

分散ソフトウェア開発環境に対する 一般化確率微分方程式モデルに関する一考察

入会申請中 鳥取大学 *内田 雅也[†] UCHIDA Masaya
02302815 鳥取環境大学 田村 慶信^{††} TAMURA Yoshinobu
01702425 鳥取大学 山田 茂^{†††} YAMADA Shigeru

E-mail: [†]98t7006@sse.tottori-u.ac.jp, ^{††}tamura@kankyo-u.ac.jp, ^{†††}yamada@sse.tottori-u.ac.jp

1 はじめに

ソフトウェアの分散開発やソフトウェアそのものの部品化などにより、現在のソフトウェアの開発環境は多数のワークステーションを相互に接続した分散型開発環境に変わりつつある。

本研究では、分散開発環境に対するソフトウェア信頼性評価法として、確率微分方程式を導入し、ソフトウェアコンポーネントを組み立てることによって新たなソフトウェアを開発する場合を考える。このとき、新たに一般化ソフトウェア信頼度成長モデルを構築し、ソフトウェアの出荷時期を見積もる最適リリース問題について議論する。特に、要求信頼性を満足するとともに総期待ソフトウェアコストを最小にする最適リリース時刻に関して議論する。

2 分散開発環境に対する一般化確率微分方程式モデル

分散開発環境におけ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) として、総合テスト工程におけるフォールト発見過程が連続的に変動していく確率過程と考えてモデル化した確率微分方程式に基づく SRGM が提案されている。

本研究で用いる確率微分方程式を用いた一般化 SRGM は、ソフトウェアコンポーネントを $(n+m)$ 個統合したものであると仮定し、フォールト発見過程は次式のように表される。

$$E[M_s(t)] = m_0 \left\{ \sum_{i=1}^{n+m} \frac{p_i e^{-b_i t} (1+c_i)}{1+c_i e^{-b_i t}} \right\} e^{\frac{\sigma^2}{2} t}. \quad (1)$$

ここで、 $M_s(t)$ は任意のテスト時刻 t におけるソフトウェア内の残存フォールト数を表し、式(1)はその期待値を表す。また、 m_0 はテスト開始前の潜在フォールト数、 $b_i (i=1, 2, \dots, n+m)$ は i 番目のコンポーネントに対する残存フォールト 1 個当りのフォールト発見率、パラメータ $p_i (i=1, 2, \dots, n+m)$ は i 番目のコンポーネントに対する重み、すなわちテストの重要度を表している。また、 $c_i (i=1, 2, \dots, n+m)$ は $(1-l_i)/l_i$ で表され、 l_i をフォールト発見能力に関するテスト習熟係数という。 σ は定数パラメータである。

テスト時刻 $t (t \geq 0)$ までに発見されたフォールト数 $N_s(t) (= m_0 - M_s(t))$ が x である条件の下で、微小時間区間 $\Delta t (t > 0)$ 後において発見フォールト数が増加しない確率、つまり不完全デバッグの発生確率と考えられる確率量を式(2)のように導出できる [1]。

$$\Pr[N_s(t + \Delta t) \leq x | N_s(t) = x] \\ = \Phi \left[\frac{\log \left\{ \frac{\sum_{i=1}^{n+m} \frac{p_i e^{-b_i (t+\Delta t)} (1+c_i)}{1+c_i e^{-b_i (t+\Delta t)}}}{\sum_{i=1}^{n+m} \frac{p_i e^{-b_i t} (1+c_i)}{1+c_i e^{-b_i t}}} \right\}}{\sigma \sqrt{\Delta t}} \right]. \quad (2)$$

ここで、 $\Pr[\cdot]$ は確率を表し、 $\Phi(\cdot)$ は標準正規分布関数を表す。

3 最適リリース問題

3.1 総期待ソフトウェアコスト

総期待ソフトウェアコストを定式化するために、以下のコストパラメータを定義する。

- $c_{1,i}$: コンポーネント i の単体テストで発見されるフォールト 1 個当りの修正コスト ($c_{1,i} > 0$),
- $c_{2,i}$: コンポーネント i の単体テストでの単位時間当りのテストコスト ($c_{2,i} > 0$),
- c_{1c} : 総合テストで発見されるフォールト 1 個当りの修正コスト ($c_{1c} > 0$),
- c_{2c} : 総合テストでの単位時間当りのテストコスト ($c_{2c} > 0$),
- c_{3c} : 運用後に発見されるフォールト 1 個当りの保守コスト ($c_{3c} > 0, c_{3c} > c_{1i}, c_{3c} > c_{1c}$).

