

テスト環境の変化を考慮した環境関数に基づくソフトウェア信頼性モデル

鳥取大学・大学院工学研究科 飛田 裕一 (Yuichi Tobita)[†]

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

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] は, ソフトウェア実行過程におけるソフトウェア故障発生頻度, または, ソフトウェア故障発生時間間隔を確率変数として捉え, テスト工程や運用段階において観測されるソフトウェアの信頼度成長過程を定量的に計測および評価する数理モデルである.

SRGM は, 通常, テスト実行時間上における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時間間隔) を基本的な確率量として取り扱い, ソフトウェア故障発生時間の確率的性質がテスト期間中を通じて同一であるという仮定に基づいて構築される. しかしながら, 開発管理面からの要因 (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など) によって, 実際のソフトウェア開発のテスト工程では, 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の統計的性質が著しく変化する現象がしばしば観測される. このような現象が発生する時刻はチェンジポイント (change-point, 以下 CP と略す) と呼ばれ, これまでに提案されている SRGM に基づいた信頼性評価の精度に影響を与える要因の 1 つとして考えられている [4, 5].

このような背景の下, 近年では, 上述した要因による CP 発生シナリオに基づいた SRGM に関する議論がなされている. 具体的には, CP 前後のソフトウェア故障発生率の違いのみを考慮した SRGM 構築枠組みに関する議論が多くなされている. しかしながら, 実際のテスト工程では, テスト期間中においてテスト環境が変化しようとも, テストされているソフトウェア自体は同じであるため, CP 前後におけるソフトウェア故障発生時間には何らかの関係性があるものとしてモデリング枠組みを考える方がより現実的であると言えよう. このような考え方は, 基本的に, 運用段階におけるソフトウェア信頼性評価手法の考え方 [3] とよく合致するものと考えられる. 本研究では, SRGM の一般化枠組みに基づいて, CP 発生前後のソフトウェア故障発生パターンの違いを考慮した SRGM の構築を行う. 最後に, 実測データを用いながら提案モデルの適用例を示し, 当該モデリング枠組みの有効性を簡単に議論する.

2 基本的な SRGM 構築枠組み

2.1 SRGM のモデリング枠組み

CP を考慮した SRGM を構築する際に, その基本となる一般的な SRGM のモデリング枠組みについて議論する. SRGM のモデリング枠組みは, 以下の 3 つの仮定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および除去される.
- (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, 独立かつ同一の確率分布 $F(t) \equiv \Pr\{T \leq t\} = \int_0^t f(x)dx$ に従う. ここで, $f(t)$ は確率密度関数 (probability density function) 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非負の確率変数とする.

ここで, 確率過程 $\{N(t), t \geq 0\}$ を, テスト時刻 t までに発見された総フォールト数を示すものとする. このとき, 上述の仮定より, 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,

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

のように定式化される。式 (1) において、初期潜在フォールト数 N_0 が平均 $\omega (> 0)$ のポアソン分布：

$$\Pr\{N_0 = n\} = \frac{\omega^n}{n!} \exp[-\omega], \quad (2)$$

に従うとき、式 (1) は、

$$\begin{aligned} \Pr\{N(t) = m\} &= \sum_n \frac{n!}{m!(n-m)!} \{F(t)\}^m \{1 - F(t)\}^{n-m} \frac{\omega^n}{n!} \exp[-\omega] \\ &= \frac{\{\omega F(t)\}^m}{m!} \exp[-\omega F(t)] \quad (m = 0, 1, 2, \dots), \end{aligned} \quad (3)$$

となる。式 (3) は、平均値関数 (mean value function) が $E[N(t)] = \omega F(t)$ である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。また、平均値関数とは、時間区間 $(0, t]$ において発見される総期待フォールト数を表す。

例えば、ハザードレートが一定の $\lambda (> 0)$ (一定型フォールト発見率) のとき、ソフトウェア故障発生時間分布は、

$$\begin{aligned} F(t) &= 1 - \exp\left[-\int_0^t z(x) dx\right] \\ &= 1 - \exp[-\lambda t] \end{aligned} \quad (4)$$

