计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
43(5):881~887,2006

考虑测试环境和实际运行环境的软件可靠性增长模型

赵 靖 刘宏伟 崔 刚 杨孝宗

(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

(zhaoj@ftcl.hit.edu.cn)

A Software Reliability Growth Model Considering Testing Environment and Actual Operation Environment

Zhao Jing, Liu Hongwei, Cui Gang, and Yang Xiaozong

(*School of Computer Science and Technology, Harbin Institute of Technology, Harbin* 150001)

Abstract The testing and operation environment may be essentially different, thus the fault detection rate of testing is different from that of the operation phase. Software reliability growth models (SRGMs) based on the non-homogeneous Poisson process (NHPP) are quite successful tools that have been proposed to assess the reliability of software. The constant environmental factor is proposed by some authors to describe the mismatch between the testing environment and operation environment in SRGMs of NHPP. Actually, the environmental factor ought to be varying with testing time. The varying environmental factor with time can be derived from actual failure data set. The fault detection rate (FDR) of operation is transformed from that of testing phase and varying environmental factors, considering the fault remove efficiency and fault introduction rate, and then an NHPP model PTEO-SRGM is presented. Finally, the unknown parameters are estimated by the least-squares method based on two failure data sets. Experiments show that the goodness-of-fit and predictive power of PTEO-SRGM is better than those of other SRGMs on these two data sets.

Key words software reliability growth model; non-homogeneous Poisson process; environmental function; imperfect debugging

摘 要 软件可靠性增长模型中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. 非齐次泊松过程类软件可靠性增长模型是评价软件产品可靠性指标的有效工具. 在一些非齐次泊松过程类模型中,有些学者提出了常量的环境因子,用来描述测试环境和运行环境的差别. 实际上,环境因子应该是随时间变化的变量. 考虑了运行阶段和测试阶段环境的不同,根据实测数据得到了变化的环境因子,并且根据测试阶段的故障检测率和变化的环境因子,转化得到了操作运行阶段的故障检测率. 考虑到故障的排除效率和故障引入率,从而建立了一个既考虑运行环境和测试环境差别,又考虑故障排除效率和故障引入率的非齐次泊松过程类软件可靠性增长模型(PTEO-SRGM). 在两组失效数据上的实验分析表明,对这组失效数据,PTEO-SRGM 模型比 G-O 模型等模型的拟合效果和预测能力更好.

关键词 软件可靠性增长模型;非齐次泊松过程;变化的环境因子;不完美调试

中图法分类号 TP311.5

收稿日期:2005-08-11;**修回日期**:2005-11-17

基金项目:国家自然科学基金项目(60503015);教育部博士点基金项目(20020213017)

1 引 言

随着计算机应用领域的不断拓展,软件的规模越来越大,结构和功能越来越复杂,人们对高质量软件的需求也更加迫切.软件可靠性是软件质量的重要指标之一,为了评估和预测软件产品的可靠性,一系列基于非齐次泊松过程(non-homogeneous Poisson process, NHPP)的软件可靠性增长模型(software reliability growth model, SRGM)被相继提出[1].基于NHPP的SRGM已经成为软件可靠性工程实践中非常成功的工具.软件测试是提高软件质量的主要手段,软件测试时间越长软件的质量越高,软件的可靠性就越高.同时软件的测试对开发成本和软件的交付时间有很大的影响.什么时候停止测试发布软件,发布后的软件在运行阶段是否可靠是一个重要的问题,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是否可靠,而且为软件的成本模型提供了很好的依据[2].绝大多数软件可靠性增长模型都假设测试和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,因而在运行和测试阶段软件的故障检测是不同的[3,4],但很少有软件可靠性模型会考虑到这一点,所以不能够精确地预测软件可靠性.

