ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.21, No.5, May 2010, pp.942−949
doi: 10.3724/SP.J.1001.2010.03539

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

考虑不完美排错情况的 NHPP 类软件可靠性增长模型*

谢景燕+, 安金霞, 朱纪洪

(清华大学 计算机科学与技术系,北京 100084)

NHPP Software Reliability Growth Model Considering Imperfect Debugging

XIE Jing-Yan+, AN Jin-Xia, ZHU Ji-Hong

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

+ Corresponding author: E-mail: xiejingyan@gmail.com, http://www.tsinghua.edu.cn

Xie JY, An JX, Zhu JH. NHPP software reliability growth model considering imperfect debugging. *Journal of Software*, 2010,21(5):942−949. http://www.jos.org.cn/1000-9825/3539.htm

Abstract: Addressing at the problem that the considerations of imperfect debugging phenomenon in the existing software reliability growth model are limited, this paper proposes a software reliability growth model which takes into account of the imperfect debugging comprehensively. Both the fault introduction and the fault removal efficiency are considered in this model, and a time-dependent fault removal efficiency function is introduced. The goodness-of-fit and the predictive power of the new model are examined by using a public software failure data set. The results show that compared with other existing models, the proposed model fits the failure data better and can predict this set of data more accurately.

Key words: software reliability growth model (SRGM); nonhomogeneous Poisson process (NHPP); imperfect debugging; fault removal efficiency; fault introduction rate

摘 要: 针对现有 NHPP 类软件可靠性增长模型对故障排错过程中不完美排错情况考虑不完全的现状,提出了一种新的软件可靠性增长模型.该模型全面考虑了不完美排错的两种情况:既考虑了排错过程中引入新错误的可能性,又考虑了不完全排错的情况,并且引入了一种故障排除率随时间变化的故障排除率函数,使模型更符合实际情况.利用公开发表的两组不同的软件失效数据对该模型进行验证的结果表明,与现有的对不完美排错情况考虑不完全的模型相比,该模型能够取得更好的拟合结果和预测效果.

关键词: 软件可靠性增长模型;非齐次泊松过程;不完美排错;故障排错率;错误引入率

中图法分类号: TP311 **文献标识码**: A

故障发现过程和故障排错过程是影响软件可靠性建模准确性的两个主要因素.现有的模型主要集中于对故障发现过程的精确建模,对故障排错过程的研究并不充分.对软件排错过程更精确的描述可以显著提高软件可靠性模型的评价能力和预测能力[1,2],而准确地评价和预测软件可靠度有助于确定软件的合理释放时间,节省测试费用和测试时间,对软件工程开发和管理工作有指导作用.非齐次泊松过程(non-homogeneous Poisson process,简称 NHPP)类软件可靠性增长模型是软件可靠性增长模型中非常重要的一类,也是目前研究较多、应

* Supported by the Science Foundation of Aeronautics of China under Grant No.20095558008 (航空科学基金)
Received 2008-07-04; Revised 2008-10-09; Accepted 2008-11-28

用较广的一类模型[3].Goel 和 Okumoto 于 1979 年提出的 G-O 模型[4]是最早提出的一种 NHPP 类可靠性模型,它假设排错过程是完美的,即排错是完全的,且排错过程不引入新的错误.近年来,国内外一些学者对不完美排错情况进行了研究.Pham 等人于 1999 年提出了 PNZ_SRGM 模型[5],其考虑了不完美排错过程中引入新错误的情况,提出了错误引入率的概念,但其假设故障排除效率为 100%,并没有考虑不完美排错情况中不完全排错的情况.Zhang 等人于 2003 年提出了一种考虑错误引入和故障排除效率的模型[6].这个模型中考虑到了不完全排错的情况,但其假设故障排除效率为一个固定的常数,并没有考虑其随时间的变化.李春芝等人提出了综合考虑测试覆盖率和故障排除率的模型[7],考虑了故障排除率不为 100%的情形,引入了一个 p 为常数的故障排除率,但没有研究排错过程中可能引入新错误的情况.刘宏伟等人对故障排错过程进行了研究,提出了一种 NHPP 类可靠性增长模型框架[8],但没有给出模型的具体形式,无法对模型进行验证并直接应用于工程实际.

