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
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(NHPP)的软件可靠性增长模型被相继提出。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。

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。但很少有软件可靠性模型会考虑到这一点,所以不能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,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。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。环境因子被定义为的平均的测试与运行阶段的故障检测率(当测试资源消耗时)。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,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。最后,利用数据进行了曲线拟合实验。结果表明,这个模型对某些失效数据集有很好的拟合精度和预测能力。注释:

1) 类似于记为到时刻检测到的累积故障数;%
2) 软件系统中潜伏的故障总数的初始值;
3) 软件开始测试时间;
4) 软件开始测试时已经发现并且除掉的故障数;
5) 当软件开始测试时已经发现并且除掉的故障数为时,
6) 是基于时间的潜伏故障总数的函数,即到时刻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;
7) 其初始值;
8) 是基于时间的函数,即软件中每个故障在时刻被检测到的平均概率;
9) 失效密度函数,即在时刻软件单位时间内失效数;
10) 到时刻为止能够发现的故障数的期望值;
11) 可靠度函数,即系统在时间段内不发生失效且最近一次失效发生在时刻。
考虑到改正软件故障时可能会引入新的故障以随软件测试时间变化的环境因子应是一个递减的函数,并且经过了足够时间的测试后,测试的检测率会随时间而变化,一个通用形式如下所示

\[ \frac{\partial}{\partial t} T(t) = \sum_{i=1}^{n} \frac{\partial}{\partial t} T_i(t) \]

\[ T(t) = \prod_{i=1}^{n} T_i(t) \]

因此,测试阶段的失效数据选用软件测试阶段结束后,然后进入到软件测试阶段在时间

\[ T(t) = \frac{T_0}{T_0 - N_{field}} \]

\[ N_{field} \]

表

<table>
<thead>
<tr>
<th>Model</th>
<th>SSE</th>
<th>R-square</th>
<th>( R^2 ) %</th>
</tr>
</thead>
<tbody>
<tr>
<td>G-O</td>
<td>327.321</td>
<td>0.99765</td>
<td>20</td>
</tr>
<tr>
<td>Goel generalized NHPP</td>
<td>469.327</td>
<td>0.99563</td>
<td>440.9</td>
</tr>
<tr>
<td>Logarithmic Poisson</td>
<td>791.482</td>
<td>0.99264</td>
<td>440.9</td>
</tr>
</tbody>
</table>

Table 1 Comparisons Among the Goodness-of-fit and Predictive Power of G-O Model, Goel Generalized NHPP Model and Logarithmic Poisson Model
Therefore, the time-varying average environmental factor can be expressed as
\[
\text{Average Environmental Factor} = \frac{N}{1 + A \times \exp(-\vartheta \times t)}.
\]

Figure 1: Average fault detection rate of operational phase.

Figure 2: Time-varying environmental factors of the operational phase.

The failure of software during the operational phase can be described by the logistic distribution function
\[
dm = b \times t \times \text{Field} \times m - p \times m \times t \times \text{Field},
\]
\[
dt = q \times \frac{dm}{dt}.
\]
其中，线方程的相关指数可以表示为故障排除效率，因此，被称为故障检测过程，而且此过程可以用模型的拟和能力使用误差平方、来表示到时间的上限和下限；