有些学者认为建立软件可靠性增长模型时应考虑测试资源的消耗[5,6],并从软件可靠性评价和开发管理方面考虑,根据实测数据和经验讨论了测试阶段和运行阶段的故障检测率的先增后减的变化趋势.一些学者通过加速测试的方法,提出了一个动态的模型用来描述运行阶段的可靠性增长模型[7].最近有些学者提出了环境因子的概念用来描述测试和运行环境的差别[8,9],环境因子被定义为 $k=\bar{b}_{\text{test}}/\bar{b}_{\text{field}}$,其中,$\bar{b}_{\text{test}}$和 $\bar{b}_{\text{field}}$分别代表平均的测试与运行阶段的故障检测率(FDR).这些学者认为环境因子是常数.实际上,软件中隐藏的故障被检测到的概率并不相同,容易检测到的故障首先被检测出来并排除掉,随着测试的进行,剩余的故障平均被检测到的概率会降低.在操作运行阶段,如果没有其他因素的影响,剩余故障被检测的概率会变得越来越低,FDR应是一个递减的函数,因此,环境因子应该是随测试时间变化的函数.

本文考虑了运行阶段和测试阶段环境的不同,从实测数据讨论了变化的环境因子.根据测试阶段的故障检测率和变化的环境因子,转化得到了操作运行阶段的故障检测率.为了更准确地描述软件的测试过程,在可靠性建模时考虑到因不完美调试而引入的新故障以及因排错延迟而产生的排错效率,从而得到了既考虑运行环境和测试环境差别,又考虑故障排除效率和故障引入率的软件可靠性增长模型(PTEO-SRGM).最后,利用数据进行了曲线拟合实验.结果表明,这个SRGM对某些失效数据集有很好的拟合精度和预测能力.

2 NHPP类SRGMs

注释:

$N(t)$:$t\geqslant 0$,一个计数过程表示到时刻 t 检测到的累积故障数;

a:软件系统中潜伏的故障总数的初始值;

t_0:软件开始测试时间;

m_0:软件开始测试时已经发现并且除掉的故障数,$m_0=m(t_0)$,当 $t_0=0$ 时,$m(t_0)=0$;

$a(t)$:基于时间的潜伏故障总数的函数,即到时刻 t 已排除的软件故障数和潜伏在软件中尚未被发现的软件故障数的和;

b:FDR 的初始值;

$b(t)$:基于时间的 FDR 函数,即软件中每个故障在时刻 t 被检测到的平均概率;

$\lambda(t)$:失效密度函数,即在时刻 t 软件单位时间内的失效数;

$m(t)$:到时刻 t 为止能够发现的故障数的期望值;

$R(x\mid t)$:可靠度函数,即 Pr{系统在$(t,t+x)$时间内不发生失效|最近一次失效发生在时刻 t}.

NHPP类型的SRGM用一个计数过程$\{N(t),t\geqslant 0\}$表示到时刻 t 为止检测到的软件故障累计数.如果故障累计数的期望值函数用 $m(t)$表示,则一个基于NHPP过程的SRGM的一般形式如下所示:

$$P\{N(t)=n\}=\frac{m(t)^n}{n!}e^{-m(t)}. \tag{1}$$

失效密度函数为

$$\lambda(t)=\frac{\mathrm{d}m(t)}{\mathrm{d}t}. \tag{2}$$

如果最后一次失效发生在测试时刻 t,则被测软件在时间段$(t,t+s)$内的可靠度为

$$R(x\mid t)=\exp[-(m(t+x)-m(t))]. \tag{3}$$

FDR 可表示为

$$b(t) = \frac{\lambda(t)}{a(t) - m(t)}. \tag{4}$$

G-O 模型假设 FDR 在测试过程中保持不变，被发现的故障立即被排除，排除故障时不引入新的故障. 因此：

$$m(t) = a \times (1 - \exp(-b \times t)). \tag{5}$$

考虑到改正软件故障时可能会引入新的故障以及软件的 FDR 会随时间而变化，一个 NHPP 类 SRGMs 的通用形式如下所示[10]：

$$m(t) = \exp(-\int_{t_0}^{t} b(\tau)\mathrm{d}\tau) \times \left[m_0 + \int_{t_0}^{t} a(\tau) \times b(\tau) \times \exp(\int_{t_0}^{t} b(\tau)\mathrm{d}\tau)\mathrm{d}\tau\right]. \tag{6}$$

