

考虑时变学习效应的软件可靠性模型

赵瑞曼, 潘冠华, 方建勇, 王 蕾

(江苏自动化研究所, 江苏 连云港 222006)

摘要:针对限制软件可靠性研究发展的瓶颈问题—软件可靠性水平受时间因素影响甚微与用户时间尺度直观度量软件可靠性需求间的矛盾,分析了时变学习效应对用户使用软件方式的影响。通过研究时间因素与用户累积经验间的关系,以及用户使用经验与维护软件可靠性之间的关系,建立了时间变量与软件可靠性指标之间的间接映射关系。通过引入时变学习率因子,对软件可靠性 Nelson 模型进行改进,提出了一个既符合软件实际特点,又满足用户度量需求的时间尺度可靠性度量模型。最后通过例证对所提模型进行了初步评价。

关键词:软件确认;学习效应;时变;软件可靠性度量

中图分类号:TP311.5

文献标识码:A

文章编号:1673-629X(2015)06-0039-05

doi:10.3969/j.issn.1673-629X.2015.06.009

Software Reliability Model Based on Time-varying Learning Effects

ZHAO Rui-man, PAN Guan-hua, FANG Jian-yong, WANG Lei

(Jiangsu Automation Research Institute of CSIC, Lianyungang 222006, China)

Abstract: For delivered software, time has little impact on its reliability. But from the user's perspective, software reliability should be measured on time scale. According to this controversy problem, the impact of a time-varying learning effect on using of software for users is analyzed. By studying the relationship between the time factor and the user experience accumulated, and between the user experience and the maintenance of software reliability, the indirect mapping relation between time variable and software reliability index is established. By introducing a time-varying factor, the software reliability model of Nelson is improved. Put forward a reliable measurement model of time scale both according with the actual characteristics of the software and meeting the demand of users to measure. In the end, a preliminary evaluation of the proposed model is given by an example.

Key words: software validation; learning effects; time-varying; measurement of software reliability

0 引言

随着计算机软件技术的迅速发展与扩大化应用,软件质量问题也受到了更加广泛的关注。可靠性作为衡量软件质量的关键指标之一^[1],一直以来缺少有效的实用度量方法。为定量分析软件可靠性水平提供数学基础,经过几十年不断深入的研究,截至目前学术界已经公开发表了百余种软件可靠性模型,但没有任何一种获得广泛一致的认可^[2]。软件可靠性研究至今,一直无法突破的瓶颈问题是如何以用户最直观理解的时间尺度度量几乎不受时间因素直接制约的软件可靠性水平。这也正是现有理论研究成果无法在软件可靠性测试实践领域获得广泛一致认可的根本原因。如何取得理论应用于实践的突破性进展,解决这一瓶颈问

题是关键。

软件的主体是程序。程序是使用计算机语言编写的,适合计算机执行的指令(语句)序列。这些指令序列能否实现预期目标或解决预期问题,取决于指令间的组合逻辑正确性。对于交付给用户的软件成品,其程序指令序列及其组合逻辑是不随时间变化的。这就导致软件缺陷水平,软件质量本身受时间因素直接影响甚微。另一方面,站在软件最终用户的角度,提出可靠性这一定量评价需求具有一定的硬件继承性。只有像硬件一样用可靠度、失效率、寿命、平均失效前时间(MTTF)、平均失效间隔时间(MTBF)等^[3]这些最为直观的时间尺度数据定量度量评价软件的可靠性水平,才能够为用户所理解并认同。因此又必须研究软

收稿日期:2014-07-01

修回日期:2014-10-09

网络出版时间:2015-05-06

基金项目:国家“十二五”装备预先研究项目(51319080202)

作者简介:赵瑞曼(1990-),女,硕士研究生,研究方向为系统工程、软件可靠性工程;潘冠华,研究员,硕士研究生导师,从事系统工程、舰船总体方向的研究。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20150506.1621.008.html>

件可靠性的时变关系。

文中针对上述度量需求与实际可度量性不匹配这一软件可靠性瓶颈问题,结合学习效应随时间增长与软件可靠性随学习效应增长双映射关系分析,通过对 Nelson 模型的分析与改进,提出了适用于软件确认的可靠性模型。理论分析与实验数据表明,所提模型既契合软件质量本身不随时间变化的实际,又满足用户时间尺度直观评价软件可靠性的需求,适合对实际软件进行可靠性度量。

