

ソフトウェア開発プロジェクトにおける最適保証期間の決定

林坂弘一郎 (01800035), 土肥正 (01307065)
広島大学大学院工学研究科情報工学専攻

1. はじめに

ソフトウェアテスト工程で発見されたフォールト数に関するデータを分析し、ソフトウェアの信頼性を定量的に評価するとともに、運用段階に移行するための最適な時期を決定することは必要不可欠である。このような問題は最適リリース問題 [1] と呼ばれ、開発プロジェクトの管理者にとってプロジェクトの成否を左右する重要な問題である。ただし、実際のテスト工程ではソフトウェア内に潜在する全てのフォールトを発見・除去することは困難であるため、開発管理者は一定の保証期間を設定し、リリース後に発生するであろうソフトウェア障害の原因究明を行い、フォールトの発見・除去を行う必要がある。保証期間中に保全を実施するために、ソフトウェア開発管理者はプロジェクトチームの維持を継続しなければならない反面、運用段階における管理費用を削減し、人的資源を有効に活用することが要求される。その意味において、ソフトウェア保証期間を決定する問題はリリース時期を決定することと同様に重要であるが、これまであまり考えられることはなかった。

山田 [2], Kimura *et al.* [3] は、ソフトウェア保証期間が確率変数である場合を想定し、最適リリース時期を決定する問題を考えている。Dohi *et al.* [4] はテスト工程におけるデバッグ過程が非同次ポアソン過程 (NHPP) によって記述されるという仮定の下で、総期待ソフトウェア費用を最小にする最適保証期間を決定する問題を定式化している。しかしながら、文献 [4] ではテスト段階と運用段階におけるデバッグ環境の違いを全く考慮しておらず、非常に単純な費用構造の下で議論を行っているため、必ずしも現実的なモデル化を行っているとは言い難い。実際、最近のソフトウェア利用技術の進展により、ユーザの運用環境が開発段階で想定した環境と一致するとは限らない状況が頻繁に見受けられる。本稿では、文献 [5] で提案された運用段階における信頼性評価モデルと同様な方法で、テスト段階と運用段階におけるソフトウェア実行環境の違いを明確にモデル化し、総期待ソフトウェア費用を最小にする最適なソフトウェア保証期間を決定する。

2. モデルの記述

ソフトウェアフォールトの発見過程を記述するために、次の仮定を設定する。

(仮定 A) ソフトウェアのテスト中に発見されたフォールトは瞬間的に発見・除去される。

(仮定 B) プログラム中に含まれる初期フォールト数 N_0 は平均 $\omega (> 0)$ のポアソン分布に従う。

(仮定 C) ソフトウェアフォールトが発見されるまでの時間は独立で同一に分布する非負の確率変数列であり、その確率分

布関数は $F(t)$ 、確率密度関数は $f(t)$ 、 $t \geq 0$ によって与えられる。

以上の仮定より、時刻 t までに発見されるフォールト数 $\{N(t), t \geq 0\}$ の確率関数 (p.m.f.) は

$$\Pr\{N(t) = m\} = \frac{[\omega F(t)]^m e^{-\omega F(t)}}{m!} \quad m = 0, 1, 2, \dots \quad (1)$$

と表現される。これより、確率過程 $\{N(t), t \geq 0\}$ は平均値関数 $\omega F(t)$ を持つ NHPP となり、単位時間当たりのフォールト発見率 (デバッグ率) は次式によって与えられる。

$$r(t) = \frac{f(t)}{1 - F(t)}. \quad (2)$$

ソフトウェアの開発工程において $t_0 (> 0)$ 時間のテストを行った後、当該製品をユーザもしくは市場にリリースする。ソフトウェア製品のライフサイクル $t_L (> 0)$ は既知であり、 t_0 に比べて十分大きいものとする。開発管理者は、リリース直後から $t_w (0 \leq t_w \leq t_L)$ までの期間 (すなわち、時刻 $t_0 + t_w$ まで) デバッグ保証を行い、保証期間中に発生したソフトウェア障害に対する保守費用を負担する。他方、保証期間が終了した後に障害が発生した場合には、ライフサイクル経過後に予定されるバージョンアップによる保守を行うため、障害の原因となるフォールトの修正・除去は行わないものとする。このとき、開発管理者側にはソフトウェア障害の発生に対してペナルティ費用が発生するものと仮定する。テスト期間中に発生するフォールト 1 個当たりのデバッグ費用を $c_0 (> 0)$ 、保証期間中に発生するフォールト 1 個当たりのデバッグ費用を $c_w (> 0)$ とし、保証期間後に発生するフォールト 1 個当たりのペナルティ費用を $c_L (> 0)$ で表すこととする。また、単位時間当たりテスト費用を $k_0 (> 0)$ 、単位時間当たり保証費用を $k_w (> 0)$ とする。

