基于多版本较准的软件费用模型研究

苏喜红 吴智博 杨孝宗 刘宏伟
(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

摘 要 软件可靠性模型为软件的费用模型提供了很好的依据。在软件可靠性增长模型的测试阶段和操作运行阶段,环境的不同导致了两个阶段故障检测率的不同。在多版本较准的软件可靠性模型基础上,构造了一个综合了软件设计费用、软件测试费用、软件维护费用、软件失效造成的风险损失的软件费用模型。最后从软件费用出发,讨论了软件的最佳发布时间。

关键词 多版本较准,软件可靠性模型,软件费用模型,软件最佳发布时间
中图法分类号 TP302.8 **文献标识码** A

Study on Software Cost Model Based on Multi-version Calibration

SU Xi-hong WU Zhi-bo YANG Xiao-zong LIU Hong-wei
(Department of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

Abstract Software reliability models provide a good base for software cost modeling. The testing and operation environments may be essentially different, thus the fault detection rate of testing is different from that of the operation phase. A cost model based on the multi-version calibration method and combining with the software design cost, the testing cost, the maintenance and risk cost due to software failures was developed. In order to minimize the total expected software cost, the optimal release policies were discussed.

Keywords Multi-version calibration, Software reliability model, Software cost model, Optimum software release time

软件测试是提高软件质量的主要手段,测试的时间越长,软件的质量越高,软件的可靠性就越高,测试的成本也越高。然而用户要求的交付期限是既定的,测试不可能无限制地进行下去,必须在交付期限之前结束。何时结束软件测试、进行软件发布是一个综合考虑软件可靠性、相关软件成本以及合同交付期的问题[1]。在过去的 20 年中,已经有许多软件成本模型从可靠性和成本的角度提出软件的最优发布策略。其中,绝大多数的成本模型都是通过利用软件可靠性增长模型(SRGMs)来建立软件开发的成本模型,通过计算成本模型的最小值来确定软件的最优发布策略[2]。

Huang 等[3]建立的软件成本模型考虑引进新的检测工具。Kuo 等[4]建立的软件成本模型考虑测试努力,并且测试启动费用为时间相关函数。Xie 等[5]使用测试级别指标来描述实际的测试情况,并使用 SRGMs 中的故障排除效率 p 作为测试级别。舒燕君等[6]建立了考虑测试级别的软件成本模型。这些成本模型中,都假设软件测试阶段和运行阶段或维护阶段的均值函数相同。赵靖等[7]指出了软件测试和运行阶段的故障检测率不同,但是很少有软件费用模型会考虑到这一点,所以不能够准确地预测软件的最佳发布时间以及当调整这些参数时对软件费用的影响。

本文首先分析杨彬等[8]建立的基于多版本较准的软件可靠性评估,这里多版本指的是同一软件各个不同版本。然后,构造了一个基于多版本较准的软件费用模型,其软件测试与操作运行阶段的均值函数不同。这里所说的费用不仅包括软件的研发和测试费用,还包括软件的维护费用和软件失效可能给用户带来的风险损失。从软件费用的角度出发,讨论了软件的最佳发布时间,并通过实例证明了该费用模型更符合软件测试和操作运行的实际情况。

1 基于多版本较准的 GO 模型

GO(Goel-Okumoto)模型于 1979 年由 Goel 和 Okumoto 提出,是关于连续时间的 NHPP (Non-Homogeneous Poisson Process) 模型中的经典模型。在很多具体的应用中,GO 模型都工作得很好[9]。

GO 模型的基本假设如下:

(1) 软件在与预期的操作环境相似的条件下运行;

(2) 在任何时间序列 $t_0 < t_1 < \cdots < t_n$ 构成的时间区间 $(t_0, t_1), (t_1, t_2), \cdots, (t_{n-1}, t_n)$ 中检测到的故障数是相互独立的;

(3) 每个故障的严重性和被检测到的可能性大致相同;

(4) 在 t 时刻检测出的累积故障数 $[N(t), t \geq 0]$ 是一个独立增量过程,$N(t)$ 服从期望函数为 $m(t)$ 的泊松分布,在 $(t, t+\Delta t)$ 时间区间中发现的故障数的期望值正比于 t 时刻剩余

