

# 考虑测试效率的软件可靠性模型研究

谈维新, 沈元隆

(南京邮电大学 电子科学与工程学院, 江苏 南京 210003)

**摘要:**对软件可靠性模型的研究,主要集中在如何使所建立的模型尽可能地贴近实际情况。目的是提高软件可靠性模型的拟合能力,使可靠性模型能更好地对软件将来的失效行为进行预测。经典非其次泊松过程模型的假设条件都忽略了不完全排错以及排错时引入错误的可能性,也没有考虑到在测试过程中资源投入等问题。笔者认为在测试过程中,投入的资源、测试环境以及测试人员的努力程度对测试的结果有一定的影响。同时,考虑到软件排错过程中新错误的引入和错误的不完全排除等情况,提出了一种考虑测试效率的可靠性模型。通过与已有模型进行仿真试验对比显示,新模型具有更好的拟合性能,具有很好的工程实践意义。

**关键词:**可靠性模型;非齐次泊松过程;不完全排错;测试效率

**中图分类号:**TP31

**文献标识码:**A

**文章编号:**1673-629X(2011)08-0067-04

## Research of Software Reliability Model with Test Efficiency

TAN Wei-xin, SHEN Yuan-long

(College of Electronic Science and Engineering, Nanjing University of Posts and Telecommunications,  
Nanjing 210003, China)

**Abstract:** The research of software reliability models focuses on how to make reliability models accord with actual situation. The aim of it is improving the ability of fitting accuracy for software reliability models and predicting the software's fault behavior in the future better. The assumptions of classical non-homogeneous Poisson process models disregard imperfect debugging and the introduction of new errors. In addition, the consumption of resources is not considered either. Think the consumption of resources in test process, test environment, and the input level of tester can affect the results of test. Meanwhile, perfect debugging and imperfect debugging in the test process are also considered, and a new NHPP model with respect to test efficiency is proposed. After contrasting with some released models, the new model shows better performance to fit data and theoretically can be applied into projects.

**Key words:** reliability model; non-homogeneous Poisson process; imperfect debugging; test efficiency

## 0 引言

随着现代信息技术的发展,计算机应用已经渗透到了国防建设和国民经济的各个部门,甚至人类活动的各个领域。软件是计算机系统的神经中枢,在计算机系统有着至关重要的作用,几乎每天都有大量的软件产品投入使用,同时,软件质量的问题也日益突现出来<sup>[1,2]</sup>。软件可靠性作为衡量软件质量的重要特性,其定量评估和预测是人们所关注和研究的重点。而作为定量评测软件可靠性的软件可靠性模型则是研究的核心和关键。

在众多的软件可靠性模型中,目前还没有能广泛

适用的软件可靠性模型,原因在于假设条件是软件可靠性模型建立的主要依据。对于同一软件来说,不同的假设条件可能会对应不同的可靠性模型。可靠性模型的适用与否,与所取的假设条件是否贴近实际情况有直接的关系。因此,对软件可靠性模型的研究,主要集中在如何使所建立模型的假设条件尽可能地贴近实际情况。文中将以软件可靠性预测作为研究工作的重点。

## 1 考虑测试效率的软件可靠性模型

### 1.1 模型改进的出发点

非齐次泊松过程(Non-Homogeneous Poisson Process, 简称为 NHPP)类软件可靠性增长模型是应用最广、影响最大的一类模型<sup>[3,4]</sup>。迄今为止,这类模型依然是拟合效果最好、结构和应用最简单的模型,已经成为软件可靠性工程实践活动中使用最多的工具。

NHPP 模型均具有如下的形式:

收稿日期:2011-01-04;修回日期:2011-04-15

基金项目:南京邮电大学校科研基金(01 院 32)

作者简介:谈维新(1986-),男,江苏盱眙人,硕士研究生,研究方向为通信系统的可靠性技术;沈元隆,教授,研究方向为通信系统的可靠性技术。

$$\frac{dm(t)}{dt} = b(t)[a(t) - m(t)] \quad (1)$$

式中,  $m(t)$  为均值函数;

$a(t)$  表示  $t$  时刻为止软件中的错误总量;

