境係数から、チェンジポイント後の方がソフトウェア故障発生時間間隔はチェンジポイント後に比べて相対的に狭くなっていることもわかる。次に、図 3 および図 4 に、それぞれ、推定された期待残存フォールト数とソフトウェア信頼度関数の時間的挙動を示す。ここで、期待残存フォールト数とは、任意のテスト時刻 t においてソフトウェア内に潜在する総期待フォールト数を表し、 $M(t) \equiv E[\bar{N}(t)] = E[N(\infty) - N(t)] = \omega - \Lambda(t)$ のように与えられる。また、ソフトウェア信頼度関数とは、任意のテスト時刻 t までソフトウェアが実行されているという条件の下で、その後の時間区間 $(t, t+x] (t \geq 0, x \geq 0)$ において、ソフトウェア故障が発生しない条件付き確率として定義され、NHPP の性質からソフトウェア信頼度関数 $R(x|t)$ は、 $R(x|t) = \exp[-\{\Lambda(t+x) - \Lambda(t)\}]$ のように導出される。図 3 から、テスト終了時においてソフトウェアに潜在する総期待フォールト数 $\hat{M}(26)$ は約 1 つと推定できる。ならびに、図 4 から、運用段階においてテスト工程と同じ環境にてソフトウェアが運用されていると仮定した場合、出荷後 1 週目におけるソフトウェア信頼度 $\hat{R}(1.0|26)$ は約 0.977 と推定できる。

5 おわりに

本研究では、ソフトウェア故障発生現象に関する確率諸量のチェンジポイントでの変化をテスト環境関数を用いながら表現すると共に、ソフトウェア信頼度成長過程に影響を与える要因の複数回の変化を考慮した SRGM を構築するための枠組みを与えた。また、提案したモデリング枠組みに基づいて具体的な SRGM

を構築して、実測データを用いた信頼性解析例も示した。今回議論したアプローチでは、テスト環境関数を導入することでチェンジポイント前後における確率諸量の関係性を特徴付けることができ、かつ、複数回の変化を考慮した場合に発生するモデルの複雑化およびパラメータの増大を比較的少なく抑えることができる。今後は、チェンジポイントでのソフトウェア信頼度成長過程の変化を考慮した既存のモデルとソフトウェア信頼性評価精度に関する比較を行いながら、提案手法の有効性を検証していく必要がある。

参考文献

- [1] C.Y. Huang, “Performance analysis of software reliability growth models with testing-effort and change-point,” *Journal of System and Software*, Vol. 76, No. 2, pp. 181–194, 2005.
- [2] S. Inoue and S. Yamada, “Software reliability measurement with change-point,” *Proc. International Conference on Quality and Reliability (ICQR 2007)*, pp. 170–175, 2007.
- [3] S. Inoue and S. Yamada, “Optimal software release and change-point problems of software systems,” *Proc. of 16th ISSAT Intern. Conf. Reliab. Qual. Des.*, 2010, pp. 105–109.
- [4] N. Langberg and N.D. Singpurwalla, “A unification of some software reliability models,” *SIAM Journal on Scientific Computing*, Vol. 6, No. 3, pp. 781–790, 1985.
- [5] J.D. Musa, D. Iannio, and K. Okumoto, *Software Reliability: Measurement, Prediction, Application*. McGraw-Hill, New York, 1987.
- [6] H. Okamura, T. Dohi, and S. Osaki, “A reliability assessment method for software products in operational phase? — Proposal of an accelerated life testing model,” (in Japanese), *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. J83-A, No. 3, pp. 294–301, 2000.
- [7] H. Pham, *Software Reliability*. Springer-Verlag, Singapore, 2000.
- [8] S. Yamada and S. Osaki, “Software reliability growth modeling: Models and applications,” *IEEE Transaction on Software Engineering*, Vol. SE-11, No. 12, pp. 1431–1437, 1985.
- [9] M. Zhao, “Change-point problems in software and hardware reliability,” *Communication in Statistics — Theory and Method*, Vol. 22, No. 3, pp. 757–768, 1993.
- [10] F.Z. Zou, “A change-point perspective on the software failure process,” *Software Testing, Verification and Reliability*, Vol. 13, No. 2, pp. 85–93, 2003.

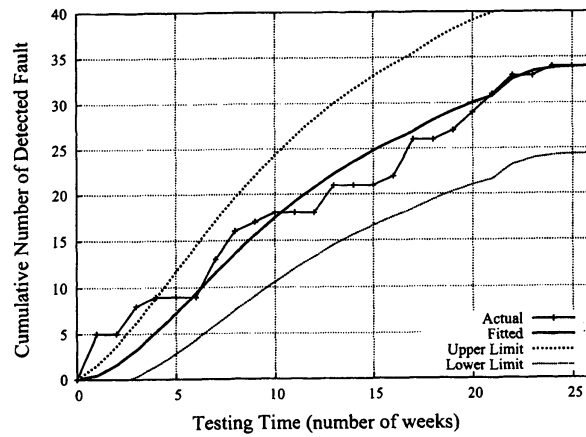


図 2： 推定された平均値関数と 90% 信頼限界. (モデル B, $\tau_1 = 17$, $\tau_2 = 21$, and $\Lambda_1(t) = \omega\{1 - (1 + bt) \exp[-bt]\}$)

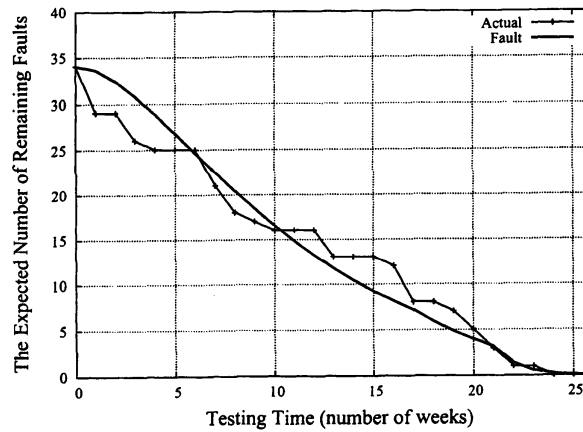


図 3： 推定された期待残存フォールト数. ($\tau_1 = 17$, $\tau_2 = 21$, and $\Lambda_1(t) = \omega\{1 - (1 + bt) \exp[-bt]\}$)

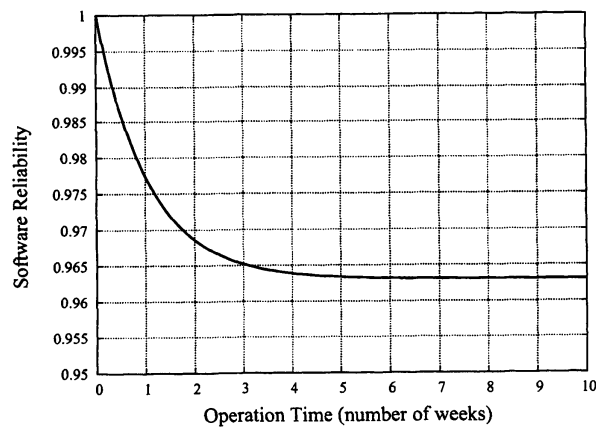


図 4： 推定されたソフトウェア信頼度関数. ($\tau_1 = 17$, $\tau_2 = 21$, and $\Lambda_1(t) = \omega\{1 - (1 + bt) \exp[-bt]\}$)