事实上,不完美排错情况包括两个方面的内容:首先,对故障排除过程的研究表明,从发现故障到故障完全被排除有一定的时间延迟[9−12],故障排除效率并非总是 100%.而且实验结果表明[13],故障排除效率在单元测试阶段为 15%~50%,集成测试阶段为 25%~40%,系统测试阶段为 25%~55%.因此,将故障排除效率看作时间的函数更为合理;其次,软件的排错过程是一个复杂的人为过程,假设排错的过程不会引入新的错误是不现实的.因此,应该把软件故障总数看作随时间变化的函数.

本文综合分析了不完美排错的两种情形,引入了一种故障排除效率随时间变化的故障排除率函数,提出了一种既考虑排错过程中引入新错误的可能性,又考虑不完全排错情况的可靠性增长模型,提高了此类可靠性增长模型的评估和预测能力.利用公开发表的两组不同的软件失效数据对这个模型进行了曲线拟合,拟合结果表明,本文提出的模型在该组失效数据上能够取得更好的拟合结果.

1 不完美排错情况下的软件可靠性建模

符号说明:

$N(t)$:在$[0,t]$时间段内发现的累计软件故障计数.

$m(t)$:$[0,t]$时间段内累计故障数的期望值,$m(t)=E[N(t)]$.

$x(t)$:到时刻 t 为止,被检测到且被正确改正的软件错误计数.

$a(t)$:软件故障总数函数,等于 $x(t)$与到时刻 t 为止,软件中尚未被检测到的故障数之和.

a_0:测试开始时软件系统中存在的故障总数.

b:故障检测率,表示软件中每个错误被检测到的概率.

$p(t)$:故障排除率函数,表示在时刻 t,每个被检测到的错误被正确改正的概率.

$R(x|t)$:软件可靠度函数,表示从时刻 t 开始到 $t+x$ 时间段内,软件的可靠性.

1.1 考虑不完美排错情况的软件可靠性增长模型框架

排错过程的不完美性主要体现在两个方面:

(a) 排错过程可能引入新的错误,即软件故障总数函数 $a(t)$是随时间变化的;

(b) 排错是不完全的,即在时刻 t,每个被检测到的错误以概率 $p(t)$被正确改正,故障排除率函数 $p(t)$是随时间变化的.

为了构建描述不完美排错情况的可靠性模型框架,需对经典 G-O 模型的假设[14]进行修改和补充,形成了以下的假设条件:

(1) 到时间 t 的累计错误数 $N(t)$服从均值函数为 $m(t)$的泊松过程.任意时间间隔 t 到 $t+\Delta t$ 内期望的错误发生数与 t 时刻剩余的错误数成比例(保留的假设).

(2) 排错过程不完美,排错过程可能引入新的错误,软件错误总数是随时间变化的(改进的假设).

(3) 排错过程不完美,排错不完全,错误排除率是时间的函数(改进的假设).

(4) 软件运行剖面与可靠性测试剖面相同(保留的假设).

(5) 软件中每个错误是相互独立的,每个错误导致系统发生失效的可能性也相同(保留的假设).