$b(t)$  表示  $t$  时刻的错误发现率。

该方程的通解是

$$m(t) = e^{-B(t)} \left[ m_0 + \int_{t_0}^t a(\tau) b(\tau) e^{B(\tau)} d\tau \right] \quad (2)$$

$$\text{式中, } B(t) = \int_{t_0}^t b(\tau) d\tau$$

边界条件  $m_0 = m(t_0)$ ,  $t_0$  为测试开始时间。

可以看出,一般的 NHPP 可靠性增长模型都是基于这样的假设:

(1) 软件失效的发生符合非齐次泊松过程;

(2) 某一时刻,软件的失效密度与此时刻软件中的剩余错误数成正比;

(3) 软件错误之间彼此独立,一旦发现就立即排除且排错时间忽略不计。

除此之外,很多非其次泊松过程模型还有其他一些重要的假设:例如,错误发现率是一个常数;排除错误过程不引入新的错误等。但这在实际过程中是不太现实的。

文献[3]中提出了一种基于测试覆盖的可靠性模型,考虑到了排错的概率以及排错时可能引入新错误等问题。文中在此基础上对假设条件进行修改和完善,认为引入错误的错误数应该与被排除的错误数成正比,而不是与错误总数成正比。另外,错误发现率不是常数,有人认为这应该是一个递减的函数,即针对相同的测试用例,错误发现率是应该越来越小的。

另外,文中还认为在软件测试过程中,测试资源的投入(包括人力、物力等)、测试环境、测试人员的努力投入程度等等因素都会对测试结果产生一定的影响,测试的效率不可能达到百分百,而是一个随着时间不断变化的过程。文考虑用一个函数的形式来描述这个测试效率。综上,文中提出一种考虑测试效率的软件可靠性模型。

## 1.2 模型的假设

模型假设如下:

(1) 错误排除的过程是非齐次泊松过程;

(2) 在任何时间序列  $t_0 < t_1 < \dots < t_m$  构成的时间区间  $(t_0, t_1)$ ,  $(t_1, t_2)$ ,  $\dots$ ,  $(t_{m-1}, t_m)$  中检测到的错误数是相互独立的;

(3) 软件失效由残留错误引起,失效强度与残留错误数与测试效率的乘积成正比;

(4) 测试阶段中发现故障后,因调试未必能真正排除故障,所以调试过程中排除故障的概率为  $p$ ,排错时间忽略不计;

(5) 因为调试过程本身固有的不完美的特点,所以排除故障的过程中,故障未必就真正被排除,可能引入的故障正比于被排除的错误数量,概率为  $q$ ;

(6) 软件的故障检测率  $b(t)$  不是一个常数,而是一个随时间增长不断递减的函数,记为  $b(t) = \alpha e^{-\beta t}$ ;

(7) 在测试过程中,考虑测试资源的投入、测试环境以及测试人员精力投入等问题,认为测试效率是一个递减的函数,记为  $\phi(t)$ 。

在以往的研究中<sup>[5-8]</sup>,也涉及了一些关于测试资源投入的问题。文中提出一种测试效率函数,来表示上述过程,记:

$$\phi(t) = \lambda t e^{-\gamma t}$$

## 1.3 模型的建立

根据式(1),结合假设(4),(7),式(1)可改写成

$$\frac{dm(t)}{dt} = b(t)[a(t) - pm(t)]\phi(t) \quad (3)$$

假设(5)认为在排错过程中,可能引入的错误与被排除的错误数成正比,比例为  $q$ ,因此软件错误总数  $a(t)$  就不是常数,而是随时间变化的函数,可表示为:

$$a(t) = a_0 + pqm(t) \quad (4)$$

式中,  $a_0$  表示测试开始时软件中的错误总量。

再根据假设(6),(7),公式(3)应改写成:

$$\frac{dm(t)}{dt} = b(t)[a - pm(t) + pqm(t)]\phi(t) \quad (5)$$

$$\frac{dm(t)}{dt} = \alpha e^{-\beta t} (a_0 - pm(t) + pqm(t)) \lambda t e^{-\gamma t} \quad (6)$$

$$m(t) = \frac{a_0 - e^{p(q-1)} \int_{t_0}^t \alpha e^{-\beta \tau} \lambda t e^{-\gamma \tau} d\tau + C}{p(1-q)} \quad (7)$$