1 学习因子对软件可靠性影响分析

根据软件失效机理^[4]:人在设计编码中无意犯下的错误(Error)导致工作成果与预期存在偏差从而产生了软件缺陷(Defect);软件执行到缺陷时表现与预期不符即发生故障(Fault);当故障传递至输出,最终产生了偏离需求的运行结果即发生软件失效(Failure)。由于人无意犯错不可避免,最终交付给用户的软件虽然从功能性能上已经达到用户的需求,但仍可能存在缺陷。这类缺陷一旦被某种运行条件触发导致软件失效,软件的可靠性就受到了质疑。显然,对于使用稳定且不作任何修改的软件而言,缺陷情况不随时间变化。但是软件用户可以依靠使用经验尽可能避免触发缺陷的运行条件,而这个使用经验是随学习时间增长而累积的过程。

已有最新研究^[5]认同时间因素通过人的学习效应间接影响软件可靠性水平。事实上,很多早期理论成果已经提及了人对软件可靠性的影响作用,如考虑人力资源消耗的 MUSA 时间模型^[6]、考虑完美排错过程的 G-O 模型^[7]等。而随着研究的进一步深入,更多的学者开始重视参与者能力水平这一重要影响因素,并将其考虑进软件可靠性建模研究中,如贾冀婷在充分考虑测试人员排错准确性和有效性的情况下,基于经典 Seeding 模型进行改进而提出的适用于软件测试的可靠性模型^[8];顾爱华等考虑测试人员学习过程使得故障检测率在满足随时间增长而先增后减的函数关系的情况下,提出改进 G-O 模型^[9]等。但是这些成果并没有明确阐述时间因素通过学习效应如何间接影响软件可靠性水平,并且所建立的模型仍然是可靠性指标关于时间变量的直接映射关系。

Chiu K C 最早将学习效应引入软件可靠性研究,通过改进已有的 NHPP 过程模型,建立了一个考虑定值学习效应的软件可靠性增长模型(SRGM)^[10]。Javaid Iqbal 等在同时考虑自主学习与被动学习两类学习模式影响效应的情况下,改进了 Chiu 提出的模型^[11]。考虑到学习效应对软件可靠性的影响在整个测评过程中可能不是固定不变的,Chiu 对所建模型再

次改进,提出了基于时变学习效应的软件可靠性增长模型^[12]。

上述模型大多是针对交付前测试阶段软件所建立的可靠性增长模型。其核心观点是:受学习效应影响的缺陷检测与排除活动促进了软件可靠性的增长。虽然此方面研究对通过测试提高软件质量,进而提升其可靠性水平本身具有非常重要的意义,但据此得到的度量结果却并非所交付软件的真实可靠性水平,甚至几乎完全不具备任何预测意义。主要原因在于以下两方面:一是用户与测试人员对软件的使用具有显著不同的倾向性(测试人员倾向于破坏,而用户倾向于维护);二是测试阶段所发现的缺陷残留至交付产品的概率非常低,用户使用过程中复现性差。因此这些模型仍然不能为用户所接受。

根据软件生命周期观点,时变学习效应通过以下三方面促进软件可靠性增长:在软件开发过程中提升设计与编程人员的工作质量;在测试过程中提高测试人员检测/排错效率;在交付使用后提升用户使用熟练度。以度量所交付软件产品可靠性,为用户提供有意义的软件可靠性数据为目的,文中研究只关注最后一类,即用户学习效应对所交付软件的可靠性的影响。考虑到产品确认阶段,软件质量本身几乎不受时间因素影响,Nelson 模型非常适合于这一阶段的可靠性度量。文中通过研究用户学习效应对软件输入选择的影响,改进 Nelson 模型,建立适用于确认阶段的软件可靠性模型。