由假设(1),有:

$$m(t+\Delta t)=b(a(t)-x(t))\Delta t+o(\Delta t) \tag{1}$$

从而可以得到微分方程:

$$\frac{\mathrm{d}m(t)}{\mathrm{d}t}=b(a(t)-x(t)) \tag{2}$$

由假设(3),有:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t}=p(t)\frac{\mathrm{d}m(t)}{\mathrm{d}t} \tag{3}$$

由方程(2)、方程(3)可得:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t}=b\cdot p(t)\cdot(a(t)-x(t)) \tag{4}$$

方程(2)、方程(3)的初始条件为

$$m(0)=0 \tag{5}$$

$$x(0)=0 \tag{6}$$

由公式(4)、公式(6)可得:

$$x(t)=b\cdot\exp\left(-b\cdot\int_0^t p(\tau)\mathrm{d}\tau\right)\cdot\int_0^t a(\tau)\cdot p(\tau)\cdot\exp\left(b\cdot\int_0^\tau p(u)\mathrm{d}u\right)\mathrm{d}\tau \tag{7}$$

从而由公式(2)、公式(5)可解得模型的累计故障数均值函数为

$$m(t)=b\cdot\int_0^t\left(a(\tau)-b\cdot\exp\left(-b\cdot\int_0^\tau p(u)\mathrm{d}u\right)\cdot\int_0^\tau a(u)\cdot p(u)\cdot\exp\left(b\cdot\int_0^u p(x)\mathrm{d}x\right)\mathrm{d}u\right)\mathrm{d}\tau \tag{8}$$

由于到时刻 t 为止的累计故障计数 $N(t)$服从均值 $m(t)$的非齐次泊松分布,所以,

$$P\{N(t)=n\}=\frac{(m(t))^n}{n!}\exp(-m(t)),\ n=0,1,2,... \tag{9}$$

根据非齐次泊松分布的性质,可靠度函数为

$$R(x\mid t)=P\{N(t+x)-N(t)=0\}=\exp[-(m(t+x)-m(t))] \tag{10}$$

1.2 一种具体的考虑不完美排错情况的软件可靠性增长模型

本节在第 1.1 节的基础上,结合不完美排错过程的特点,提出了一种具体的考虑了不完美排错情况的软件可靠性增长模型.

假设故障总数函数是随时间线性变化的[5],则故障总数函数可以用如下的函数来描述:

$$a(t)=a_0\cdot(1+\alpha\cdot t) \tag{11}$$

其中,$\alpha>0$,α的大小决定了排错过程中引入新错误概率的大小.

随着测试的进行,较容易被发现和排除的错误逐渐被改正,较难改正的错误所占比例越来越大,故软件测试的故障排除效率会逐渐降低.基于以上的分析,故障排除率函数 $p(t)$应该满足如下条件:

$p(t)\in[0,1]$且 $p(t)$为递减函数.当 $t\to\infty$时,$p(t)\to 0$.

因此,可以选取如下的函数来定义故障排除率函数:

$$p(t)=\frac{1}{1+kt} \tag{12}$$

其中 $k>0$,k 的大小决定了故障排除率变化的快慢.

将公式(11)、公式(12)代入公式(7)、公式(8),易得:

$$x(t)=a_0(1+\alpha t)-a_0(1+kt)^{-\frac{b}{k}}-\alpha\cdot\frac{(1+kt)-(1+kt)^{-\frac{b}{k}}}{b+k} \tag{13}$$

$$m(t)=b\cdot\left[\frac{a_0(1+kt)}{k-b}+\frac{\alpha}{b+k}\left(\frac{(1+kt)^2}{2k}-\frac{(1+kt)^{1-\frac{b}{k}}-1}{k-b}\right)\right] \tag{14}$$

得到累计故障计数的均值函数 $m(t)$后,就可以用最小二乘法估计法得到 $m(t)$中的参数.

2 模型验证与评价

2.1 模型评价标准

应用最小二乘法估计软件可靠性模型的参数,用于评价模型拟合能力的标准主要有误差平方和(sum of squared errors,简称 SSE)与回归曲线方程相关指数(R-square)两个评价标准.