因为  $m(0) = 0$ ,可以解出  $C = \ln a_0$ ,可得:

$$m(t) = \frac{a_0}{p(1-q)} \left( 1 - e^{p(q-1)} \int_{t_0}^t \alpha e^{-\beta \tau} \lambda t e^{-\gamma \tau} d\tau \right) \quad (8)$$

$$m(t) =$$

$$\frac{a_0}{p(1-q)} \left( 1 - e^{p(q-1)} \left( -\frac{\lambda \alpha}{\beta + \gamma} (t e^{-(\beta + \gamma)t} - \frac{1}{\beta + \gamma} (1 - e^{-(\beta + \gamma)t})) \right) \right) \quad (9)$$

令  $A = \beta + \gamma$ ,式(9)可以简化为:

$$m(t) = \frac{a_0}{p(1-q)} \left( 1 - e^{p(q-1)} \left( -\frac{\lambda \alpha}{A} (t e^{-At} - \frac{1}{A} (1 - e^{-At})) \right) \right) \quad (10)$$

式(10)中共有六个参数,可以通过参数估计的方法求出其近似解。

## 1.4 参数估计

式(10)中含有六个未知数:  $a_0, p, q, \alpha, \lambda, A$ 。要想使用文中提出的模型进行错误的预测就必须知道这几个参数的具体值。文中采用最大似然法对  $a_0, b, p, q, \alpha, \beta$  六个参数进行推导和计算,对均值函数  $m(t)$  构造最大似然函数过程如下:

$$L = \prod_{k=1}^n \frac{[m(t_k) - m(t_{k-1})]^{(y_k - y_{k-1})}}{(y_k - y_{k-1})!} \exp\{-[m(t_k) - m(t_{k-1})]\} \quad (11)$$

对上式两边取自然对数有

$$\begin{aligned} \ln L &\approx \sum_{k=1}^n \{(y_k - y_{k-1}) \ln[m(t_k) - m(t_{k-1})] - \\ &\quad [m(t_k) - m(t_{k-1})]\} - \sum_{k=1}^n \ln(y_k - y_{k-1}) \\ &= \sum_{k=1}^n (y_k - y_{k-1}) \ln[m(t_k) - m(t_{k-1})] - \\ &\quad \sum_{k=1}^n [m(t_k) - m(t_{k-1})] - \sum_{k=1}^n \ln(y_k - y_{k-1}) \quad (12) \end{aligned}$$

式中

$$m(t_k) = \frac{a_0}{p(1-q)} (1 - e^{p(q-1)(-\frac{\Delta\alpha}{A}(te^{-At_k} - \frac{1}{\beta+\gamma}(1-e^{-At_k})))}) \quad (13)$$

对  $\ln L$  分别求关于  $a_0, p, q, \alpha, \lambda, A$  的偏导数, 并令其为零, 得到

$$\frac{\partial \ln L}{\partial a_0} = 0, \frac{\partial \ln L}{\partial p} = 0, \frac{\partial \ln L}{\partial q} = 0, \frac{\partial \ln L}{\partial \alpha} = 0, \frac{\partial \ln L}{\partial \lambda} = 0, \frac{\partial \ln L}{\partial A} = 0 \quad (14)$$

$$\text{st. } 0 < a_0 < \infty, 0 < p < 1, 0 < q < 1, 0 < b < 1, \alpha > 0, 0 < A < 1$$

可以应用牛顿迭代法或弦截法求解上述几个等式构成的非线性方程组, 从而解出  $a_0, p, q, \alpha, \lambda, A$  的值。

## 2 实验数据与仿真

### 2.1 模型的拟合能力和预测能力的常用评价参数

(1) 模型的常用拟合能力参数  $SSE^{[9]}$ 。

$SSE$  用来描述累计失效数的观测值与预测值之间的距离, 定义如下

$$SSE = \sum_{i=1}^n (y_i - \hat{m}(t_i))^2 \quad (15)$$

$SSE$  越小, 则表示曲线的拟合能力越好。

(2) 回归曲线方程的相关指数 R-Square。

$$R - \text{Square} = \frac{\sum_{i=1}^n (\hat{m}(t_i) - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (16)$$