となり、テスト時間区間 $(0, t]$ において発見される総フォールト数を表す NHPP の平均値関数は、

$$\begin{aligned} E[N(t)] &\equiv H(t) \\ &= \omega F(t) = \omega(1 - \exp[-\lambda t]), \end{aligned} \quad (5)$$

となる。式 (5) を NHPP の平均値関数とする SRGM は、指数形 (exponential) SRGM (Goel-Okumoto モデル) [1] と本質的に等価となる。

3 チェンジポイントを考慮した SRGM

前述した基本的仮定の 1 つである (A2) をテスト工程における CP を考慮しながら拡張して、一般化モデリング枠組みに基づいた SRGM 構築枠組みを議論する。ただし、テスト期間 $(0, T]$ を通じて CP (τ) ($0 < \tau < T$) が 1 回のみ発生する場合について議論する。図 1 に、今回の議論で用いるソフトウェア故障発生現象に関する確率諸量を示す。CP 発生前、および発生後において i ($= 1, 2, \dots$) 番目のソフトウェア故障が観測される時刻をそれぞれ X_i 、および Y_i とし、ソフトウェア故障発生時間間隔をそれぞれ、 $S_i = X_i - X_{i-1}$ 、および $T_i = Y_i - Y_{i-1}$ 、その時間間隔に対する確率分布をそれぞれ、 $\Pr\{S_i \leq t\} = F_i(t)$ 、および $\Pr\{T_i \leq t\} = G_i(t)$ とし、以下の仮定を設定する。

$$Y_i = \alpha(X_i), \quad (6)$$

$$T_i = \alpha(S_i), \quad (7)$$

$$G_i(t) = F_i(\alpha^{-1}(t)). \quad (8)$$

ここで、CP 前後におけるソフトウェア故障発生現象の違いを表す環境関数として以下の式を設定する。

$$\alpha(t) = \alpha t^\beta \quad (t \geq 0, \alpha > 0, \beta > 0). \quad (9)$$

いま、CP 発生までに $n (> 1)$ 個のフォールトが発見され、それぞれのフォールト発見時刻が、 $0 < x_1 < x_2 < \dots < x_n \leq \tau$ であるとする。CP 発生前までの時間区間 $(0, t]$ ($t \leq \tau$) において発見および除去される

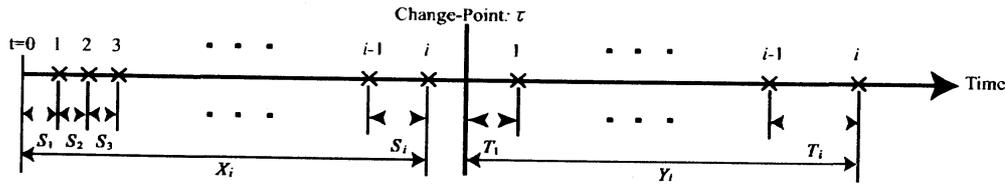


図 1: チェンジポイント発生前後におけるソフトウェア故障発生現象に関する確率諸量.

総累積フォールト数が平均値関数 $H(t)$ をもつ NHPP に従う場合, CP 発生時刻 τ 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 T_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.

$$\begin{aligned} \Pr\{T_1 > t\} &= \frac{\Pr\{S_{n+1} > \tau - x_n + \ell\sqrt{t/\alpha}\}}{\Pr\{S_{n+1} > \tau - x_n\}} \\ &= \frac{\exp\{-[H(\tau + \ell\sqrt{t/\alpha}) - H(x_n)]\}}{\exp\{-[H(\tau) - H(x_n)]\}}. \end{aligned} \quad (10)$$

式 (10) より, CP 発生後 ($t \geq \tau$) において発見される総期待フォールト数を表す NHPP の平均値関数は,

$$\begin{aligned} H_0(t) &= -\log[\Pr\{\sigma_\tau > t - \tau\}] \\ &= -\log[\bar{G}_i(t - \tau)] \\ &= H(\tau + \sqrt[\ell]{\frac{t - \tau}{\alpha}}) - H(\tau), \end{aligned} \quad (11)$$