其中，表示用于估测模型预测能力所选的失效数据集中失效样本的数量；

表示用于估测模型预测能力所选的失效数据集所估测的软件测试结束时的累计故障数的实测值；

表示用故障时间间隔{表示失效间隔数据，{表示到时间{时刻为止故障累计数的估算值；

表示软件测试结束时实际测得的累计故障数；

表示用故障间隔间隔{表示失效间隔，{表示到时间{时刻为止的累计故障数，{表示时刻为止故障累计数的实测值；

表示用故障间隔间隔{表示失效间隔，{表示到时间{时刻为止的累计故障数，{表示时刻为止故障累计数的实测值；

表示软件测试结束时实际测得的累计故障数；

表示用故障间隔间隔{表示失效间隔，{表示到时间{时刻为止的累计故障数，{表示时刻为止故障累计数的实测值；

表示软件测试结束时实际测得的累计故障数；

表示用故障间隔间隔{表示失效间隔，{表示到时间{时刻为止的累计故障数，{表示时刻为止故障累计数的实测值；

表示软件测试结束时实际测得的累计故障数；

表示用故障间隔间隔{表示失效间隔，{表示到时间{时刻为止的累计故障数，{表示时刻为止故障累计数的实测值；

表示软件测试结束时实际测得的累计故障数；

表示用故障间隔间隔{表示失效间隔，{表示到时间{时刻为止的累计故障数，{表示时刻为止故障累计数的实测值；

表示软件测试结束时实际测得的累计故障数。
Table 2 Comparison Results of Reliability Estimation with Some SRGMs for the First Data Set

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameter Value</th>
<th>SSE</th>
<th>R-square</th>
<th>AE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTEO-SRGM</td>
<td>a 26.33, b 0.0018, c 0.127</td>
<td>378.358</td>
<td>0.9976</td>
<td>0.00059</td>
</tr>
<tr>
<td>Yamada Delayed S-Shape</td>
<td>a 213.5, b 0.0349, c 0.1419</td>
<td>806.3</td>
<td>0.9962</td>
<td>0.00581</td>
</tr>
<tr>
<td>G-O</td>
<td>a 301.3, b 0.02431, c 1.543</td>
<td>3914</td>
<td>0.9814</td>
<td>0.10611</td>
</tr>
<tr>
<td>General G-O</td>
<td>a 208, b 0.008258, c 1.543</td>
<td>565.8</td>
<td>0.9973</td>
<td>0.01162</td>
</tr>
<tr>
<td>Logarithmic Poisson</td>
<td>a 203.7, b 0.0377, c 1.543</td>
<td>4957</td>
<td>0.9764</td>
<td>0.12823</td>
</tr>
<tr>
<td>Logistic Growth Curve</td>
<td>a 198.9, b 12.41, c 0.1419</td>
<td>882.9</td>
<td>0.9958</td>
<td>0.03926</td>
</tr>
<tr>
<td>Comperz Growth</td>
<td>a 208.9, b 0.03101, c 0.9139</td>
<td>434.6</td>
<td>0.9979</td>
<td>0.00749</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Model</th>
<th>SSE</th>
<th>R-square</th>
<th>AE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTEO-SRGM</td>
<td>0.0341</td>
<td>0.9926</td>
<td>0.0043</td>
</tr>
<tr>
<td>Yamada Delayed S-Shape</td>
<td>0.2443</td>
<td>0.9685</td>
<td>0.1205</td>
</tr>
<tr>
<td>G-O</td>
<td>0.0644</td>
<td>0.9917</td>
<td>0.0549</td>
</tr>
<tr>
<td>General G-O</td>
<td>0.06285</td>
<td>0.9919</td>
<td>1.7570</td>
</tr>
<tr>
<td>Logarithmic Poisson</td>
<td>0.9675</td>
<td>0.8753</td>
<td>0.6481</td>
</tr>
<tr>
<td>logistic Growth Curve</td>
<td>0.1191</td>
<td>0.9847</td>
<td>0.1073</td>
</tr>
<tr>
<td>Comperz Growth</td>
<td>0.3002</td>
<td>0.8972</td>
<td>0.7492</td>
</tr>
</tbody>
</table>

6

8. Xuewei Zhang and Daniel R. Jeske and Hong 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
12. S. Yamada and M. oba and S. Osaki. S-shaped software reliability
growth modeling for software error detection. IEEE Transactions on Reliability 1983; 32: 475～484
15 Chin-Yu Huang, Sy-Yen Kuo. Analysis of incorporating logistic testing-effort function into software reliability modeling. IEEE Transactions on Reliability 2002; 51: 261～270

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

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.

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.

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.

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.