$$\text{其中 } \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

R-Square 越接近于 1, 则表示曲线的拟合能力越好。

(3) 模型的常用预测能力参数 AE。

模型的预测能力 AE 的定义如下

设  $t_0 < t_1 < t_2 < \dots < t_{n-1} < t_n$  表示失效间隔,  $\{t_i, y_i\}$  表示失效时间间隔数据。  $y_i$  表示到时间  $t_i$  为止的累计故障数。那么 AE 的定义可以表示为:

$$AE = \frac{1}{m} \sum_{i=1}^n \left| \frac{D_i - D}{D} \right| \quad (17)$$

AE 的值越小, 那么表示模型的预测能力越强。

### 2.2 数据分析及仿真实验

(1) 第一组数据。

本次仿真试验的数据来自文献[10]。文中利用该组数据, 将 G-O 模型、原模型<sup>[11]</sup>以及文中所提出的新模型进行比较。模型参数估计及仿真结果如表 1 和图 1 所示。

表 1 模型参数 I

参数名	$a_0$	$A$	$p$	$q$	$\alpha$	$\lambda$
参数值	109.734	0.174	0.849	0.095	0.614	0.101

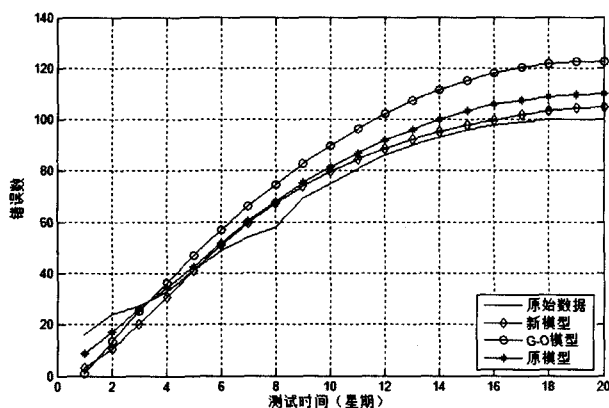


图 1 G-O 模型、原模型和改进后模型对实际数据的拟合曲线 I

通过图 1 对试验数据的拟合情况, 可以明显得出: 新模型的预计错误数比 G-O 模型和原模型更加接近于实际错误数。另外, 新模型在对未来错误数的预测方面体现出了一定的优势。通过对预计错误数的比较和拟合曲线的比较可以得出新模型的预测效果要好于 G-O 模型和原模型。各模型相应的评测标准数据如表 2 所示。

表 2 G-O 模型、原模型和改进后模型比较结果 I

模型	SSE	R-Square	AE
G-O 模型	5145	2.0061	0.2418
原模型	950.1313	1.4542	0.1103
新模型	668.0613	1.3119	0.1024

显然, 从表 2 可以得知, 新模型的拟合性能强于其它两个模型。

(2) 第二组数据。

本次仿真试验的数据也来自文献[10]。文中利用该组数据, 将 G-O 模型、原模型以及文中所提出的新模型进行比较。模型参数估计及仿真结果如表 3 和图 2 所示。

通过图 2 对试验数据的拟合情况, 可以明显得出: 新模型的预计错误数比 G-O 模型和原模型更加接近

于实际错误数。另外,新模型在对未来错误数的预测方面体现出了一定的优势。通过对预计错误数的比较和拟合曲线的比较可以得出新模型的预测效果要好于 G-O 模型和原模型。各模型相应的评测标准数据如表 4 所示。

表 3 模型参数 II

参数名	$a_0$	$\Lambda$	$p$	$q$	$\alpha$	$\lambda$
参数值	128.970	0.173	0.904	0.102	0.59	0.0973