2 考虑学习效应的软件可靠性模型

2.1 Nelson 模型简述

从本质上讲,软件可靠性最关注的仍然是程序数据处理正确率。Nelson 模型是数据域软件可靠性模型的代表。Nelson 模型把计算机程序看作对一个可计算函数 F 的说明,程序的输入域为:

$$E = \{E_i \mid i = 1, 2, \dots, N\} \quad (1)$$

其中, E_i 是能够使程序运行一次的输入数据; N 是程序可行输入数据总数。

输入数据 E_i 经程序运行后,得到的输出数据值为 $F'(E_i)$ 。

用 $F(E_i)$ 表示函数的真值,用 Δi 表示允许出现的最大误差值。则当

$$|F'(E_i) - F(E)| \leq \Delta i \quad (2)$$

时,程序运行成功。

如果程序在运行中发生非正常中断事件、无法终止运行的事件,或是程序运行得到的输出值 $F'(E_i)$ 使得

$$|F'(E_i) - F(E)| > \Delta i \quad (3)$$

时,程序运行发生失效。

将 E 划分为子集 E_n 和 E_f , 即:

$$E = E_n \cup E_f \quad (4)$$

其中, E_n 为保证程序正常运行的输入数据集合; E_f 为导致程序运行失效的输入数据集合。如果程序运行 m 次,每次运行输入数据的选择是独立的,则程序不发生失效的概率为:

$$R(m) = \left[\sum_{i=1}^n p_i (1 - y_i) \right]^m \quad (5)$$

式中, p_i 为 E_i 被选中的概率;而 y_i 满足:

$$y_i = \begin{cases} 1, & E_i \in E_f \\ 0, & E_i \in E_n \end{cases} \quad (6)$$

2.2 Nelson 模型应用

Nelson 模型一般应用于输入空间有限且离散的,小规模软件确认阶段的可靠性度量。其基本原理为在测试资源允许的条件下,从程序输入空间 E ,尽可能多且随机抽取输入数据,输入并运行程序。对所取输入数据 E_i ,统计其出现频次以计算选中概率 p_i ;统计其是否获得程序的正确处理,以划分其所属类别 E_n 或 E_f 。最后通过计算(5)式得到确认阶段软件的可靠性度量结果。具体流程如图1所示。

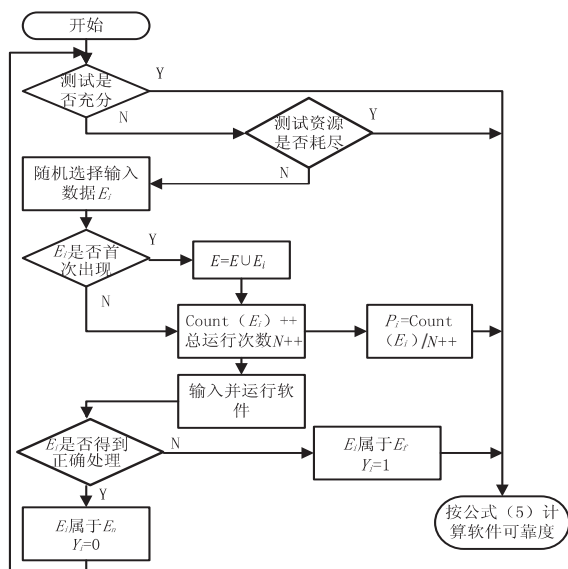


图1 Nelson 模型应用流程

虽然理论上,Nelson 模型仅适用于输入空间有限离散的小规模软件,但伴随各种基于模块化架构软件可靠性模型^[13-14]的提出,Nelson 模型的适用范围可以进一步扩展到对大规模软件基础模块的可靠性度量应用领域。并且,由于 Nelson 模型仅关注输入数据获得正确处理的概率这一静态数据,并不关心软件缺陷检测与处理这个动态变化过程,符合确认阶段固定版本的软件质量本身不随时间变化的实际情况。

目前 Nelson 模型应用的一个缺陷是:无论是软件

测试,还是用户真正的使用,都不可能完全随机地从整个输入空间选取输入数据。用户对输入数据的选择受到依赖于时变学习效应的使用经验影响。在软件已进入确认阶段,即可行数据处理空间在一定程度上匹配用户使用剖面空间的条件下,显然,越有经验的用户越了解怎样使用软件才不至于陷入崩溃,毕竟所接受的软件是在权衡各方利弊的条件下最好的选择。因此,随学习效应累积的经验会驱使用户选择一个有利于维护软件继续运行下去的输入数据,从而促进软件可靠性随用户熟练程度增长而增长。