到稿日期:2008-07-20 本文受国家自然科学基金项目(60503015)资助。
苏喜红(1981-),女,博士研究生,主要研究方向为软件可靠性评估等,E-mail:suxh@ftcl.hit.edu.cn;吴智博(1954-),男,博士生导师,主要研究方向为体系结构、容错计算技术等;杨孝宗(1939-),男,博士生导师,主要研究方向为可穿戴计算技术、容错计算技术、软件可靠性技术等;刘宏伟(1971-),男,副教授,主要研究方向为软件测试、软件可靠性评估、容错计算、可穿戴计算。

故障的期望值;

(5)累积故障数的期望函数 $m(t)$ 是一个有界单调增函数,并满足

$$m(0) = 0 \quad (1)$$

$$\lim_{t\to\infty} m(t) = a$$

其中,a 是最终可能被检测出的故障总数的期望值。根据前面的假设,函数 $m(t)$ 可以用下面的公式计算:

$$m(t) = a(1 - e^{bt}) \quad (2)$$

b 是剩余故障发现率,或者说 b 是在时刻 t 每个故障的查出率[10]。

因此,软件可靠度函数为:

$$R(x/t) = e^{-(m(t+x)-m(t))} = \exp\{-e^{-bt}\cdot m(x)\} \quad (3)$$

若给定软件可靠度,由式(3)可推出软件交付时间[11,12]为

$$t = \frac{1}{b}\left\{\ln m(x) - \left(\ln\ln\frac{1}{R}\right)\right\} \quad (4)$$

软件可靠性增长模型能够如实地反映运行阶段失效率对于维护软件可靠性与软件费用的平衡,决定何时发布软件是非常重要的。绝大多数软件可靠性增长模型都假设软件测试与操作运行环境是相似的,实际上,软件的性能依赖于它的执行环境,软件的执行环境包括操作系统、硬件平台以及操作剖面。软件的测试剖面很难真实地反映运行剖面,因而在运行阶段和测试阶段的软件的故障检测率是不同的[7]。但很少有软件费用模型会考虑到这一点,所以不能够精确地估计软件费用。

由于软件的测试环境与运行环境可能差别很大,因此假设(1)与实际的情况不相符,需要对假设条件作进一步的改进。在测试操作剖面与运行操作剖面一致的情况下,实际运行可以当作是测试的延续,两者服从相近的随机分布;当测试操作剖面与运行操作剖面相差很大时,用剖面差异性因子[8]来刻画这种差异。对于多版本的软件而言,软件各版本测试操作剖面与实际运行剖面之间的差异性是相近的,称之为剖面差异性因子。本文将软件测试失效数据和实际使用过程中的失效数据用 G-O 模型来描述。

剖面差异性因子 C_a 和 C_b 的计算,定义统计量:

$$\Lambda = -2\log(L_{H_0}(\text{碲},\text{碘}) / L(\hat{a},\hat{b},\hat{c},\hat{d}))$$

其中,碲,碘把测试阶段的失效数据和运行阶段的失效数据结合起来形成新的失效数据集,将此新的失效数据应用于 G-O 模型,得出参数 a 和 b 的估计值。L_{H_0}(碲,碘)为将软件测试阶段失效数据和运行阶段失效数据合起来的新数据的 a,b 的极大似然函数。$L(\hat{a},\hat{b},\hat{c},\hat{d})$ 为 a,b,c,d 的极大似然函数。a,c 分别为测试和实际运行开始前,软件中的初始故障数;b,d 分别表示测试和实际运行中每个故障平均引发的失效数;碲为测试结束时软件中残存的故障数。该统计量服从自由度为 2 的 χ^2 分布,给定显著性水平 α,则可以确定是否接收假设 H_0。显著性水平的取值,可以根据实际情况来定,通常取值为 0.1 或者 0.05。如果接收假设 H_0,则认为 $C_a = 1$, $C_b = 1$,意味着测试操作剖面和运行操作剖面差异不大;否则认为两者相差很大。

$$C_a = c/\text{碲} \quad C_b = d/b$$

根据软件测试和操作运行阶段失效数据[13],得出测试剖面的参数估计量为 $\hat{a} = 237.63$, $\hat{b} = 1.43\times10^{-3}$,运行剖面的参数估计量为 $\hat{c} = 22.45$, $\hat{d} = 6.07\times10^{-5}$,统计量 $\Lambda = 424.37$。给定的显著水平 $a = 0.10$,拒绝 H_0,认为测试剖面与运行剖面存在很大差异,求得