3　随软件测试时间变化的环境因子

NHPP 类的 SRGM 包括 G-O 模型[10]、Yamada delayed S-shaped 模型[11]、Logistic growth curve 模型[11]、Goel generalized NHPP 模型[12]、Logarithmic Poisson 模型和 Compertz growth 模型[13]等. 在各种 SRGM 中，两个最重要的参数就是软件初始故障数和故障检测率. 故障检测率表示当前时刻单位时间内每个故障被检测到的平均概率，用来衡量测试的效率，是测试技术和测试环境的函数. 测试阶段故障检测率函数主要有下面 3 种：

① 常数 b，G-O 模型[11]；

② 关于测试时间的增函数，例如 PNZ-SRGM 模型[10]；

③ 关于测试时间的减函数，G-O S-形模型[12].

实际上，软件中隐藏的故障被检测到的概率并不相同，容易检测到的故障首先被检测出来并排除掉，随着测试的进行，剩余的故障平均被检测到的概率会降低. 在实际运行阶段，如果没有其他因素的影响，剩余的故障被检测到的概率会越来越低，运行阶段 FDR 应是一个递减的函数，并且经过了足够长的时间还没被发现的故障，其被检测到的概率趋近于 0. 软件测试阶段的 FDR 无论是哪种情况，环境因子都应是随着时间变化的函数.

为了近似描述变化的环境因子，定义环境因子为

$$k(t) = b_{\text{test}}(t)/b_{\text{field}}(t), \tag{7}$$

$b_{\text{test}}(t)$和 $b_{\text{field}}(t)$分别代表测试和运行阶段的故障检测率. 为了近似地描述环境因子的变化趋势，根据实测数据，用离散的随时间变化的均值描述环境因子的变化趋势. 因此，随时间变化的环境因子均值被定义为

$$\bar{k}(t) = \bar{b}_{\text{test}}(t)/\bar{b}_{\text{field}}(t), \tag{8}$$

$\bar{b}_{\text{best}}(t)$和 $\bar{b}_{\text{field}}(t)$分别代表测试和运行阶段随时间变化的平均的故障检测率.

因为测试阶段的失效数据呈现凹形[9]，用 G-O 模型[11]、Goel generalized NHPP 模型[13]、Logarithmic Poisson 模型[14]拟合测试阶段的失效数据，结果如表 1 所示. 模型的预测能力[8] p 定义为

$$p = \frac{\overline{N}(T) - N_{\text{field}}}{N_{\text{field}}}, \tag{9}$$

假设软件测试阶段在时间 T 结束，然后进入到软件的现场运行阶段. $\overline{N}(T)$表示软件测试阶段结束后估计的剩余故障数；N_{field}表示在运行阶段实际观察的故障数. 表 1 列出了以上各模型拟和测试阶段数据的拟和能力和预测能力. 从表 1 可以看到，G-O 模型与其他模型比较具有很好的拟和与预测能力，因此，测试阶段的失效数据选用 G-O 模型拟和.

