

## デバッグ率を考慮したソフトウェアの最適リリース政策

02005295 鳥取大学 \*井上 真二 INOUE, Shinji  
01702425 鳥取大学 山田 茂 YAMADA, Shigeru

## 1 はじめに

ソフトウェア信頼度成長モデル (software reliability growth model: SRGM) [1] は、テスト工程におけるフォールト発見事象やソフトウェア故障発生現象を、確率・統計論により記述して、ソフトウェアの信頼性を定量的に計測・評価するためのモデルである。その中で、フォールト修正時にフォールトが完全に除去できる完全デバッグ環境と、そうでない不完全デバッグ環境を仮定したモデルが存在する。実際のテスト工程においては、不完全デバッグ環境が実際のテスト環境を比較的忠実に反映した仮定であると言え、この不完全デバッグ環境を仮定したモデルは、不完全デバッグモデルと呼ばれている。

不完全デバッグ環境のような比較的現実的なデバッグ環境の中で、ソフトウェア開発管理者は、ソフトウェア信頼度の達成度合と限られたコストの両面を考慮して、効率的に高信頼性を有するソフトウェアシステムを開発するための戦略を立てることが必要となってくる [2]。多くの不完全デバッグモデルに含まれる完全デバッグ率は、テスト工数の投入量増減、テストおよびその熟練者の人員配置、ソフトウェア開発管理ツールの使用などのテスト戦略により制御できるものと考えられる。したがって、ソフトウェア開発管理者の視点から考えると、最適な完全デバッグ率およびソフトウェアのリリース時刻を推定することは、テスト戦略を立案する上で興味ある問題の 1 つと考えられる。Xie and Yang [2] は、完全デバッグ率がソフトウェアコストに与える影響について議論を行い、不完全デバッグ環境下におけるソフトウェアの最適リリース政策を導出している。

本研究では、まず、不完全デバッグ環境が影響を与えるソフトウェアコストパラメータを再抽出して、Xie and Yang が提案したソフトウェアコストと完全デバッグ率との関係性を再定式化する。この関係性に基づいて、コスト評価基準の下で最適な完全デバッグ率とソフトウェアの最適リリース時刻を導出するための最適政策について議論する。

## 2 ソフトウェアの最適リリース問題

ソフトウェアの最適リリース問題を取り扱う場合、まず、総期待ソフトウェアコストの定式化にあたり、次のようなコストパラメータが定義される。

- $c_1$  = テスト工程において発見されるフォールト 1 個当りの修正コスト。
- $c_2$  = 運用段階において発見されるフォールト 1 個当りの修正コスト ( $0 < c_1 < c_2$ )。
- $c_3$  = テストに要する単位時間当りのテストコスト ( $c_3 > 0$ )。

これらのコストパラメータを用いて、ソフトウェアの最適リリース問題における目的関数である総期待ソフトウェアコストが定式化できる。代表的な SRGM の 1 つである非同次ポアソン過程 (nonhomogeneous Poisson process: NHPP) モデルを適用すると、テスト工程における総テスト時間を  $T$  としたときの総期待ソフトウェアコスト  $C(T)$  は、次式のように定式化される。

$$C(T) = c_1 H(T) + c_2 [H(\infty) - H(T)] + c_3 T. \quad (1)$$

ここで、 $H(T)$  は NHPP の平均値関数であり、総テスト時間  $T$  までに発見される総期待フォールト数を示す。この場合、コスト評価基準に基づくソフトウェアの最適リリース時刻は、式 (1) を最小にするテスト終了時刻である。