$$C_a = 2.52,\ C_b = 23.56$$

对于同一个软件的不同版本,各个版本的测试剖面与运行剖面之间的差异性是相近的。因此,通过软件当前版本的测试失效数据和以前版本的剖面差异性因子,可估计出当前版本的运行阶段的可靠性。因此软件测试阶段均值函数为:

$$m_1(t) = a(1 - e^{-bt}) \quad (5)$$

运行阶段的均值函数为:

$$m_2(t) = \text{碲}C_a(1 - e^{-btC_b}) \quad (6)$$

2 基于多版本较准的软件费用模型

2.1 注释

T 为软件测试时间;

T^* 为软件最佳发布时间;

C_0 为软件设计与开发阶段的费用和测试启动费用;

C_1 为单位时间软件测试费用;

C_2 为维护阶段单位时间排除一个错误的费用;

C_3 为单位软件故障导致软件失效给用户造成的损失;

$E(T)$ 为到时间 T 软件系统费用的期望值;

μ_w 为维护期间排除一个错误时间的期望值;

T_W 为软件维护期。

2.2 软件费用模型

软件费用包含以下几部分:

(1)软件的设计与开发费用、测试的启动费用。这些费用在测试的开始阶段就已经确定,因此,可认为是一个常数 C_0[14]。

(2)软件的测试费用 $E_1(T)$,假设测试费用与测试时间成正比,因为在开始测试阶段,虽然故障出现的密度比较高,但确定故障的位置、排除故障需要花费较长的时间。所以有

$$E_1(T) = C_1 T \quad (7)$$

(3)维护期的排错费用 $E_2(T)$。在维护阶段排除故障的费用与在时间间隔 $[T, T+T_W]$ 内排除所有检测到的故障所需的时间成正比。排错费用的期望值为

$$E_2(T) = C_2\mu_w m_2(T_W) \quad (8)$$

(4)由于软件失效产生的风险费用 $E_3(T)$。对风险费用的估算与软件的应用目标相关,可能很大也可能很小,而且通常我们无法预期软件的使用时限。从最坏的情况考虑,可以认为软件中潜伏的任何软件失效,都会给用户造成损失,并且假设每个软件失效给用户带来的损失都是相同的。因此,假定风险费用与软件发布后软件中剩余的故障数成正比。故有

$$E_3(T) = C_3[-m_1(T)] \quad (9)$$

于是,软件费用的期望值可以表示为

$$E(T) = C_0 + C_1 T + C_2\mu_w m_2(T_W) + C_3[a - m_1(T)] \quad (10)$$

3 软件的最佳发布时间

3.1 最佳发布时间讨论

将式(5)-式(9)代入式(10)中,得出软件系统的预期费用,并在此基础上讨论软件的最佳发布时间。

$$E(T) = C_0 + C_1 T + C_2 C_a\mu_w ae^{-bT}(1 - e^{-bC_bT_w}) +$$

$$aC_3 e^{-bT} \tag{11}$$

$$y(T) = \frac{dE(T)}{dT} = C_1 - abe^{-bT}(C_2\mu_w C_a - C_2\mu_w C_a e^{-bC_b T_w} + C_3) = C_1 - abu(t) \tag{12}$$

$$k(T) = \frac{d^2 E(T)}{d^2 T} = b^2 ae^{-bT}(C_2\mu_w C_a - C_2\mu_w C_a e^{-bC_b T_w} + C_3) = b^2 au(T) \tag{13}$$

其中，$u(T) = e^{-bT}(C_2\mu_w C_a(1 - e^{-bC_b T_w}) + C_3)$ (14)

容易看出，$u(T)$ 是一个关于时间 T 的减函数。

定理 1 如果给出了 C_0 C_1 C_2 C_3 μ_w，T_w 则可以描述软件的预期总费用最小值 T^* 如下：

1) 如果 $u(0) \leq 0$

a) 如果 $y(0) \leq 0$，则 $T^* = \infty$；

b) 如果 $y(\infty) > 0$，则 $T^* = 0$；

c) 如果 $y(0) > 0$，$y(\infty) < 0$，则存在 T'，在区间 $(0, T')$，$y(T) > 0$；在区间 (T', ∞)，$y(T) < 0$，其中 $T' = y^{-1}(0)$。因此，当 $E(0) \leq E(\infty)$，则 $T^* = 0$；当 $E(0) > E(\infty)$，$T^* = \infty$。