Table 1　Comparisons Among the Goodness-of-fit and Predictive Power of G-O Model，Goel Generalized NHPP Model，and Logarithmic Poisson Model

表 1　G-O 模型、Goel generalized NHPP 模型、Logarithmic 泊松模型拟合与预测能力比较

Model	SSE	R-square	P (%)
G-O	327.321	0.99765	20
Goel Generalized NHPP	469.327	0.99563	440.9
Logarithmic Poisson	791.482	0.99264	440.9

软件运行阶段的失效率可以通过式(10)计算，其中，a 表示用 G-O 模型拟合整个测试阶段和运行阶段的失效数据而得到的估计数；$m(t)$表示到时间 t 累计的期望失效数：

$$\lambda(t) = b_{\text{field}}(t) \times [a - m(t)]. \tag{10}$$

离散运行阶段的平均失效率可以用式(11)表示：

$$\bar{\lambda}(t_i) = \frac{N(t_i)}{t_i}, \tag{11}$$

$N(t_i)$表示到时间 t_i 实测的累计失效数. 如果 $m(t)$用 $N(t)$代替，$\lambda(t)$用 $\bar{\lambda}(t)$代替，那么随时间变化的平均的 FDR 可表示为

$$\bar{b}_{\text{field}}(t_i) = \frac{\bar{\lambda}(t_i)}{\hat{a} - N(t_i)}. \tag{12}$$

可以得到运行阶段随时间变化的平均 FDR，变化趋势如图 1 所示：

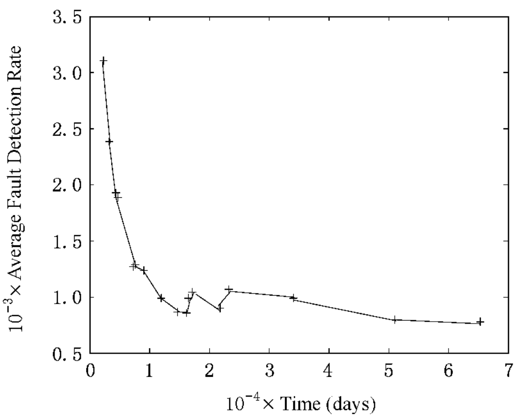

Fig. 1 Average fault detection rate of operational phase.
图 1 运行阶段平均的故障检测率

因此,随时间变化的离散的平均环境因子可表示为

$$\bar{k}(t_i) = \bar{b}_{\text{test}}(t_i)/\bar{b}_{\text{field}}(t_i), \tag{13}$$

可以得到环境因子 $\bar{k}(t)$,变化趋势如图 2 所示:

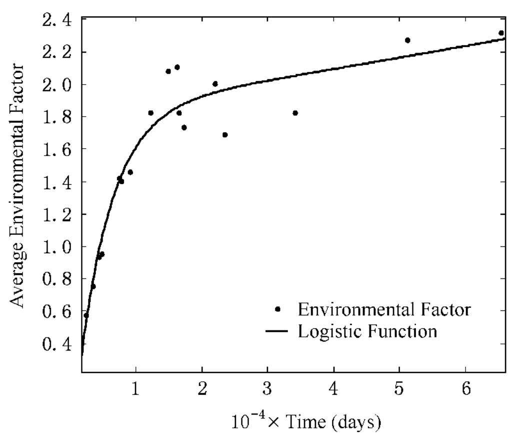

Fig. 2 Time-varying environmental factors of $\bar{k}(t)$.
图 2 随时间变化的环境因子

环境因子的这种非递减趋势可以用 Logistic 曲线(logistic curve)表示[15],Logistic 曲线在实际应用中有很多优秀表现[15],如式(14)所示:

$$\bar{k}(t) = \frac{N}{1 + A \times \exp(-\partial \times t)}. \tag{14}$$

对于软件产品,软件运行阶段的失效数据可以在以前发布后的版本中得到. 因此,可以根据经验数据得到环境因子[8,9],从而可以预测软件运行阶段的失效率.