と導出される. ここで, $\bar{G}_i(t)$ は確率分布関数 $G_i(t) \equiv \Pr\{T_1 \leq t\}$ の余関数であり $\bar{G}_i(t) \equiv 1 - G_i(t)$ である. 以上の議論から, 任意のテスト時刻 t までに発見・除去された $\Lambda(t)$, CP 発生前までに発見・除去された総期待フォールト数を表す NIIPP の平均値関数を $\Lambda_1(t)$, また, CP 発生後において発見・除去されたそれを $\Lambda_2(t)$ とすると CP を考慮した NHPP の平均値関数は,

$$\Lambda(t) = \begin{cases} \Lambda_1(t) = H(t) & (0 \leq t \leq \tau) \\ \Lambda_2(t) = H_0(t) + H(\tau) & (t > \tau) \\ & (= H(\tau + \sqrt[\ell]{\frac{t - \tau}{\alpha}})), \end{cases} \quad (12)$$

のようになる.

式 (12) より, CP 前において観測されたデータに適したソフトウェア故障発生時間分布を与えることによって, CP を考慮した SRGM を構築することができる.

4 本研究での信頼性解析

4.1 評価基準

本研究では, 今回提案した SRGM について, 平均偏差平方和 (mean squared errors, 以下 MSE と略す) とコルモゴロフ - スミルノフ (Kolmogorov-Smirnov) 適合度検定法 (以下, K-S 検定法と略す) を評価基準としたモデルの選択を行う.

MSE は, 実測値と推定値との 2 乗誤差をデータ数で平均化したものであり, 実測値と推定値の誤差が小さい程, 適合性が高いモデルと言える. ここで, 一定のテスト時刻 t_k , ($k = 1, 2, \dots, K$) までに発見された総フォールト数の実測値 y_k に関する K 組のフォールト発見数データ: (t_k, y_k) ($k = 0, 1, 2, \dots, K$) が観測されているものとする, MSE は,

$$\text{MSE} = \frac{1}{K} \sum_{k=1}^K \{y_k - \hat{y}(t_k)\}^2, \quad (13)$$

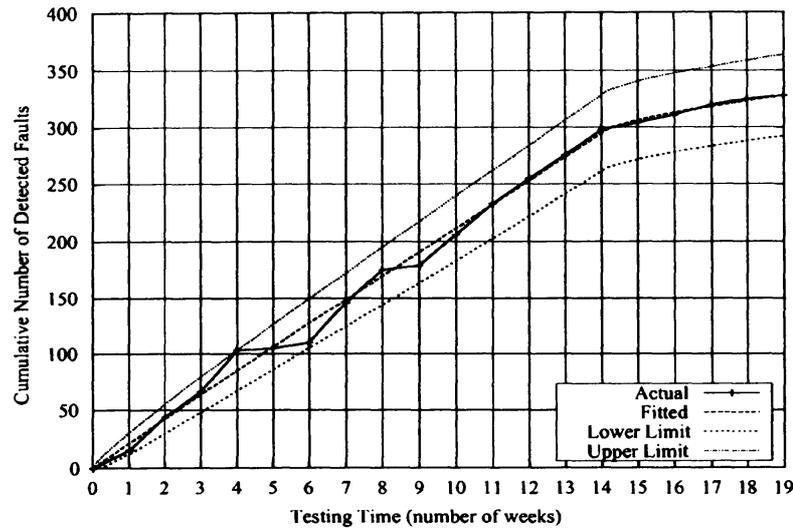


図 2: 推定された $\hat{H}(t)$ とその 95% 信頼限界 ($\hat{\alpha} = 2.5, \hat{\beta} = 1.5, \hat{\tau} = 14$).

として算出される. ここで, $\hat{y}(t_k)$ はテスト時刻 $t_k (k = 1, 2, \dots, K)$ までに発見されたフォールト数の推定値を表す.