さらに、以下のような各コンポーネントのテストに要する期待コストの式が得られる。

$$C_i(t_i) = c_{1,i} H_i(t_i) + c_{2,i} t_i + G_i(t_i) \\ (i = 1, 2, \dots, n, \dots, n+m). \quad (3)$$

ここで、 $H(t)$ は非同次ポアソン過程に基づく習熟 S 字形 SRGM[2] の平均値関数を表し、

$$H(t) = \frac{a(1 - e^{-bt})}{1 + c \cdot e^{-bt}} \quad (a > 0, b > 0, c > 0), \quad (4)$$

により与えられる。式(4)において、 a はテスト開始前に潜在している総期待フォールト数、 b は残存フォールト 1 個当りのフォールト発見率、 c は $(1-l)/l$ により定義されるもので、 l をフォールト発見能力に関するテスト習熟係数という。 $G(t)$ は、単体テストから総合テスト工程への引渡し遅延によるペナルティコスト関数を表す。

よって、総合テストに要する期待コスト、および運用後の保守コストは、

$$C_c(t_c) = c_{1c} E[N_s(t_c)] + c_{2c} t_c, \quad (5)$$

$$C_d(t_c) = c_{3c} E[M_s(t_c)], \quad (6)$$

と表される。したがって、総期待ソフトウェアコストは、次式のように表せる。

$$C(t_1, \dots, t_n, \dots, t_{n+m}, t_c) \\ = \sum_{i=1}^{n+m} C_i(t_i) + C_c(t_c) + C_d(t_c). \quad (7)$$

式(7)を最小にする時刻 t_c が最適リリース時刻 t_c^* となる。

3.2 信頼度を考慮した問題

次に、信頼度を十分に満足するとともに総期待ソフトウェアコストを最小にするような最適リリース時刻について議論する。各ソフトウェアコンポーネントは、単体テスト工程である程度の信頼性要求を満足するものとする。よって、総合テスト工程では、ソフトウェアコンポーネントを結合した後のソフトウェアの信頼度を評価する。特に、そのソフトウェア信頼性を評価するために、次式の変動係数を用いる。

$$CV(t) \equiv \frac{\sqrt{\text{Var}[N_s(t)]}}{E[N_s(t)]} \quad (8)$$

ここで、 $\text{Var}[N_s(t)]$ は、テスト時刻 t における残存フォールト数の分散を表す。

変動係数目標を CV_0 とすると信頼度に対する最適リリース問題は次のように定式化される。

$$\text{Minimize } C(t_c) \quad \text{subject to } CV(t) \leq CV_0. \quad (9)$$

但し、総期待ソフトウェアコストを最小にするのみに注目した最適リリース時刻 t_c^* から、以下のような条件が成り立つ。

$$\begin{cases} t_c = t_c^* & (CV(t_c^*) \leq CV_0) \\ t_c = t_c' & (CV(t_c^*) > CV_0). \end{cases} \quad (10)$$

つまり、目標変動係数 CV_0 が最適リリース時刻 t_c^* において満足されていないならば、テスト時刻を延長することによって満足するテスト時刻 t_c' を見つけ、最適リリース時刻と定義する。

4 数値例

数値例を示すにあたり、実際の総合テスト工程において観測されたデータを適用する。

図1は、式(2)において時間区間 $\Delta t = 1$ のときの不完全デバッグの発生確率を表す。図1から、テスト工程の初期段階においては、不完全デバッグ率は低いが、テスト工程の中盤になると、不完全デバッグ率は大きくなる。これは、テストの進捗とともにテストケースの網羅率の変化やコンポーネント間の相互作用などに影響されたのではないかと考えられる。