4 考虑软件运行阶段和测试阶段不同的可靠性模型

研究结果表明[16],实验结果指出有 19% 的故障是以前已经检测到的重复性故障. 这表明,检测到的故障立即被排除的假设与软件测试的实际情况并不相符,因此在建立软件可靠性增长模型时要考虑故障的排除效率. 软件的调试过程是非常复杂的过程,在软件修复的过程中可能会引入新的故障. 为更准确地描述软件的测试过程,在可靠性建模时应考虑故障检测率函数、因不完美调试引入的新故障以及因排错延迟产生的故障排除效率函数. 如何从测试阶段的故障检测率(fault detection rate) *FDR* 得到运行阶段的故障检测率呢?由式(8)和式(14)可得运行阶段的 *FDR*,用式(15)表示:

$$\bar{b}_{\text{field}}(t) = \frac{\bar{b}_{\text{test}}(t)}{\bar{k}(t)} = \frac{b}{\bar{k}(t)} = \frac{b}{N}(1 + A \times \exp(-\partial \times t)). \tag{15}$$

本节建立一个既考虑环境因素,又考虑故障排除效率和故障引入率的从测试阶段到运行阶段的软件可靠性增长模型. 本节中用到的符号及含义说明如下:

$a(t)$:与时间相关的故障总数函数,包括软件中的初始故障和故障排除时引入的新故障;

$b(t)$:与时间相关的故障检测率函数,单位时间内每个故障被发现的概率;

$m(t)$:中值函数,到时刻 t 能够发现的软件故障总数的期望值;

a:软件的初始故障数;

p:故障排除率;

q:故障引入率.

本模型的假设:

(1) 无论是测试阶段或者运行阶段,软件失效都分别遵循一个 NHPP 过程;

(2) 故障的排除效率描述了故障被正确排除的概率,是一个常数 p;

(3) 软件调试过程不完美,即调试时可能会引入新的软件故障,故障引入率是一个常数 q;

(4) 在任何时刻软件的失效率与软件中隐藏的故障数成正比;

(5) 测试阶段每个故障的严重性和被检测到的可能性大致相同;

(6) 运行阶段故障检测率是通过测试阶段故障检测率和变化的环境因子转化而得到的.

根据假设条件,推导测试阶段的均值函数为

$$\begin{cases} \dfrac{\mathrm{d}m_{\text{te}}(t)}{\mathrm{d}t} = b_{\text{te}}(t) \times (a(t) - p \times m_{\text{te}}(t)), \\ \dfrac{\mathrm{d}a(t)}{\mathrm{d}t} = q \times \dfrac{\mathrm{d}m_{\text{te}}(t)}{\mathrm{d}t}. \end{cases} \quad t \leqslant T. \tag{16}$$

这个方程组的初始条件为

$$m(0)=0, a(0)=a.$$

可以得到测试阶段的均值函数:

$$m_{\mathrm{te}}(t)=\frac{a}{p-q}(1-\exp(-(p-q)\times bt)), t\leqslant T. \tag{17}$$

那么,对于运行阶段的均值函数,假设运行阶段也可以被看做一个故障检测过程,而且此过程可以用 NHPP 表示[6],那么运行阶段的均值函数 $m^o(t)$ 推导如下:

$$\begin{cases}\dfrac{\mathrm{d}m^o(t)}{\mathrm{d}t}=b_{\mathrm{field}}(t)[a(t)-p\times m_{\mathrm{te}}(T)-P\times m^o(t)], & t>T, \quad (18)\\ \dfrac{\mathrm{d}[a(t)-p\times m_{\mathrm{te}}(T)]}{\mathrm{d}t}=q\times\dfrac{\mathrm{d}m^o(t)}{\mathrm{d}t}, & t>T, \quad (19)\end{cases}$$

其中,运行阶段的故障检测率 $b_{\mathrm{field}}(t)$ 可以用式(15)中的 $\bar{b}_{\mathrm{field}}(t)$ 近似地代替,因此,运行阶段的均值函数可以表示为

$$m^o(t)=[a(\infty)-p\times m_{te}(T)]\cdot[1-\mathrm{e}^{-(p-q)\times B^*(t)}], \quad t>T, \tag{20}$$