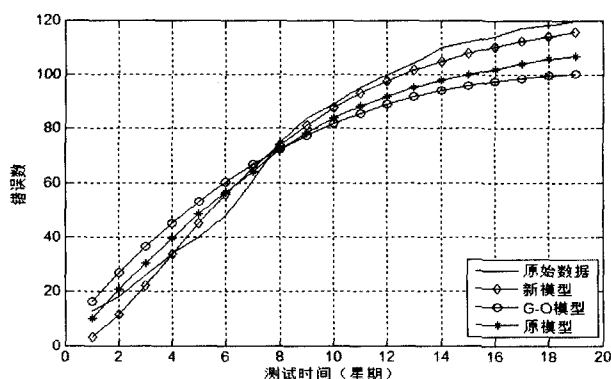
图 2 G-O 模型、原模型和改进后模型  
对实际数据的拟合曲线 II

表 4 G-O 模型、原模型和改进后模型比较结果 II

模型	SSE	R-Square	AE
G-O 模型	2976.7	0.5208	0.1878
原模型	1406.8	0.6978	0.1115
新模型	404.8399	0.9826	0.1054

同样,新模型的拟合性能强于其它两个模型。

通过以上两组数据的实例仿真,可以很明显地看出新模型在各项评测指标上都优于其它两种模型,尤其在对未来错误数的预测上,有着一定的优势。综上,文中提出的新模型是成功的。

### 3 结束语

在深入研究已发表的 NHPP 模型的基础上,本节对软件可靠性模型的假设条件进行了修改和完善。在测试过程中,投入的资源(人力、物力、财力等)、测试环境以及测试人员的努力程度对测试的结果有一定的

影响。基于此,结合错误发现率函数,并考虑了软件排错过程中新错误的引入和错误的不完全排除等情况,提出了一种考虑测试效率的可靠性模型。最后,引用两组已发表的数据进行实例仿真,结果显示,新模型有着良好的拟合能力和预测性能。综上,文中提出的新模型是成功的。

### 参考文献:

- [1] 邹丰忠,刘海青,王 林. 软件可靠性综合模型[J]. 武汉大学学报(工学版),2003,36(1):86-88.
- [2] 蔡开元. 软件可靠性工程基础[M]. 北京:清华大学出版社,1995.
- [3] 徐仁佐. 软件可靠性工程[M]. 北京:清华大学出版社,2007.
- [4] 黄锡滋. 软件的可靠性与完全性[M]. 北京:科学出版社,1995.
- [5] Huang Chin-Yu, Kuo Sy-Yen, Lyu M R. An Assessment of Testing-Effort Dependent Software Reliability Growth Models [J]. IEEE Transaction on Reliability, 2007, 56(2):198-211.
- [6] Huang Chin-Yu, Kuo Sy-Yen, Chen Ing-Yi. Analysis of a Software Reliability Growth Model with Logistic Testing-Effort Function[C]//Proc of 8th International Symposium on Software Reliability Engineering. [s. l.]: [s. n.], 1997:378-388.
- [7] Yamada S. Software-Reliability Growth with a Weibull Test-Effort: A Model & Application [J]. IEEE Transactions on Reliability, 1993, 42(1):100-106.
- [8] Kuo Sy-Yen, Huang Chin-Yu, Lyu M R. Framework for Modeling Software Reliability, Using Various Testing-Efforts and Fault-Detection Rates [J]. IEEE Transaction on Reliability, 2001, 50(3):310-320.
- [9] Malaiya Y K, Karunanithi N, Verma P. Predictability measures for software reliability models [C]//Proc. 1990 IEEE Int. Computer Software and Applications Conference, Chicago, IL: [s. n.], 1990:7-12.
- [10] 顾林君. 一种错误发现率为时间函数的软件可靠性模型 [D]. 南京:南京邮电大学,2009.
- [11] 丁晓光. 基于测试覆盖的软件可靠性模型 [D]. 南京:南京邮电大学,2008.

(上接第 66 页)

- [9] Thomas K, Thao N, Linh L. A Software as a Service with Mtenancy Support for an Electronic Contract Management Application [C]//IEEE International Conf on Services Computing. [s. l.]: IEEE Computer Society, 2008:179-186.
- [10] 周 健,孙丽艳. 面向对象 XML 的存储模式的研究[J]. 计

算机技术与发展, 2009, 19(3):114-122.

- [11] 王安全. SAAS 模式下大数据量统计框架的研究和实现 [J]. 计算机技术与自动化, 2009, 28(2):128-130.
- [12] 潘海兰, 吴翠红, 葛晓敏. XML 及其在 MVC 模式中的应用 [J]. 计算机技术与发展, 2010, 20(2):202-205.

《计算机技术与发展》欢迎投稿, 欢迎订阅!