また, K-S 検定法は, モデルに基づいた理論分布が実測データに基づいた経験分布にどの程度適合しているかを判別する統計的検定手法であり, 観測データ数が少ない場合でも効果的であるとされている. フォールト発見数データ $(t_k, y_k) (k = 1, 2, \dots, K)$ が観測されたとき,

$$D = \max_{1 \leq i \leq n-1} \{D_i\} \quad (14)$$

$$D_i = \max \left\{ \left| \frac{H(t_i)}{H(t_n)} - \frac{y_i}{y_n} \right|, \left| \frac{H(t_i)}{H(t_n)} - \frac{y_{i-1}}{y_n} \right| \right\}$$

となる K-S 検定統計量を計算する. フォールト発見数データを扱う場合, データから計算された検定統計量 D と有意水準 δ , 自由度 n のときの棄却限界 $D_{n,\delta}$ と比較される. このとき, $D < D_{n,\delta}$ ならば, 有意水準 δ で観測データに対して提案した NHPP モデルは適合していると判断する. 一方, $D > D_{n,\delta}$ となってモデルが適合していなければ, 他のモデルを適用する.

本研究では, 上記の 2 つの評価基準に基づき提案モデルの適合性評価を行う. 具体的には, MSE を 1 番目の評価基準とし, MSE の値が最小となったモデルには, 2 番目の評価基準である K-S 検定法を実施する. 有意であればそのモデルを適用する.

4.2 ソフトウェア信頼度関数

ソフトウェア信頼性評価尺度は, 定量的な信頼性評価を行う上で有用な評価尺度として知られている. その中でも, ソフトウェア信頼度関数 (software reliability function) は, 任意のテスト時刻 t までテストが進行し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確率と定義される. したがって, ソフトウェア信頼度関数 $R(x|t)$ は,

$$R(x|t) = \exp[-\{H(t+x) - H(t)\}], \quad (15)$$

のように導出される.

5 モデルパラメータの推定

本研究では, 最尤法に基づいて提案モデルに含まれるパラメータを推定する. ここで, 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 K 組のフォールト発見数データ $(t_k, y_k) (k =$

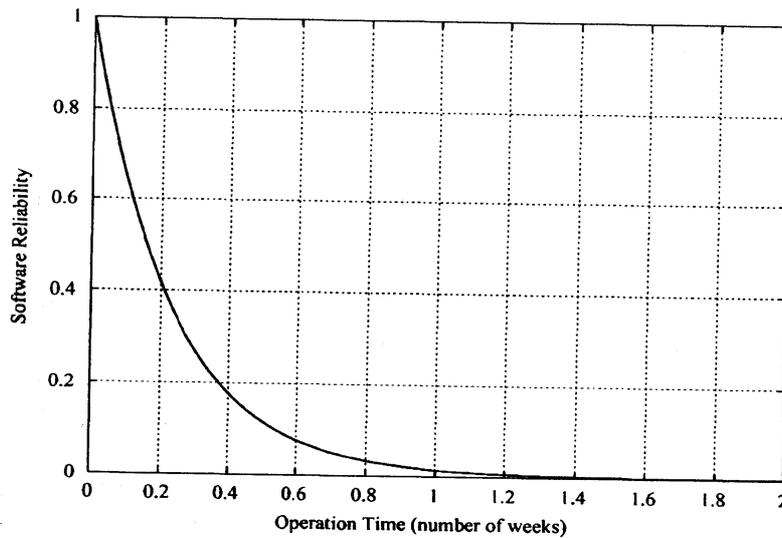


図 3：推定されたソフトウェア信頼度関数 ($\hat{\alpha} = 2.5, \hat{\beta} = 1.5, \hat{\tau} = 14$) .

$0, 1, 2, \dots, K$ が観測されたものとする. このとき, τ, α および β が所与の下で, 確率過程 $\{N(t), t \geq 0\}$ に関する対数尤度関数 $\ln L(\theta | \tau, \alpha, \beta)$ は,

$$\begin{aligned} & \ln L(\theta | \tau, \alpha, \beta) \\ &= \sum_{k=1}^K (y_k - y_{k-1}) \ln [H(t_k; \theta | \tau, \alpha, \beta) - H(t_{k-1}; \theta | \tau, \alpha, \beta)] - H_i(t_K; \theta | \tau, \alpha, \beta) - \sum_{k=1}^K \ln [y_k - y_{k-1}], \quad (16) \end{aligned}$$

のように導出される. ここで, $L(\theta | \tau)$ は, 確率過程 $\{N(t), t \geq 0\}$ に関する同時確率関数もしくは同時尤度関数を示しており, θ はモデル内部に含まれるパラメータを表す. これより, 式 (16) から導出できる同時対数尤度方程式:

$$\frac{\partial \ln L(\theta | \tau, \alpha, \beta)}{\partial \theta} = 0, \quad (17)$$

を, モデル内部に含まれるパラメータに対して, それぞれ, 数値的に解くことにより, パラメータ推定値を得ることができる.

