

分散ソフトウェア開発環境の統合テスト工程における
確率微分方程式モデルに基づく信頼性評価に関する一考察

02302815 鳥取環境大学 *田村 慶信† TAMURA Yoshinobu
01702425 鳥取大学 山田 茂†† YAMADA Shigeru
01507374 富士通周辺機株式会社 藤原 隆次††† FUJIWARA, Takaji

E-mail: †tamura@kankyo-u.ac.jp, ††yamada@sse.tottori-u.ac.jp, †††takaji@fpe.fujitsu.com

1 はじめに

ソフトウェア製品の開発プロセスにおけるテスト進捗管理や出荷品質の把握のための信頼性評価を行うアプローチとして、ソフトウェア故障の発生現象を不確定事象としてとらえて確率・統計論的に取り扱う方法がとられている。その1つがソフトウェア信頼度成長モデル (software reliability growth model, 以下 SRGM と略す) である [1]。これまでに提案されてきたソフトウェア信頼度成長モデルは、テスト工程でのフォールト発見過程を離散過程として記述するものが多かった。しかし、ソフトウェアの需要の急速な増加とともに、ソフトウェアシステムも複雑化・多様化・大規模化している。こうした大規模ソフトウェアシステムが対象となる場合、フォールト発見過程を連続的に変動していく過程と考えてモデル化することが可能となる。代表的なものとして確率微分方程式に基づくソフトウェア信頼度成長モデルが提案されている [2]。特に、分散開発環境のように対象とするソフトウェアシステムが大規模な場合や、中に含まれる記述プログラム言語の種類やシステム内部の構成要素数が多い場合、各ソフトウェアコンポーネント間の相互作用もより一層顕在化することから、テスト工程における発見フォールト数の挙動が把握しにくくなる。こうした背景から、現在のところ分散開発環境における有効なテスト方法は提案されていない。

本研究では、確率微分方程式を導入することにより、分散開発環境の総テスト工程において信頼性予測可能なソフトウェア信頼度成長モデルを構築する。特に、サブシステム内におけるテスト工程では、従来からの非同次ポアソン過程モデルを適用し、ソフトウェアコンポーネントの結合テスト工程に対しては確率微分方程式から導出されたソフトウェア信頼度成長モデルを適用する。これにより、テスト工程の初期段階からの信頼性予測が可能となる。

2 分散開発環境における信頼性評価

2.1 各コンポーネントに対する信頼性評価

従来から、ソフトウェアの信頼性を定量的に評価する手法として、ソフトウェア信頼度成長モデルによる方法がとられている。中でも非同次ポアソン過程 (nonhomogeneous Poisson process, 以下 NHPP と略す) モデルは、実利用上極めて有効でありモデルの簡潔性が高いゆえにその適用性も高く、実際のソフトウェア信頼性評価に広く応用されている。この NHPP モデルは、所定の時間区間内に発見されるフォールト数や発生するソフトウェア故障数を観測して、これらの個数を数え上げる計数過程 $\{N(t), t \geq 0\}$ を導入し、以下の式で与えられる確率変数すなわちポアソン過程を仮定するソフトウェア信頼度成長モデルである [1]。

$$\Pr\{\dot{N}(t) = n\} = \frac{\{H(t)\}^n}{n!} \exp[-H(t)] \quad (n = 0, 1, 2, \dots) \quad (1)$$

ここで、 $\Pr\{\cdot\}$ は確率を表し、 $H(t)$ 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、すなわち $N(t)$ の期待値を表し、NHPP の平均値関数と呼ばれる。

本テスト工程では、各コンポーネントについて累積発見フォールト数データの成長曲線の形状により、以下に示す NHPP モデル [1] を適用する。

- 指数形ソフトウェア信頼度成長モデル
- 遅延 S 字形ソフトウェア信頼度成長モデル
- 習熟 S 字形ソフトウェア信頼度成長モデル
- テスト労力依存型ソフトウェア信頼度成長モデル