其中,

$$B^*(t)=\int_T^t b_{\mathrm{field}}(\tau)\mathrm{d}\tau, \quad t>T \tag{21}$$

将式(15)代入式(21),可得:

$$B^*(t)=\int_T^t\frac{b}{N}(1+A\times\exp(-\partial\times t))\mathrm{d}\tau=(b/N)\times((t-T)-(A/\partial)\times(\exp(-\partial\times t)-\exp(-\partial\times T))), \quad t>T. \tag{22}$$

从而可得测试和运行阶段的均值函数如下:

$$m(t)=\begin{cases}\dfrac{a}{p-q}(1-\mathrm{e}^{-(p-q)bt}), & t\leqslant T\\ a\mathrm{e}^{-(p-q)bT}(1-\mathrm{e}^{-(p-q)(\frac{b}{N}\times((t-T)-(\frac{A}{\partial})\times(\mathrm{e}^{-\partial\times t}-\mathrm{e}^{-\partial\times T})))})+m(T), & t>T.\end{cases} \tag{23}$$

本模型既表示测试环境和运行环境差别又考虑故障排除效率,因此,被称为 PTEO-SRGM.

5　实验分析

5.1　比较标准

模型的拟和能力使用误差平方、SSE 和回归曲线方程的相关指数 $R\text{-}square$ 度量曲线拟合效果,这两个指标分别定义如下:

$$SSE=\sum_{i=1}^{n}(y_i-\hat{m}(t_i))^2, \tag{24}$$

$$R\text{-}square=\frac{\sum_{i=1}^{n}(\hat{m}(t_i)-\bar{y})^2}{\sum_{i=1}^{n}(y_i-\bar{y})^2}, \tag{25}$$

其中,n 表示失效数据集中失效样本的数量;$\hat{m}(t_i)$,表示到 t_i 时刻为止故障累计数的估算值;y_i 表示到 t_i 时刻为止故障累计数的实测值.

$$\bar{y}=\frac{1}{n}\sum_{i=1}^{n}y_i. \tag{26}$$

SSE 的值越小曲线拟合得越好;$R\text{-}square$ 的值越接近于 1 曲线拟合得越好.

模型的预测能力使用 AE 表示[17],$t_0<t_1<t_2<\cdots<t_n$ 表示失效间隔,$\{t_i, y_i\}$ 表示失效间隔数据,y_i 表示到时间 t_i 为止的累计故障数,AE 定义如下:

$$AE=\frac{1}{m}\sum_{i=l}^{u}\left|\frac{D_i-D}{D}\right|, \tag{27}$$

其中,t_l,t_u 表示用于估测模型预测能力所选的失效时间的上限和下限;D 表示软件测试结束时实际测得的累积故障数;D_i 表示用故障时间间隔 $\{t_0, y_0\}$,$\{t_1, y_1\}$,…,$\{t_i, y_i\}$ 所估测的软件测试结束时的累积故障数;m 表示位于 t_l,t_u 之间的时间间隔数. AE 的值越小模型的预测能力越强.

5.2　模型评估比较

为了检验 PTEO-SRGM 模型的拟合效果,用最小二乘法,在发表的第 1 组数据[9]和第 2 组数据[2]上,分别应用 PTEO-SRGM 模型、G-O 模型[11]、Yamada delayed S-shaped 模型[12]、logistic growth curve 模型[12]、Goel generalized NHPP 模型[13]、Logarithmic 泊松模型和 Compertz growth 模型[14]. 第 1 组数据模型预测能力 AE 的计算选用失效时间间隔的上限和下限分别为第 40 周和第 51 周,模型求解的参数值 N,A,∂ 分别为 3.291,2.749E-7 和 9.99. 最小二乘法估计的参数值、拟合结果和预测结果如表 1 所示. 由于第 2 组发表的数据是标准化数据,模型的最小二乘法估计参数值不代表实际的意义,因此没有给出参数值. 模型预测能力 AE 的计算选用失效时间间隔为运行阶段的后 20 组失效数据,模型在第 2 组数据拟合结果和预测结果如表 2 所示.