2.3 Nelson 模型改进

为扩大 Nelson 模型的应用范围,改进其离散输入域描述 $E = \{E_i | i = 1, 2, \dots, N\}$ 为连续输入空间描述 I ,对该输入空间进行正/异常划分:

$$I = I_n + I_u \quad (7)$$

其中, I_n 表示程序处理正常的输入子空间; I_u 表示程序处理失效的输入子空间。分别用 $|I_n|$ 和 $|I_u|$ 表示对应输入子空间的规模。

在不考虑学习因子影响的条件下,用户选择一个输入落入正常与失效输入子空间的自然概率为:

$$p = \frac{|I_n|}{|I_n| + |I_u|} \quad (8)$$

$$q = 1 - p = \frac{|I_u|}{|I_n| + |I_u|} \quad (9)$$

在用户整个使用过程中是一个常数,对应 Nelson 模型的应用:

$$p = \sum_{E_i \in E_n} p_i \quad (10)$$

$$q = 1 - p = 1 - \sum_{E_i \in E_n} p_i = \sum_{E_i \in E_f} p_i \quad (11)$$

然而考虑到随着用户对软件使用时间,即学习时间 t 增长,其选择一个输入落入程序正确处理域的概率也随之增大。文中研究引入一个时变学习率因子 $\eta(t)$ 改进 Nelson 模型,使其更符合用户使用特点,对所交付软件具有更高的预测性。

在时变学习因子 $\eta(t)$ 的影响下,用户所选择的输入落在程序正常处理范围的概率随学习经验增长:

$$P(t) = \int_0^t \eta(\tau) d\tau \quad (12)$$

对于任意给定的时刻 t (产品交付用户首次使用开始计时为 $t=0$),用户的 n 次输入选择控制软件无失效运行,即落入程序正确处理范围的次数 k ,是一个随机变量,满足参数为经验选择概率 $P(t)$ 的二项分布。根据统计原理,此时软件正确处理用户输入的概率,即可靠度为:

$$R(t) = \binom{n}{k} P(t)^k (1 - P(t))^{n-k} \quad (13)$$

根据 k 与 $P(t)$ 的含义,显然有:

$$k(t) = nP(t) \quad (14)$$

考虑到次数 k 必然是一个整数,令:

$$k = \lfloor k(t) \rfloor \quad (15)$$

将依赖于时变学习因子 $\eta(t)$ 的经验概率 $P(t)$ (式(12))代入计算得到:

$$R(t) = \frac{n!}{k! (n-k)!} \left(\int_0^t \eta(\tau) d\tau \right)^k \left(1 - \int_0^t \eta(\tau) d\tau \right)^{n-k} \quad (16)$$

2.4 改进模型的参数估计

实际学习率因子 $\eta(t)$ 可以根据用户对具体软件的掌握速率,选择合适分布函数进行模拟估计。文中研究考虑到:随着学习时间 t 的增长,人对一个新事物的认识过程满足:完全不了解→逐渐增长了解→趋近于完全了解(达不到完全了解)这样一个链式增长过程。用户对于一个软件的认识使用同样如此。因此,从程序数据处理的角度,用户输入选择落入软件正确处理范围内的经验概率取值从 0 开始,随学习时间增

长,呈现单调递增态势,逐渐趋近于 1,但是不等于 1。

经典概率分布中的指数分布 $F(t) = 1 - e^{-\frac{t}{\alpha}}$ (如图 2 所示)恰好满足这一规律。

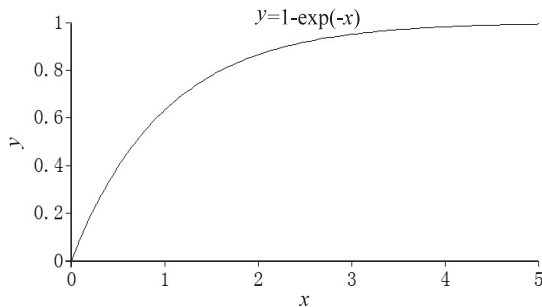


图 2 一个理想的输入选择经验概率函数曲线
因此文中假设学习率因子:

$$\eta(t) = \frac{1}{\alpha} e^{-\frac{t}{\alpha}} \quad (17)$$

根据式(12)得到,经验选择概率:

$$P(t) = 1 - e^{-\frac{t}{\alpha}} \quad (18)$$

将式(14)、(15)和(18)代入式(13)计算得到软件可靠性度量函数:

$$R(t) = \frac{n!}{\lfloor n * (1 - e^{-\frac{t}{\alpha}}) \rfloor! (n - \lfloor n * (1 - e^{-\frac{t}{\alpha}}) \rfloor)!} \left(1 - e^{-\frac{t}{\alpha}} \right)^{\lfloor n * (1 - e^{-\frac{t}{\alpha}}) \rfloor} \left(e^{-\frac{t}{\alpha}} \right)^{n - \lfloor n * (1 - e^{-\frac{t}{\alpha}}) \rfloor} \quad (19)$$