6 適用例

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対する適用例を示す. 本研究にて用いる実測データは, 19組のフォールト発見数データ $(t_k, y_k) (k = 0, 1, 2, \dots, 19; t_{19} = 19 \text{ (週)}, y_{19} = 328)$ [6] である. また, CP 前における瞬間ソフトウェア故障発生率が b (一定), すなわち, ソフトウェア故障発生時間分布がパラメータ b の指数分布に従う場合を考える. このとき, CP 前の NHPP の平均値関数は指数形 SRGM に従い, モデル内部に含まれるパラメータ ω および b の推定値 $\hat{\omega}$ および \hat{b} は最尤法を用いて推定できる. ただし, α, β および τ は所与として, それらのパラメータを推定することにする.

本研究では, 一例として, $\alpha = 2.5, \beta = 1.5$ と仮定した場合の適用例を与える. 図 2 に, 上記の方法に基づいて推定された NHPP の平均値関数とその 95% 信頼限界を示す. なお, CP は, 実測データに対する平均偏差平方和 (MSE) が最も小さいときの値を適用している. 図 2 から, 提案モデルは, CP において実測データが示す信頼度成長曲線の変化をうまく記述できていることがわかる. また, 図 3 に, 推定されたソフトウェア信頼度関数 $\hat{R}(x|19)$ を示す. 図 3 より, 当該ソフトウェアがテスト開始後 19 週目にリリース

(出荷) され運用段階においてもテスト工程と同様の環境で使用した場合、リリース後 0.5 週目におけるソフトウェア信頼度は約 1.41714×10^{-2} と推定される。

7 おわりに

本研究では、テスト環境の変化に起因して観測されるソフトウェア故障発生現象およびフォールト発見事象の変化を考慮した SRGM の構築を行った。具体的には、当該現象はソフトウェアのテスト環境と運用段階における使用環境との違いによく似ているため、文献 [3] において議論されたアプローチをベースとしながら、ソフトウェア信頼性評価のための CP モデルを構築した。当該アプローチを適用した利点としては、CP を考慮した従来のモデル [4,5] と比較して、CP 前後において変化するソフトウェア故障発生時間の関係性を陽にモデルへと反映できる点である。したがって、CP 前におけるソフトウェア故障発生現象を具体的に特徴付けると共に、CP 前後におけるソフトウェア故障発生時間の関係性を解析的に与えることによって、比較的容易にモデルを構築することができる。

今後は、実測データに基づきながら、適切かつ現具的なテスト環境係数の構築を行う必要がある。また、今回議論したアプローチについて、CP を考慮したソフトウェア信頼性モデルとしての適合性および有用性についての検証を行う必要がある。

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.
- [2] 山田茂，高橋宗雄，ソフトウェアマネジメント入門—ソフトウェア品質の可視化と評価法—，共立出版，東京，1993.
- [3] 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. J-83-A，No. 3，pp. 294-301，2000年3月.
- [4] 井上真二，山田茂，“チェンジポイントを考慮したソフトウェア信頼性評価法に関する一考察”，電子情報通信学会技術研究報告 [信頼性]，Vol. 107，No. 43，pp. 25-30，2007年5月.
- [5] S. Inoue and S. Yamada，“Software reliability measurement with change-point,” *Proceedings of the Fifth International Conference on Quality and Reliability. (ICQR2007)*, Chiang Mai, Thailand, November 5-7, 2007, pp.170-175.
- [6] M. Ohba，“Software reliability analysis models,” *IBM Journal of Research and Development*，Vol. 28，No. 4，pp. 428-443, 1984.