从表 2 和表 3 可以看出,在参与比较的 7 个模

型中,PTEO-SRGM 模型在这两组数据上的拟合效果和预测能力最好.

Table 2 Comparison Results of Reliability Estimation with Some SRGMs for the First Data Set

表 2 模型对第 1 组数据的评估结果对比

Model	Parameter Value				SSE	R-square	AE
	a	b	c	$p-q$			
PTEO-SRGM	26.33	0.0018			378.358	0.9976	0.00059
Yamada Delayed S-Shape	213.5	0.09348		0.127	806.3	0.9962	0.00581
G-O	301.3	0.02431			3914	0.9814	0.10611
General G-O	208	0.008258	1.543		565.8	0.9973	0.01162
Logarithmic Poisson	203.7	0.0377			4957	0.9764	0.12823
Logistic Growth Curve	198.9	12.41	0.1419		882.9	0.9958	0.03926
Compertz Growth	208.9	0.03101	0.9139		434.6	0.9979	0.00749

Table 3 Comparison Results of Reliability Estimation with Some SRGMs for the Second Data Set

表 3 模型对第 2 组数据的评估结果对比

Model	SSE	R-square	AE
PTEO-SRGM	0.0341	0.9926	0.0043
Yamada Delayed S-Shape	0.2443	0.9685	0.1205
G-O	0.0644	0.9917	0.0549
General G-O	0.06285	0.9919	1.7570
Logarithmic Poisson	0.9675	0.8753	0.6481
logistic Growth Curve	0.1191	0.9847	0.1073
Compertz Growth	0.3002	0.8972	0.7492

6 结 论

本文考虑了运行环境和测试环境的不同,通过变化的环境因子和测试阶段的故障检测率转化得到运行阶段的故障检测率,考虑到软件测试时排除效率,提出了一个被称为 PTEO-SRGM 的 SRGM. 并使用公开发表的两组失效数据集对这个模型的拟合效果和预测能力进行了评估. 实验结果表明,PTEO-SRGM 在这两组失效数据集上的拟合效果和预测能力较好,能够比较准确地评估和预测软件的可靠性.

参 考 文 献

1 M. R. Lyu. Handbook of Software Reliability Engineering. New York: McGraw-Hill and IEEE Computer Society Press, 1996

2 Xiaolin Teng, H. Pham. A software cost model for quantifying the gain with considering of random field environments. IEEE Trans. Computers, 2004, 53(3): 380~384

3 J. D. Musa. Sensitivity of field failure intensity to operational profile errors. In: Proc. 5th Int'l Symposium on Software Reliability Engineering. Piscataway, NJ: IEEE Press, 1994. 334~337

4 J. D. Musa. Adjusting measured field failure intensity for operational profile variation. In: Proc. 5th Int'l Symposium on Software Reliability Engineering. Piscataway, NJ: IEEE Press, 1994. 330~333

5 Sy-Yen Kuo, Chin-Yu Huang, Michael R. Lyu. Framework for modeling software reliability, using various testing-efforts and fault-detection rates. IEEE Trans. Reliability, 2001, 50(3): 310~320

6 Chin-Yu Huang, Sy-Yen Kuo, Michael R. Lyu, *et al*. Quantitative software reliability modeling from testing to operation. In: Proc. 11th Int'l Symposium on Software Reliability Engineering (ISSRE '00). Piscataway, NJ: IEEE Press, 2000

7 Hiroyuki Okamura, Tadashi Dohi, Shunji Osaki. A reliability assessment method for software products in operation phase-proposal of an accelerated life testing model. Electronics and Communications in Japan, Part 3, 2001, 84(8): 294~301

8 Xuemei Zhang, Daniel R. Jeske, Hoang Pham. Calibrating software reliability models when the test environment does not match the user environment. Applied Stochastic Models Business and Industry, 2002, 18: 87~99