これより、問題はテスト以降の開発管理における総期待ソフトウェア費用を最小にする最適なソフトウェア保証期間 t_w^* を求めることである。

3. 総期待ソフトウェア費用の定式化

ソフトウェアの運用環境とテスト段階におけるデバッグ環境は異なるものがあるのが一般的であり、この相違点はハードウェアにおける加速寿命試験環境と通常操作環境の違いに類似している。本稿でも文献 [5] と同様に、リリース後の運用環境の相対的な厳しさを表すパラメータ (環境係数) を $\alpha (> 0)$ を導入し、ソフトウェアの運用プロファイルをモデル化する。ここで、 $\alpha = 1$ の場合は運用環境とテスト環境はほぼ同様であることを意味し、 $\alpha > 1$ ($\alpha < 1$) は運用環境がテスト環境よりも厳しい (緩い) ことを示している。このとき、保証期

間中 $(t_0, t_0 + t_W]$ に発見・除去されるソフトウェアバグ数の p.m.f. は次式で与えられる。

$$\Pr\{N(t_0 + t_W) - N(t_0) = m\} = \frac{\{\omega [F(t_0 + at_W) - F(t_0)]\}^m}{m!} e^{-\omega [F(t_0 + at_W) - F(t_0)]} \quad (3)$$

保証期間後からライフサイクル終了までの $(t_0 + t_W, t_0 + t_L]$ において発生したソフトウェア障害に対してはフォールの発見・除去が行われないので、ソフトウェアの信頼度成長現象は保証期間終了時点 $t_0 + t_W$ までで停止すると考えても差し支えない。保証期間終了時点でのデバッグ率は $r(t_0 + at_W)$ であるので、その後デバッグ率の変化が観測されないものと仮定すれば、保証期間後のソフトウェア障害発生時の振る舞いは次式によって表現される。

$$\Pr\{N(t_0 + t_L) - N(t_0 + t_W) = m\} = \frac{\{\omega [e^{-r(t_0 + at_W)(t_0 + at_W)} - e^{-r(t_0 + at_W)(t_0 + at_L)}]\}^m}{m!} \times e^{-\omega [e^{-r(t_0 + at_W)(t_0 + at_W)} - e^{-r(t_0 + at_W)(t_0 + at_L)}]} \quad (4)$$

以上のことから総期待ソフトウェア費用 $C(t_W)$ は次式となる。

$$\begin{aligned} C(t_W) &= k_0 t_0 + c_0 \omega F(t_0) + k_W t_W \\ &\quad + c_W \omega [F(t_0 + at_W) - F(t_0)] \\ &\quad + c_L \omega [e^{-r(t_0 + at_W)(t_0 + at_W)} - e^{-r(t_0 + at_W)(t_0 + at_L)}] \end{aligned} \quad (5)$$

4. 最適保証期間の決定

各ソフトウェアフォールトが発見されるまでの時間がパラメータ $\lambda (> 0)$ の指数分布に従うものと仮定する [1]。次の仮定を設定する。

(A-I) $c_L > c_W > c_0$

以上の仮定の下で、最適ソフトウェア保証期間に関する次の定理を得る。

定理 1: ソフトウェアの障害発生時間分布がパラメータ $\lambda (> 0)$ の指数分布に従う場合、仮定 (A-I) の下で総期待ソフトウェア費用を最小にする最適ソフトウェア保証期間は以下のように与えられる。

(1) $k_W \geq (c_L - c_W)\omega\lambda a e^{-\lambda t_0}$ のとき、 $t_W^* = 0$ となり、

$$C(t_W^*) = k_0 t_0 + c_0 \omega (1 - e^{-\lambda t_0}) + c_W \omega [e^{-\lambda t_0} - e^{-\lambda(t_0 + at_L)}] \quad (6)$$