さらに、モデルに含まれる未知パラメータの推定方法として最尤法を適用する。上記の NHPP モデルから、種々のソフトウェア信頼性評価のための定量的尺度を導出できる。

本研究では、各 NHPP モデルから導出されるフォールト発見率

$$b_i(t) = \frac{dH_i(t)}{a - H_i(t)} \quad (2)$$

を各ソフトウェアコンポーネントの特徴を近似的に表すものと仮定する。ここで、 $H_i(t)$ は i 番目のソフトウェアコンポーネントに対して適用された NHPP モデルにおける平均値関数であり、時間区間 $(0, t]$ において発見される期待累積フォールト数を表す。また、 a は、各コンポーネントにおいて最終的に発見される総期待フォールト数を表す定数パラメータである。

2.2 コンポーネント結合段階における信頼性評価

従来からの分散開発環境を対象としたソフトウェア信頼度成長モデルは総合テスト工程を対象としたものであった。本研究では、ソフトウェアコンポーネント結合段階におけるテスト工程に対して、確率微分方程式から導出されたソフトウェア信頼度成長モデルを提案する。時刻 $t = 0$ でコンポーネント結合段階におけるソフトウェアテストが開始され、任意の時刻 t におけるソフトウェア内の残存フォールト数 $\{M(t), t \geq 0\}$ は以下の常微分方程式によって記述されるものと仮定する。

$$\frac{dM(t)}{dt} = -b(t)M(t) \quad (3)$$

ここで、 $b(t) (> 0)$ は時刻 t における残存フォールト 1 個当りのフォールト発見率を表す。実際には、1 個のフォールトが発見される可能性はテスト期間を通じて必ずしも一定とは限らない。また、分散開発環境の特徴であるソフトウェアコンポーネント間の相互作用を近似的に表現するために、フォールト発見率 $b(t)$ に不規則性を導入すると、式 (3) は、

$$\frac{dM(t)}{dt} = - \left\{ \int_0^t \sum_{i=1}^n \frac{b_i(s)}{n} ds + \sigma\gamma(t) \right\} M(t) \quad (4)$$

となる。ここで、 $\sigma(>0)$ は定数パラメータであり、 $\gamma(t)$ は標準化された Gauss 型白色雑音である。 n はソフトウェアのコンポーネント数を表す。さらに、 $b_i(s)$ は i 番目のコンポーネントに対するサブシステム内のテスト時刻 s におけるフォールト発見率を表す。

式 (4) の確率微分方程式を Itô の公式を用いて解くと、

$$M(t) = m_0 \cdot \exp \left[- \int_0^t \sum_{i=1}^n \frac{b_i(s)}{n} ds - \sigma W(t) \right], \quad (5)$$

となる。ここで、 m_0 はソフトウェア内の初期残存フォールト数を表す。また、 $\{W(t), t \geq 0\}$ は Wiener 過程であり、 $E[W(t)] = 0$, $\text{Var}[W(t)] = \sigma^2 t$ を満たす。

3 数値例

3.1 各コンポーネントに対する信頼性評価

実際のテスト工程において観測されたデータを適用した数値例を示す。ここに示す数値例は、実際にある企業で開発されたプロジェクトデータに基づいている。本研究で用いたデータは、7つのソフトウェアコンポーネントから構成されたソフトウェアシステムのテスト工程から採取されたものである。また、テスト時間 t の測定単位は日である。