2) 如果 $u(\infty) > 0$

a) 如果 $y(0) \geq 0$，则 $T^* = 0$；

b) 如果 $y(\infty) \leq 0$，则 $T^* = \infty$；

c) 如果 $y(0) < 0$，$y(\infty) > 0$，则存在 T''，使 T 在 $(0, T'')$，$y(T) < 0$；在区间 (T'', ∞)，$y(T) \geq 0$，其中 $T'' = y^{-1}(0)$，则 $T^* = y^{-1}(0)$。

3) 如果 $u(0) > 0$，$u(\infty) < 0$，则存在 T^0，当 T 在 $(0, T^0]$，$u(T) > 0$；当 T 在 (T^0, ∞)，$u(T) < 0$，其中，$T^0 = u^{-1}(0)$。

当 T 在 $(0, T^0]$ 区间，有：

a) 如果 $y(0) \geq 0$，则 $T^* = 0$；

b) 如果 $y(T^0) \leq 0$，则 $T^* = T^0$；

c) 如果 $y(0) < 0$，则 $T^* = y^{-1}(0)$；

当 T 在 (T^0, ∞) 区间，有：

a) 如果 $y(T^0) \leq 0$，则 $T^* = \infty$；

b) 如果 $y(\infty) > 0$，则 $T^* = T^0$；

c) 如果 $y(T^0) > 0$ 且 $y(\infty) < 0$，则有当 $E(T^0) \leq E(\infty)$，则 $T^* = T^0$；当 $E(0) > E(\infty)$，$T^* = \infty$。

证明：根据高等数学中有关定理易证。

3.2 实例说明

为了验证推荐的软件费用模型，应用数据[13]进行分析。

应用最小二乘法估算 CGOM[15] 模型中的参数，得

$\hat{a} = 234.0326$，$\hat{b_1} = 48.2630$，

$\hat{b_2} = 0.0013$，$\hat{N} = 11.6626$

失效累积数的期望函数为：

$$m(t) = 234.0326(1 - e^{-0.0013t}) + 11.6626(e^{-0.0013t} - e^{-48.2630t})$$

费用模型中的参数根据经验数据进行估算，我们取值如下[10]（单位：人员-单元）

$C_0 = 50$，$C_1 = 700$，$C_2 = 3600$，$C_3 = 50000$，$\mu_w = 0.5$，$T_w = 1500$。

可求得 $T^* = y^{-1}(0) = 2352$（天）

用极大似然估计估算出推荐模型中的参数，得

$\hat{a} = 237.63$，$\hat{b} = 1.43 \times 10^{-3}$，

$\hat{c} = 22.46$，$\hat{d} = 6.07 \times 10^{-3}$，故有

测试阶段和运行阶段的失效累积数的期望函数分别为：

$m_1(t) = 237.63 \times (1 - e^{-0.00143 \times t})$

$m_2(t) = 22.45 \times (1 - e^{-0.0000607 \times t})$

将上面式子带入式(12)和式(14)，易得

$u(0) > 0$，$y(0) < 0$。

则根据前面的定理得，$T^* = y^{-1}(0) = 2291$（天）

两个软件费用模型的测试时间和软件测试费用关系如图 1 所示，其中软件维护期取为 1500 天。表 1 列出两个模型软件运行阶段数据的拟合情况。

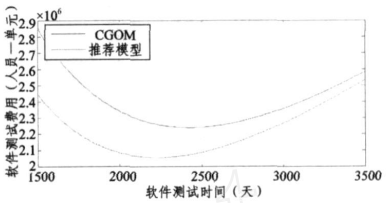

图 1 软件测试时间和软件测试费用的关系

表 1 两个模型软件运行阶段数据拟合情况对比

比较标准	推荐模型	CGOM 模型
SSE	13.2175	26306.5271

其中，$SSE = \sum_{i=1}^{n}(y_i - m(t_i))^2$，$n$ 表示软件运行阶段失效数据集中失效样本的数量，$m(t_i)$ 表示到 t_i 时刻为止故障累计数的估计值。y_i 表示到 t_i 时刻为止故障累计数的实测值。SSE 的值越小，曲线拟合得越好。