次に、推定された総期待ソフトウェアコストの時間変化の様子を図2に示す。図2から、総期待ソフトウェアコストの

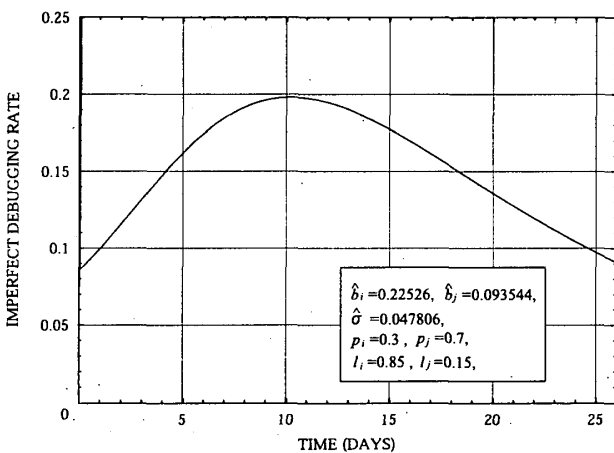


図1: 不完全デバッグの発生確率。

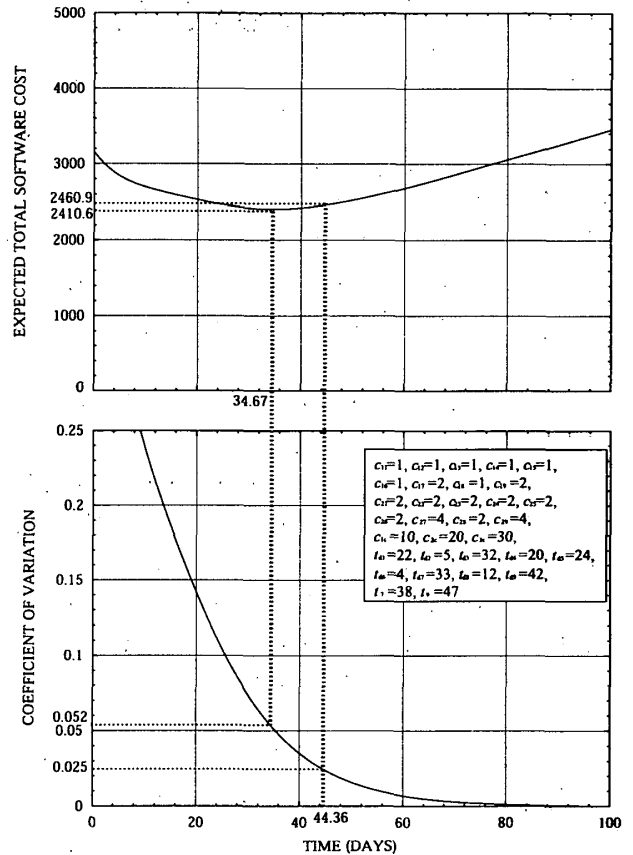


図2: 推定された総期待ソフトウェアコスト。

みを考慮した場合には、最適リリース時刻は $t^* = 34.67$ と推定され、このときの総期待ソフトウェアコストは 2410.6 となった。また、目標信頼度 (変動係数) $CV_0 = 0.025$ として信頼度を考慮した場合には、最適リリース時刻は $t' = 44.36$ へと延長する必要がある。このときの総期待ソフトウェアコストも増加して 2460.9 となることかわかる。

5 おわりに

本研究では、確率微分方程式を用いた一般化 SRGM に対して、ソフトウェアフォールトを除去できない、またはフォールト修正時に新たなフォールトを作り込んでしまう不完全デバッグの発生確率を導出した。さらに、最適リリース問題として、要求されるソフトウェア信頼性を考慮するとともに総期待ソフトウェアコストを最小にする最適リリース時刻を求めた。これにより、より現実的なリリース時刻を推定することができた。

謝辞

本研究の一部は、日本学術振興会科学研究費補助金基盤研究(C)(2) (課題番号 15510129) の援助を受けたことを付記する。

参考文献

- [1] 田村慶信, 木村光宏, 山田茂, “分散開発環境に対するソフトウェア信頼度成長モデル: 確率微分方程式アプローチとその推定,” 日本応用数理学会論文誌, vol. 11, no. 3, pp. 121-132, 2001年9月。
- [2] 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。