## 3 デバッグ環境とコストパラメータとの関係性

多くの不完全デバッグモデルに含まれる完全デバッグ率  $p$  は、工数 (費用) を投じることによるテスト戦略を実施することで、完全デバッグ率を制御できるものと考えられる。したがって、完全デバッグ率を上げるためのテスト戦略を積極的に実行すると、テスト期間の短縮化、運用段階における保守コストの減少が期待されるが、テスト戦略の実施に伴う多大なテストコストが必要となる。一方、完全デバッグ率を上げるためのテスト戦略を積極的に実施しない場合は、テストコストを抑制することができるが、不効率なテスト実施によるテスト期間の長期化や運用段階における保守コストの増大を招くことが考えられる。したがって、不完全デバッグ環境において、ソフトウェアの最適リリース時刻と最適完全デバッグ率を同時に推定する手法の必要性が存在することが確認できる。

Xie and Yang は、これまで定数として取り扱っていたコストパラメータの中で、 $c_3$  のみについて、完全デバッグ率に依存した関数となるように拡張して、総期待ソフトウェアコストを定式化する手法を提案している。本研究では、完全デバッグ率が影響を与えるコストパラメータとして、 $c_1$  および  $c_3$  を取り上げて、それらを以下のように拡張することによって、不完全デバッグ環境下における総期待ソフトウェアコストを再定式化する。

$$c_1 \rightarrow \frac{c_1}{1-p}, \quad c_3 \rightarrow \frac{c_3}{1-p}. \quad (2)$$

式 (2) のように拡張されたコストパラメータは、完全デバッグ率  $p$  に関して (単調) 増加関数である。

## 4 不完全デバッグ環境下におけるソフトウェアの最適リリース政策

本研究では、式 (2) のように拡張されたコストパラメータを適用すると、不完全デバッグ環境を仮定したテスト工程およびソフトウェアの運用段階において費や

される総期待ソフトウェアコスト  $C(T, p)$  は、式 (3) より、次式のように与えられる。

$$C(T, p) = \frac{c_1}{1-p} H_I(T) + c_2 [H_I(\infty) - H_I(T)] + \frac{c_3}{1-p} T. \quad (3)$$

式 (3) における添え字  $I$  は、式 (3) の  $H(t)$  が不完全デバッグモデルであることを示す。コスト評価基準の下での最適完全デバッグ率およびソフトウェアの最適リリース時刻を導出する問題は、以下に示すような総期待ソフトウェアコストに関する最小化問題として帰着される。

$$\left. \begin{array}{l} \text{minimize } C(T, p) \\ \text{subject to } 0 < p < 1, T > 0 \end{array} \right\}. \quad (4)$$

本研究では、最適政策を導出するにあたり、比較的単純な構造をもつ不完全デバッグモデルである、指数形不完全デバッグモデル [3]、

$$H_I(t) = \frac{a}{p} (1 - e^{-pbt}) \quad (0 < p < 1), \quad (5)$$

を適用する。ここで、 $a$  はテスト開始前にソフトウェア内に潜在する総期待フォールト数、 $b$  は 1 個当りのフォールト発見率を表す。

式 (4) を解析すると、最適完全デバッグ率  $p^*$  およびソフトウェアの最適リリース時刻  $T^*$  を導き出すための最適政策は、次のように導出される。

[最適政策]

- (1)  $K \leq 0$  のとき、 $0 < p < 1 - (c_1/c_2)$  の範囲において  $h(p)$  は減少関数となる。したがって、唯一の解として最適完全デバッグ率  $p^*$  が存在する。このとき、最適リリース時刻  $T^*$  は、

$$T^* = \frac{1}{pb} \ln \left[ \frac{ab\{c_2(1-p) - c_1\}}{c_3} \right], \quad (6)$$

である。

- (2)  $K > 0$  のとき、 $h(p)$  は、 $dh(p)/dp = 0$  となる点  $p'$  ( $0 < p' < 1 - (c_1/c_2)$ ) を境界に増加関数から減少関数となり、