SSE 用来描述累计错误数的实际值与预测值之间的距离,其定义为

$$SSE=\sum_{i=1}^{n}(y_i-\hat{m}(t_i))^2,$$

其中:n 表示失效数据集中失效样本的数量;y_i 表示到时刻 t_i 观测到的错误数; $\hat{m}(t_i)$ 表示到时刻 t_i,模型估算得到的错误数.SSE 的值越小,说明曲线拟合效果越好.

R-square 定义为

$$R\text{-}square=\frac{\sum_{i=1}^{n}(\hat{m}(t_i)-\overline{y})^2}{\sum_{i=1}^{n}(y_i-\overline{y})^2},$$

其中, $\overline{y}$ 表示观测到错误数的平均值, $\overline{y}=\frac{1}{n}\sum_{i=1}^{n}y_i$ .R-square 的值越接近 1,说明曲线拟合得越好.

2.2 对比模型的选择

为使实验分析更具有对比性,本文选择了没有考虑不完美排错,或者考虑不完美排错不完全的 4 个现有软件可靠性增长模型作为对比模型,与本文提出的模型做了对比分析.

模型 I:Goel 和 Okumoto 于 1979 年在文献[4]中提出的 G-O 模型,其故障计数均值函数为 $m(t)=a(1-\exp(-bt))$,其中,a 为软件初始故障总数,b 为故障检测率.G-O 模型假设错误检测到后被立即排除,且排错过程不引入新的错误.

模型 II:李春芝等人在文献[7]中提出的模型,其故障计数均值函数为 $m(t)=\frac{a}{p}\left[1-\left(\frac{1+bt}{\mathrm{e}^{bt}}\right)^{\frac{1}{p}}\right]$,其中,$a$ 为软件初始故障总数,b 为故障检测率,p 为故障排除率.此模型考虑了故障排除率不为 100%的情形,引入了一个 p 为常数的故障排除率;另一方面,此模型没有考虑不完美排错过程中可能引入新错误的情况.

模型 III:Pham 等人于 1999 年在文献[5]中提出的 PNZ_SRGM 模型,其故障计数均值函数为

$$m(t)=\frac{a}{1+\beta\cdot\exp(-b(1+\beta)t)}\cdot\left[(1-\exp(-b(1+\beta)t))\cdot\left(1-\frac{\alpha}{b(1+\beta)}\right)+\alpha\cdot t\right],$$

其中,a 为软件初始故障总数,α决定了引入新错误概率的大小,b 和β为体现学习过程的故障检测率函数中的两个参数.PNZ-SRGM 考虑了排错过程中引入新错误的可能性,即故障总数函数 $a(t)$是测试时间的函数;另一方面,PNZ-SRGM 没有考虑不完全排错的情况,认为排错率函数为 100%.

模型 IV:Zhang 等人于 2003 年在文献[6]中提出的模型,其故障计数均值函数为

$$m(t)=\frac{a}{p-\beta}\left[1-\left(\frac{(1+\alpha)\mathrm{e}^{-bt}}{1+\alpha\mathrm{e}^{-bt}}\right)^{\frac{c(p-\beta)}{b}}\right],$$

其中,a 为软件初始故障总数,p 为故障排除率,β为新错误引入概率,α,b 和 c 为体现学习过程的故障检测率函数中的 3 个参数.此模型考虑了不完美排错过程中排错过程可能引入新错误的情形,即把故障总数函数看作是测试时间的函数;另一方面,此模型还考虑了不完全排错的情况,引入了故障排除率 p,但其并没有把故障排除率看作随时间变化的函数.

这 4 个模型分别考虑了不完美排错的不同情形,从而使第 2.3 节中的模型对比分析更具针对性.

2.3 模型验证和分析

为了验证模型的有效性,同时验证模型评估和预测能力的稳定性,本文用两组经典的用于模型验证的失效数据集对模型进行了验证,并对实验结果进行了分析和解释.