软件测试的截止时间为 1001 天[13]，不同的维护期，两个模型估计的软件费用期望值的比较结果如表 2 所列。其中，$E1(T)$ 和 $E2(T)$ 分别为到时间 T 为止，CGOM 模型与推荐模型估计软件费用的期望值。

表 2 两个模型软件费用估计情况对比

(T) \ w	1500	2500	3500
E1(T)	3820403	3831679	3834752
E2(T)	3543551	3545724	3547769
(E1(T) - E2(T))/ E2(T)	0.0781	0.0806	0.0808

推荐模型的参数 C_2 和 C_3 变化软件费用和软件最佳发布时间的影响如表 3 - 表 5 所列，其中 $T_W = 1500$ 天，$T^* = 2291$，$E(T^*) = 3543551$。

表 3 参数 C_2 改变时的软件费用和最佳发布时间的变化

参数 C_2 的改变	最佳发布时间 T	软件费用 E(T)	$\left\|\frac{T - T^*}{T^*}\right\|$	$\left\|\frac{E(T) - E(T^*)}{E(T^*)}\right\|$
$C_2 - C_2 * 30\%$	2273	3542497	0.7856%	0.02974%
$C_2 - C_2 * 20\%$	2279	3542848	0.5237%	0.01983%
$C_2 - C_2 * 10\%$	2285	3543200	0.2618%	0.00990%
$C_2 + C_2 * 10\%$	2297	3543904	0.2618%	0.00996%
$C_2 + C_2 * 20\%$	2303	3544255	0.5237%	0.01986%
$C_2 + C_2 * 30\%$	2308	3544607	0.7420%	0.02980%

表 4 参数 C_3 改变时的软件费用和最佳发布时间的变化

参数 C_3 的改变	最佳发布时间 T	软件费用 E(T)	$\left\|\frac{T - T^*}{T^*}\right\|$	$\left\|\frac{E(T) - E(T^*)}{E(T^*)}\right\|$
$C_3 - C_3 * 30\%$	2066	2691766	9.8210%	24.0373%
$C_3 - C_3 * 20\%$	2149	2975695	6.1982%	16.0250%
$C_3 - C_3 * 10\%$	2224	3259623	2.9245%	8.0125%

$C_3 + C_3$ *10 %	2352	3827480	2.6626 %	8.0126 %
$C_3 + C_3$ *20 %	2409	4111409	5.1506 %	16.0251 %
$C_3 + C_3$ *30 %	2461	4395337	7.4203 %	24.0376 %

表 5　参数 C_2、C_3 改变时的软件费用和最佳发布时间的变化

参数 C_3 和 C_2 的改变	最佳发布时间 T	软件费用(T)	$\left\|\frac{T - T^*}{T^*}\right\|$	$\left\|\frac{E(T) - E(T^*)}{E(T^*)}\right\|$
C_2 - C_2 *30 %,C_3 - C_3 *30 %	2042	2690711	10.8686 %	24.0674 %
C_2 - C_2 *20 %,C_3 - C_3 *20 %	2135	2974991	6.8093 %	16.0449 %
C_2 - C_2 *10 %,C_3 - C_3 *10 %	2217	3259271	3.2300 %	8.0225 %
C_2 + C_2 *10 %,C_3 + C_3 *10 %	2358	3827832	2.9245 %	8.0225 %
C_2 + C_2 *20 %,C_3 + C_3 *20 %	2419	4112112	5.5871 %	16.0450 %
C_2 + C_2 *30 %,C_3 + C_3 *30 %	2474	4396392	7.9878 %	24.0674 %

从实例分析,我们可得到下面结论:

1)一个好的软件可靠性模型不但能够预测发布后的软件在运行阶段的可靠性,而且为软件的成本模型提供了很好的依据。推荐的软件费用模型考虑了运行和测试阶段软件的故障检测率不同,如表 1 所列,推荐模型在软件运行阶段失效数据拟合效果优于 CGOM 模型。

2) CGOM 模型没有考虑软件测试与运行环境差别,如表 2 中不同软件维护期,CGOM 模型的软件费用估计值大于推荐模型。并且 CGOM 模型估算出的软件最佳发布时间比推荐模型多 61 天。

3)随着维护时间增长,两个模型软件费用估计差值与推荐模型软件费用估计值的比值呈递增趋势。在软件运行阶段,剩余的故障被检测到的概率会变得越来越小,因此,推荐模型的软件费用估计值相对越来越小,更接近于真实的软件费用估计的期望值。