1.  $h(p') > 0$  のとき、最適完全デバッグ率  $p^*$  に関して 2 つの解が得られ、 $C(T^*, p^*) = \min[C(T_1, p_1), C(T_2, p_2)]$  で与えられる。
2.  $h(p') = 0$  のとき、唯一の最適完全デバッグ率  $p^*$  が存在して、最適リリース時刻  $T^*$  は、 $p^*$  を式 (6) に代入することにより求められる。
3.  $h(p') < 0$  のとき、最適完全デバッグ率  $p^*$  および最適リリース時刻  $T^*$  は存在しない。

上の最適政策において、

$$\begin{aligned} h(p) = & abc_1(2p-1)\{c_2(1-p) - c_1\} \\ & - c_1c_3(2p-1) - c_2(1-p)^2 \\ & + c_3[(1-p)(c_1 - c_2) + p\{c_2(1-p) - c_1\}] \\ & \cdot \ln \left[ \frac{ab\{c_2(1-p) - c_1\}}{c_3} \right], \end{aligned} \quad (7)$$

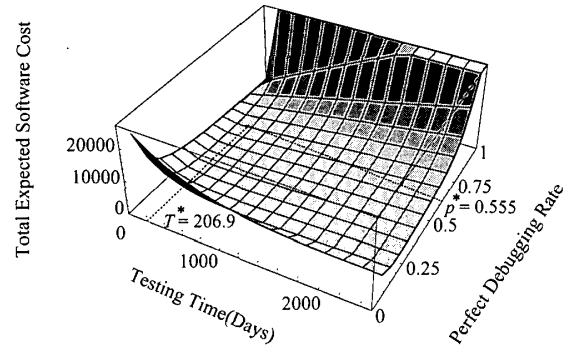


図 1: 総期待ソフトウェアコスト,  $\hat{C}(T, p)$ .

であり、

$$K = (c_2 - c_1) \left[ abc_1 + c_3 \ln \left\{ \frac{ab(c_2 - c_1)}{c_3} \right\} \right] - c_1c_3 + c_2, \quad (8)$$

である。

## 5 適用例

最初に、実際のテスト工程において観測された実測データを適用して、不完全デバッグモデルである式 (5) に含まれるパラメータ  $a$  および  $b$  に対する推定値を得る必要がある。本研究では、文献 [3] において推定されたパラメータ推定値  $\hat{a} = 10.19$  および  $\hat{b} = 0.019$  を適用することにする。また、本研究では、コストパラメータを  $c_1 = 20$ ,  $c_2 = 150$ ,  $c_3 = 1$  と設定する。

図 1 に総期待ソフトウェアコスト  $C(T, p)$  の挙動を示す。このとき、 $K = 824.7 (> 0)$  である。また、 $p' = 0.683$  より、 $h(p') = h(0.6833) = 34.35 (> 0)$  であることから、この場合、[最適政策] の (2) 1. が適用されることがわかる。そこで、

$$\begin{aligned} C(T^*, p^*) = & \min[C(T_1, p_1), C(T_2, p_2)] \\ = & \min[C(206.8, 0.554), C(15.62, 0.823)] \\ = & \min[1501, 1846] \\ = & 1501 \end{aligned} \quad (9)$$

より、最適完全デバッグ率および最適リリース時刻は、それぞれ、 $p^* = 0.554$  および  $T^* = 206.8$  と推定される。

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) および笹川科学研究助成 (研究番号 16-064) の援助を受けたことを付記する。

参考文献

- [1] J.D. Musa, D. Iannio, and K. Okumoto, *Software Reliability: Measurement, Prediction, Application*. McGraw-Hill, New York, 1987.
- [2] M. Xie and B. Yang, "A study of the effect of imperfect debugging on software development cost," *IEEE Trans. Software Engineering*, Vol. 29, No. 5, pp. 471-473, 2003.
- [3] 山田 茂, 山根 勉, 尾崎 俊治, "エラー修正率を考慮したソフトウェア信頼度成長モデル," 情報処理学会論文誌, Vol. 27, No. 1, pp. 64-71, 1989.