2.3.1 在某实时控制系统失效数据上的实验

本节使用文献[6]中给出的一组实时控制系统的失效数据,对本文提出的模型及第 2.2 节中的 4 个对比模型进行了曲线拟合,并比较了这些模型的拟合结果.Zhang 和 Teng[6]也采用这组失效数据进行了模型验证工作.为了方便比较,本文和文献[6]一样,也采用前 122 个失效数据进行了曲线拟合、参数估计,采用第 123 个~第 136 个失效数据检验模型的预测效果.表 1 给出了模型拟合结果的 SSE 参数标准和 R-square 参数标准.图 1~图 5 给出了各模型对失效数据的拟合结果以及置信度为 95%的置信区间.

Table 1 Comparison results of different models

表 1 各模型拟合结果比较

Model	SSE (fitted)	SSE (predicted)	R-square
Model I	2 479	5 009.3	0.983 6
Model II	10 750	9 694.1	0.928 94
Model III	7 713	13 968	0.949 0
Model IV	8 910	15 830	0.915
Model V (the proposed model)	515	715.76	0.996 6

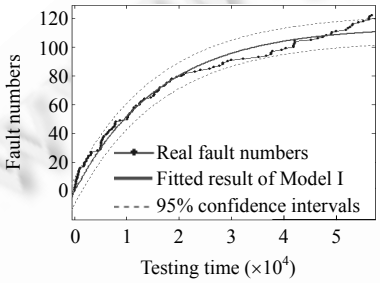

Fig.1 Fitted result and confidence intervals (95%) of Model I

图 1 模型 I 拟合结果与置信区间(95%)

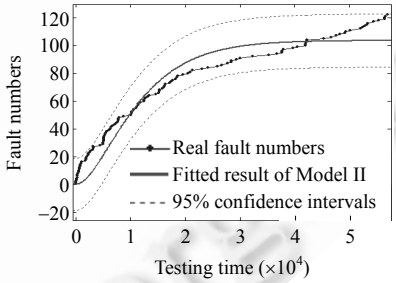

Fig.2 Fitted result and confidence intervals (95%) of Model II

图 2 模型 II 拟合结果与置信区间(95%)

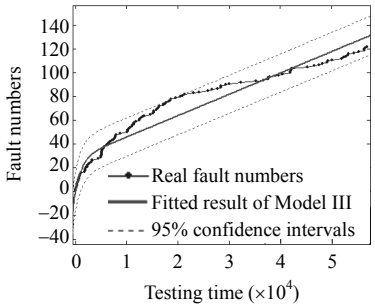

Fig.3 Fitted result and confidence intervals (95%) of Model III

图 3 模型 III 拟合结果与置信区间(95%)

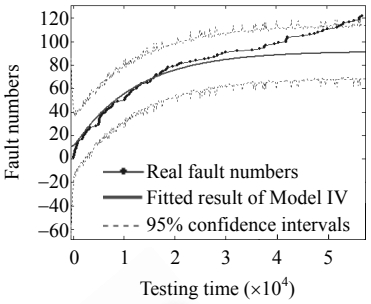

Fig.4 Fitted result and confidence intervals (95%) of Model IV

图 4 模型 IV 拟合结果与置信区间(95%)

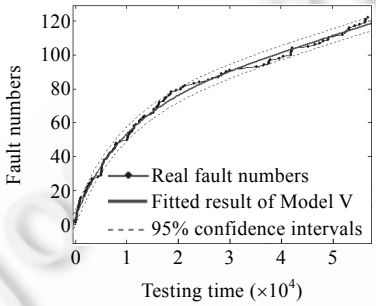

Fig.5 Fitted result and confidence intervals (95%) of Model V

图 5 模型 V 拟合结果与置信区间(95%)