(2) $k_W < (c_L - c_W)\omega\lambda a e^{-\lambda t_0}$ かつ $k_W > (c_L - c_W)\omega\lambda a \times e^{-\lambda(t_0 + at_L)}$ のとき、総期待ソフトウェア費用 $C(t_W)$ を最小にする唯一の t_W^* ($0 < t_W^* < t_L$) が存在し、

$$\begin{aligned} C(t_W^*) &= k_0 t_0 + c_0 \omega (1 - e^{-\lambda t_0}) + k_W t_W^* \\ &\quad + c_W \omega [e^{-\lambda t_0} - e^{-\lambda(t_0 + at_W^*)}] \\ &\quad + c_W \omega [e^{-\lambda(t_0 + at_W^*)} - e^{-\lambda(t_0 + at_L)}] \end{aligned} \quad (7)$$

(3) $k_W \leq (c_L - c_W)\omega\lambda a e^{-\lambda(t_0 + at_L)}$ のとき、 $t_W^* = t_L$ となり、

$$C(t_W^*) = k_0 t_0 + c_0 \omega (1 - e^{-\lambda t_0}) + k_W t_L + c_W \omega [e^{-\lambda t_0} - e^{-\lambda(t_0 + at_L)}] \quad (8)$$

表 1: 環境係数 a に関する感度分析結果。

a	Exponential		Gamma	
	t_W^*	$C(t_W^*)$	t_W^*	$C(t_W^*)$
0.50	669.3	136.1	141.3	83.4
0.75	475.6	133.9	102.2	82.9
1.00	372.3	132.7	80.9	82.7
1.25	307.6	131.9	67.4	82.5
1.50	262.9	131.4	57.9	82.4
2.00	205.0	130.7	45.6	82.2
3.00	144.0	130.0	32.4	82.1

5. 数値例

ここでは実際の開発工程で観測されたソフトウェアフォールト発見時刻データ (データ数 86) を用い、単一のフォールト発見時刻が指数分布及び 2 次ガンマ分布に従うとした場合の数値例を示す。70 個のフォールトデータが得られた時点 ($t_0 = 67.374$) で製品を市場にリリースし、最尤法に基づいてモデルパラメータの推定が可能であるとする。モデルパラメータの推定値は、指数分布の場合 $(\hat{\omega}, \hat{\lambda}) = (98.5188, 1.84e-02)$ 、2 次ガンマ分布に対して $(\hat{\omega}, \hat{\lambda}) = (75.1746, 6.46224e-02)$ となった。その他のパラメータは $k_0 = 0.02$ 、 $k_W = 0.01$ 、 $c_0 = 1.0$ 、 $c_W = 2$ 、 $c_L = 20$ 、 $t_L = 1000$ とした。表 1 は、2 種類のモデルにおいて環境係数 a が最適保証期間に与える影響を調べたものである。表 1 の結果より、 a が大きくなる、すなわち運用環境がテスト環境に比べてより厳しくなるにつれて、最適保証期間 t_W^* は短くなり、総期待ソフトウェア費用も小さくなるのがわかる。

参考文献

- [1] K. Okumoto and L. Goel, "Optimum release time for software systems based on reliability and cost criteria," *J. Sys. Software*, 1 315-318 (1980).
- [2] 山田 茂, "ソフトウェア保守コストモデルに基づく保証期間を考慮した最適リリース問題," 情処学論, 35 (10), 2197-2202 (1994).
- [3] M. Kimura, T. Toyota, and S. Yamada, "Economic analysis of software release problems with warranty cost and reliability requirement," *Reliab. Eng. & Sys. Safe.*, 66 49-55 (1999).
- [4] T. Dohi, H. Okamura, N. Kaio and S. Osaki, "The age-dependent optimal warranty policy and its application to software maintenance contract," *Proc. 5th Int'l Conf. on Probab. Safe. Assess. and Mgmt.* (S. Kondo and K. Furuta, eds.), 4 2547-2552, University Academy Press Inc. (2000).
- [5] 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法 - 加速寿命試験モデルの提案 -, " 信学論 (A), J830 A 294-301 (2000).