9 Daniel R. Jeske, Xuemei Zhang, Loan Pham. Adjust software failure rates that are estimated from test data. IEEE Trans. Reliability, 2005, 54(1): 107~114

10 H. Pham, L. Nordmann, X. M. Zhang. A general imperfect-software-debugging model with S-shaped fault-detection rate. IEEE Transactions on Reliability, 1999, 48(2): 169~175

11 Goel A. L, Okumoto K. Time-dependent error-detection rate model for software and other performance measures. IEEE Transactions on Reliability, 1979, 28(3): 206~221

12 S. Yamada, M. ohba, S. Osaki. S-shaped software reliability

growth modeling for software error detection. IEEE Transactions on Reliability, 1983, 32(5): 475~484

13 A. L. Goel. Software reliability models: Assumptions, Limitations, and applicability. IEEE Transactions on Software Reliability, 1985, 11(12): 1411~1423

14 Chin-Yu Huang, Michael R. Lyu, Sy-Yen Kuo. A unified scheme of some nonhomogenous Poisson process models for software reliability estimation. IEEE Transactions on Software Engineering, 2003, 29(3): 261~269

15 Chin-Yu Huang, Sy-Yen Kuo. Analysis of incorporating logistic testing-effort function into software reliability modeling. IEEE Transactions on Reliability, 2002, 51(3): 261~270

16 M. Defamie, P. Jacobs, J. Thollembeck. Software reliability: Assumptions, realities and data. The IEEE Int'l Conference on Software Maintenance(ICSM '99), Oxford, England, 1999

17 Y. K. Malaiya, N. Karunanityi, P. Verma. Predictability measures for software reliability models. The 1990 IEEE Int'l Computer Software and Applications Conf. (COMPSAC '90), Chicago, IL, 1990

Zhao Jing, born in 1973. Ph. D. candidate. Her main research interests include software testing, software reliability evaluation, fault tolerance computing and wearable computing.

赵靖,1973 **年生,博士研究生,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.**

Liu Hongwei, born in 1971. Ph. D. and associate professor. His main research interests include software testing, software reliability evaluation, fault tolerance computing and wearable computing.

刘宏伟,1971 **年生,博士,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.**

Cui Gang, born in 1949. Professor and Ph. D. supervisor. His main research interests include fault tolerance computing, wearable computing, software testing, and software reliability evaluation.

崔刚,1949 **年生,教授,博士生导师,主要研究方向为容错计算、可穿戴计算、软件可靠性评估.**

Yang Xiaozong, born in 1939. Professor and Ph. D. supervisor. His main research interests include wearable computing, fault tolerance computing, software testing, and software reliability evaluation.

杨孝宗,1939 **年生,教授,博士生导师,主要研究方向为可穿戴计算技术、容错计算、软件测试和软件可靠性评估.**

Research Background

NHPP (nonhomogeneous Poisson process) models, as a class of SRGMs, are extensively used. NHPP SRGMs have been quite successful tools in practical software reliability engineering. Under the assumption that testing is performed in accordance with a given operational profile, SRGMs use the failure history, which is obtained during testing, to predict the field behaviors of the program. Actually, the testing profile and operational profile are not similar, in other words, there exist some differences between the testing reliability and operation reliability. Supported by the National Natural Science Foundation of China (Research on software reliability growth model of NHPP class considering testing environment and operational environment, grant No.60503015), we have studied the differences of testing environment and operational environment. In this paper, we focus on the evaluation accuracy and prediction precision of a software reliability growth model incorporating environmental function. Supported by the Ph. D. Programs Foundation of the Ministry of Education of China (research on dependability evaluation algorithms of computer system, grant No. 20020213017), we have worked on the evaluation algorithms for fault-tolerance and dependability. Some progress has been achieved. Furthermore, we will develop evaluation algorithms for dependability of a very large scale network system.