$\alpha_1, \alpha_2, \dots, \alpha_m$, 取:

$$\hat{\alpha} = \frac{1}{m} \sum_{i=1}^m \alpha_i \quad (23)$$

最后求得:

$$\hat{\alpha} = \frac{1}{m} \sum_{i=1}^m \frac{t_i}{\ln \left[\frac{1}{1 - k(t_i)/n} \right]} \quad (24)$$

考虑到人为因素各种不确定性,用矩估计方法获得的参数估计值比最大似然结果更有效。

3 模型初步评价

文中通过改进 Nelson 模型所建立的考虑学习效应的软件可靠性模型具有如下特点:

(1)以简单的数学公式阐明了软件可靠性随用户学习经验增长而增长的函数关系,具有简单、直观、设置参数少等优点;

(2)所建立的模型不关注软件本身缺陷状况,完全满足软件交付使用后缺陷及本身质量不受时间因素影响的实际情况;

(3)通过引入学习率因子,建立了软件可靠性水平与时间因素的间接映射关系,依据所获得的模型进行确认阶段的软件可靠性度量结果直观,易于理解;

(4)便于指导开展确认阶段的软件可靠性测试。

为对文中所提出的改进模型进行初步评价,给出

$$R(t) = \frac{n!}{\lfloor n * (1 - e^{-\frac{t}{\alpha}}) \rfloor! (n - \lfloor n * (1 - e^{-\frac{t}{\alpha}}) \rfloor)!}$$

在指数函数学习率因子(如式(17)所示)的条件下,模型唯一未知参数为 α ($\alpha > 0$)。可以利用最大似然估计法或者矩估计法进行参数估计。 α 的最大估计似然函数为:

$$L(t_1, t_2, \dots, t_m; \alpha) = \frac{1}{\alpha^m} \prod_{i=1}^m e^{-\frac{t_i}{\alpha}} \quad (20)$$

令:

$$\frac{\partial \ln L(t_1, t_2, \dots, t_m; \alpha)}{\partial \alpha} = 0 \quad (21)$$

得到:

$$\hat{\alpha} = \frac{\sum_{i=1}^m t_i}{m} \quad (22)$$

利用简单的矩估计进行 α 参数估计,具体操作按如下步骤进行:

(1)从用户群体中随机抽取能力相当的 N 个作为样本用户;

(2)以同一交付时间作为时间起点 $t = 0$,随机抽取时间序列 $T = \{t_1, t_2, \dots, t_m\}$ 作为模型参数估计的样本选取时间点;

(3)统计这 N 个样本用户在此时间序列 T 上选取的输入数据获得正确处理次数:

$$\{k(t_1), k(t_2), \dots, k(t_m)\}$$

(4)分别对 $i = 1, 2, \dots, m$ 计算 $nP(t_i) = k(t_i)$ 得到

下面一个软件实例:

针对某公司自研办公软件系统,从用户群体(公司全体员工)中选取工作经验五年以上能力相当的员工 $N=40$ 人作为样本用户;在以交付试用时间为起点 $t=0$ 的,三十天试用期(每天按工作时长 7.5 h 计算)共计 225 h 时间段内,等间隔选取测试时间序列 $T=\{t_1=180, t_2=360, \cdots, t_{75}=13\ 500\}$ (单位 min),统计每 180 min 该样本用户使用软件获得正确处理的频次 $\{k(t_1), k(t_2), \cdots, k(t_{75})\}$ (给出部分数据列表如表 1 所示)。

表 1 某公司办公系统软件部分实例数据

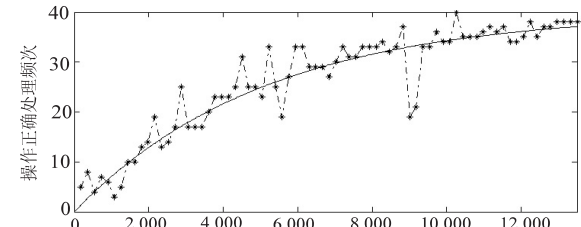
数据标号	T/min	$K/\text{次}$	$NP(t)$
1	180	5	1.494 1
2	360	8	2.932 4
3	540	4	4.317 0
...
61	10 980	36	36.077 5
...
75	13 500	38	37.698 0