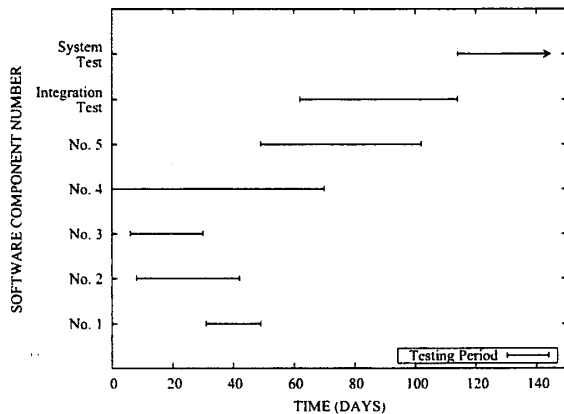


図 1: 各ソフトウェアコンポーネントのテスト期間。

各ソフトウェアコンポーネントのテスト期間を図 1 に示す。本研究で対象とするテストの全体の流れとしては、各コンポーネントの単体テストはプログラム作成者がテストまで一貫して責任を持ち実施されるものとし、システム全体の統合 (Integration Test) から本番運用確認までの総合テスト (System Test) の開始時期およびその期間は、開発されるソフトウェアシステムの納期やテストの進捗度合いに応じてシステム開発管理者により決定されるものとする。まず、各コンポーネントに対するモデルの適用基準としては平均偏差 2 乗和 (mean squared errors, 以下 MSE と略す) を適用した。各コンポーネントに対する MSE の推定結果を表 1 に示す。

3.2 コンポーネント結合段階における信頼性評価

次に、ソフトウェアコンポーネントの結合段階を対象とした信頼性評価結果の一例として、式 (5) における推定された残存フォールト数のサンプルパスを図 2 に示す。各コンポーネントに対するテスト工程において検出できなかったフォールトの総和は 5 個である。統合テスト終了時刻における総期待残存フォールト数は 8.8166 であることから、

表 1: 各コンポーネントに対する MSE の推定結果。

	指数形 SRGM	遅延 S 字形 SRGM	習熟 S 字形 SRGM	テスト労力依存型 SRGM
No.1	—	2.0507*	—	—
No.2	0.9702*	2.5230	—	—
No.3	1.7087*	3.0427	—	1.8898
No.4	4.0448*	11.600	—	—
No.5	—	3.3259	2.6973*	—

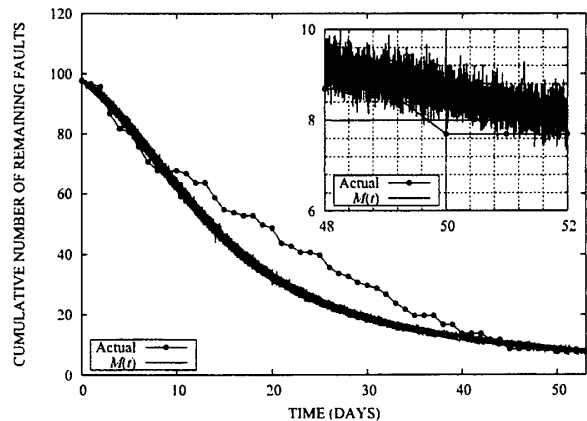


図 2: 推定された残存フォールト数のサンプルパス。

約 4 個程度のフォールトが各コンポーネント内に残存していることが分かる。

4 おわりに

本研究では、ソフトウェアコンポーネント結合段階におけるテスト工程に対して、確率微分方程式から導出されたソフトウェア信頼度成長モデルを適用した。これにより、分散化のために複雑化したソフトウェアシステム内における各ソフトウェアコンポーネント間の相互作用を包括できるものとする。また、従来に提案された分散開発環境を対象としたソフトウェア信頼度成長モデルは、本番運用確認までの総合テストの開始時期およびその期間、いわゆる総合テスト工程を対象としたものであった。本研究におけるモデルは、コンポーネント結合段階における統合テスト工程を対象としたものであることから、従来のモデルと比較してテスト工程の初期段階からの信頼性予測が可能となるものと考えられる。

謝辞

本研究の一部は、文部科学省科学研究費基盤研究 (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] 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] S. Yamada, M. Kimura, H. Tanaka, and S. Osaki, "Software reliability measurement and assessment with stochastic differential equations," *IEICE Trans. Fundamentals*, vol. E77-A, no. 1, pp. 109-116, Jan. 1994.