从表 1 中的 SSE(拟合)、SSE(预测)和 R-square 参数上可以看出,本文提出的模型是 5 个模型中最优的.值得注意的一个现象是,除了本文提出的模型外,在其余 4 个模型中,模型 I 对此组失效数据的拟合和预测结果最好,但它反而是对不完美排错情况最欠考虑的一个模型.造成这一现象的原因是由于文献[5]中提到的由参数个数导致的模型复杂性问题,当模型中未知参数过多时,若用于进行参数估计的失效数据偏少,则有可能导致模型参数估计值不准确,从而导致拟合和预测结果发生偏差.即当失效数据偏少时,模型中若包含过多的参数则可能会导致模型评估和预测能力的下降.但这并不能说明模型 II、模型 III 和模型 IV 对不完美排错情况的考虑没有意义,因为在实际软件工程开发实践中,用于进行可靠性评估的失效数据会远大于本实验中的 122 个.此外,这一现象也恰恰表明,本文提出的模型在参数个数和模型评估预测能力上相对于其他几个模型做到了较好的平衡,即在增加模型参数个数的同时,并没有导致模型评估和预测能力的降低.

从图 1~图 5 可以直观地看出,与现有的 4 个模型相比,本文提出的模型能够更精确地对失效数据进行拟合,且置信区间明显较小,即本文提出的模型对可靠度的评估能力比其他 4 个模型要好.

2.3.2 在 Tandem 计算机公司某软件产品失效数据上的实验

本节使用的失效数据集来自文献[15]提供的 Tandem 计算机公司的失效数据集.文献[5,6,16]都采用了此组数据进行模型分析.文献[15]对现有的经典 NHPP 类模型进行比较分析发现,在此组失效数据集上,G-O 模型的拟合预测能力最好.因此,本节在此组失效数据集上用 G-O 模型与本文提出的模型进行了比较.实验结果见表 2,其中,失效数据集的前 9 个数据用于对模型参数进行最小二乘估计,后 11 个数据用于检测模型的预测能力.从表 2 中数据可以看出,本文提出的模型对错误数的预计值比 G-O 模型的预计值更接近实际错误数,预测结果更好.

Table 2 Comparison result of G-O model and the model proposed

表 2 本文提出模型与 G-O 模型的比较

Test period (week)	Accumulated testing time (hour)	Real fault numbers	Fault numbers predicted by G-O model	Fault numbers predicted by the proposed model
1	519	16	—	—
2	968	24	—	—
3	1 430	27	—	—
4	1 893	33	—	—
5	2 490	41	—	—
6	3 058	49	—	—
7	3 625	54	—	—
8	4 422	58	—	—
9	5 218	69	—	—
10	5 823	75	98	73.7
11	6 539	81	107	80.8
12	7 083	86	116	86.3
13	7 487	90	123	90.4
14	7 846	93	129	94.2
15	8 205	96	129	97.9
16	8 564	98	134	101.8
17	8 923	99	139	105.6
18	9 282	100	138	109.6
19	9 641	100	135	113.6
20	10 000	100	133	117.6

3 结 论

本文对目前最常用的 NHPP 类软件可靠性增长模型的假设进行了改进,全面分析了不完美排错过程的两个重要方面:排错过程引入新错和不完全排错的情况,提出了一种故障总数函数和故障排除率函数均随时间变化的软件可靠性增长模型,更为准确地体现了软件不完美排错过程的特点,使得模型更符合实际测试情况.本文使用两组公开发表的软件失效数据集对模型进行了实验验证分析,结果表明,本文提出的模型在这两组失效数据集上的拟合效果都比较好,能够比较准确地评估和预测软件可靠性.

致谢 在此,衷心感谢清华大学计算机科学与技术系林闯教授对本文工作提供的指导性意见和帮助.

References:

[1] Lo JH. Effect of the delay time in fixing a fault on software error models. In: Proc. of the 31st Annual Int'l Computer Software and Applications Conf., Vol.2. Washington: IEEE Computer Society Press, 2007. 711−716.

[2] Shibata K, Rinsaka K, Dohi T, Okamura H. Quantifying software maintainability based on a fault-detection/correction model. In: Proc. of the 13th Pacific Rim Int'l Symp. on Dependable Computing. Washington: IEEE Computer Society Press, 2007. 35−42.