4)如表 3 - 表 5 所列,参数 C_2 和 C_3 同时改变,每增加(减少) 10 %,软件费用变化率约为 8 %。参数 C_2 和 C_3 每同时减少 10 %,软件最佳发布时间平均变化率约为 3.623 %;参数 C_2 和 C_3 每同时增加 10 %,软件最佳发布时间平均变化率约为 2.663 %。参数 C_2 每增加(减少) 10 %,软件最佳发布时间变化率约为 0.26 %,软件费用变化率约为 0.0099 %;参数 C_3 每增加(减少) 10 %,软件最佳发布时间变化率约为 3 %,软件费用变化率约为 8 %,所以 C_3 是推荐模型中的敏感参数。

结束语　本文提出了一个基于多版本较准的软件费用模型,并与 CGOM 模型进行了比较,结果表明推荐模型的软件费用估计更接近于真实的软件费用;以基于多版本较准的模型为基础建立了一个综合考虑了软件生命周期各个阶段费用的软件费用模型,讨论了基于本模型的软件最佳发布时间;利用一组实测数据估算了一个软件的最佳发布时间,并分析了推荐模型中的敏感参数。

软件测试人员利用这个软件费用模型,决定是继续进行软件测试还是发布软件,如果不能发布,则可以预测软件的发布时间,并为用户预测软件费用,使用户能够以合理的花费拥有软件。这个模型为估算软件费用和软件最佳发布时间作了有益的探索。

参 考 文 献

[1] Lyu M R. Handbook of software reliability engineering [M]. New York: McGraw-Hill and IEEE computer society, 1996

[2] Pham H. Software reliability and cost models: Perspectives, comparison, and practices [J]. European Journal of operational research, 2003, 149(3): 475-489

[3] Chin Y H, Jung H L, Sy Y K, et al. Software reliability modeling and cost estimation incorporating testing-effort and efficiency [C] // Proceedings of the 10th IEEE International Symposium on Software Reliability Engineering (ISSRE'99). Boca Raton, FL, USA, 1999: 62-72

[4] Chin Y H, Sy Y K, Michael R L. Optimal software release policy based on cost and reliability with testing efficiency [C] // Proceedings of the 23rd IEEE Annual International Computer Software and Applications Conference (COMPSAC'99). Phoenix, Arizona, USA, 1999: 468-473

[5] Xie M, Yang B. A study of the effect of imperfect debugging on software development cost [J]. IEEE Transaction on software engineering, 2003, 29(5): 471-473

[6] 舒燕君, 吴智博, 刘宏伟, 等. 基于测试级别的软件成本模型的研究[J]. 北京化工大学学报, 2007, 34(1): 59-63

[7] 赵靖, 刘宏伟, 崔刚, 等. 考虑测试与运行差别的软件可靠性增长模型[J]. 计算机研究与发展, 2006, 43(3): 503-508

[8] 杨彬, 陈丽蓉. 基于多版本较准的软件可靠性评估[J]. 计算机工程与设计, 2007, 28(19): 4597-4599

[9] Pham H, Zhang X M. A software cost model with warranty and risk costs [J]. IEEE Transactions on computer, 1999, 48(1): 71-75

[10] Malaiya Y K, Denton. What do the software reliability growth model parameters represent [C] // Proceedings of the eighth international symposium on software reliability engineering (ISSRE'97). Albuquerque, 1997: 124-135

[11] Xie M, Hong G Y. A study of the sensitivity of software release time [J]. The journal of systems and software, 1998, 44(2): 163-168

[12] Kimura M, Toyota T, Yamada S. Economic analysis of software release problems with warranty cost and reliability requirement [J]. Reliability engineering and system safety, 1999, 66(1): 49-55

[13] Daniel R J, Zhang X M. Adjusting software failure rates that are estimated from test data [J]. IEEE Transaction on reliability, 2005, 54(1): 107-114

[14] Chin Y H, Michael R L. Optimal release time for software systems considering cost, testing-effort, and test efficiency [J]. IEEE transactions on reliability, 2005, 54(4): 583-591

[15] 刘宏伟, 杨孝宗, 曲峰, 等. 基于 CGOM 的软件费用模型研究[J]. 计算机学报, 2003, 26(10): 1332-1336