[3] Huang CY, Lyu MR, Kuo SY. A unified scheme of some nonhomogenous Poisson process models for software reliability estimation. IEEE Trans. on Software Engineering, 2003,29(3):261−269. [doi: 10.1109/TSE.2003.1183936]

[4] Goel AL, Okumoto K. Time-Dependent error-detection rate model for software and other performance measures. IEEE Trans. on Reliability, 1979,28:206−211. [doi: 10.1109/TR.1979.5220566]

[5] Pham H, Nordmann L, Zhang XM. A general imperfect-software-debugging model with S-shaped fault-detection rate. IEEE Trans. on Reliability, 1999,48(2):169−175. [doi: 10.1109/24.784276]

[6] Zhang XM, Teng XL, Pham H. Considering fault removal efficiency in software reliability assessment. IEEE Trans. on Systems, Man, and Cybernetics: Systems and Humans, 2003,33(1):114−119.

[7] Li CZ, Yue XG, Liu HW. A NHPP software reliability growth model incorporating test coverage and fault removal efficiency. Electronics Quality, 2005,(3):34−36 (in Chinese with English abstract).

[8] Liu HW, Yang XZ, Yue XG, Qu F. A framework for NHPP software reliability growth models. Computer Engineering & Science, 2005,27(4):1,2,18 (in Chinese with English abstract).

[9] Lo JH. Considering both failure detection and fault correction activities in software reliability modeling. In: Proc. of the IEEE Region 10 Conf. Hong Kong: IEEE/IET Electronic Library, 2006. 1−4.

[10] Wu YP, Hu QP, Xie M, Ng SH. Modeling and analysis of software fault detection and correction process by considering time dependency. IEEE Trans. on Reliability, 2007,56(4):629−642. [doi: 10.1109/TR.2007.909760]

[11] Huang CY, Lin CT. Software reliability analysis by considering fault dependency and debugging time lag. IEEE Trans. on Reliability, 2006,55(3):436−450. [doi: 10.1109/TR.2006.879607]

[12] Huang CY, Lin CT, Kuo SY, Lyu MR, Sue CC. Software reliability growth models incorporating fault dependency with various debugging time lags. In: Proc. of the 28th Annual Int'l Computer Software and Applications Conf. (COMPSAC 2004). Washington: IEEE Computer Society Press, 2004. 186−191.

[13] Jones C. Software defect-removal efficiency. IEEE Computer, 1996,29(4):73−74.

[14] Lyu MR, ed.; Liu XC, Zhong WY, *et al*., Trans. Handbook of Software Reliability Engineering. Beijing: Publishing House of Electronics Industry, 1997 (in Chinese).

[15] Wood A. Predicting software reliability. IEEE Computer, 1996,29(11):69−77.

[16] Pham H, Zhang XM. NHPP software reliability and cost models with testing coverage. European Journal of Operational Research, 2003,145(3):443−454. [doi: 10.1016/S0377-2217(02)00181-9]

附中文参考文献:

[7] 李春芝,岳晓光,刘宏伟.考虑测试覆盖率和故障排除效率的软件可靠性增长模型.电子质量,2005,(3):34−36.

[8] 刘宏伟,杨孝宗,岳晓光,曲峰.一个 NHPP 类软件可靠性增长模型框架.计算机工程与科学,2005,27(4):1,2,18.

[15] Lyu MR,编;刘喜成,锺婉懿,等,译.软件可靠性工程手册.北京:电子工业出版社,1997.

谢景燕(1983－),女,北京人,硕士,主要研究领域为软件可靠性.

安金霞(1973－),女,博士生,高级工程师,主要研究领域为软件可靠性,容错计算.

朱纪洪(1968－),男,博士,教授,博士生导师,主要研究领域为飞行控制,容错计算.