对表 1 数据采用矩估计方法进行参数估计,计算式(24),得到 α 估计值 $\hat{\alpha}=5\ 142.8$ 。

代入式(18)中得到用户输入选择的经验概率函数:

$$P(t) = 1 - e^{-\frac{t}{5\ 142.8}}$$

实际数据与模型估计对比如图 3 所示。



模型估计与样本数据对比图: 真实 $k(t)$ (" * "); 估计 $NP(t)$ (" + ")

图 3 实例数据与模型估计对比图

从图中可以看出,由于文中改进的模型相对较简单,所以并不能很好地拟合所有的实验数据,但是模型的总体趋势走向与实验数据相匹配,并且该实验数据取值自产品试用阶段,因此对所交付产品具有一定的预测能力。最关键的是,文中所改进的模型简单实用,能够为用户提供满意的、易于理解的软件可靠性数据。

4 结束语

如何以用户最直观理解的时间尺度度量几乎不受时间因素直接制约的软件可靠性水平一直以来都是困扰软件可靠性研究与发展中的关键瓶颈问题。文中提出

的软件可靠性模型,既满足了软件质量本身受时间因素影响较小,又通过学习效应建立了软件可靠性水平与时间尺度度量标准间的间接映射关系,并且充分考虑了最终用户的学习效应特点。因此所建立的模型既符合软件运行实际,又满足用户时间尺度度量评价需求,解决了上述矛盾。同时,所提供的可靠性度量结果也非常具有实用预测意义。综上,文中所建立的可靠性模型具有较强的实用性。

参考文献:

[1] ISO/IEC 9126 software engineering - product quality-part1: quality model[S]. 2002.

[2] 郭平. 软件可靠性工程中的计算智能方法[M]. 北京: 科学出版社, 2012.

[3] 陆民燕. 软件可靠性工程[M]. 北京: 国防工业出版社, 2011.

[4] 薛利兴, 左德承, 张展. 考虑系统调用和失效模式的软件可靠性模型[J]. 计算机工程与应用, 2014, 50(9): 5-11.

[5] Chiu K C. A study of software reliability growth model for time-dependent learning effects[C]//Proc of IEEE international conference on industrial engineering and management. [s. l.]: IEEE, 2012: 1015-1019.

[6] 谈维新. 考虑测试效率的软件可靠性模型[D]. 南京: 南京邮电大学, 2011.

[7] 苗扬. 软件可靠性测试与评估方法的改进[D]. 上海: 上海交通大学, 2010.

[8] 贾冀婷. 软件测试中可靠性模型的设计与研究[J]. 计算机技术与发展, 2014, 24(3): 110-112.

[9] 顾爱华, 张春阳, 胡庆新. 软件可靠性增长 G-O 模型的改进[J]. 内蒙古大学学报: 自然科学版, 2014, 45(2): 192-195.

[10] Chiu K C, Huang Y S, Lee T Z. A study of software reliability growth from the perspective of learning effects[J]. Reliability Engineering and Systems Safety, 2008, 93(10): 1410-1421.

[11] Iqbal J, Ahmad N, Quadri S M K. A software reliability growth model with two types of learning and a negligence factor[C]//Proceedings of the 2013 IEEE second international conference on image information processing. [s. l.]: IEEE, 2013.

[12] Chiu K C. A discussion of software reliability growth models with time-varying learning effects[J]. American Journal of Software Engineering and Applications, 2013, 2(3): 92-104.

[13] Cheung R C. A user-oriented software reliability model[J]. IEEE Transactions on Software Engineering, 1980, SE-6(2): 118-125.

[14] 潘冠华. 软件可靠性的功能点模型[J]. 舰船科学技术, 1997, 19(4): 56-58.

考虑时变学习效应的软件可靠性模型

作者：[赵瑞曼](#)，[潘冠华](#)，[方建勇](#)，[王蕾](#)，[ZHAO Rui-man](#)，[PAN Guan-hua](#)，[FANG Jian-yong](#)
，[WANG Lei](#)

作者单位：[江苏自动化研究所, 江苏 连云港, 222006](#)

刊名：[计算机技术与发展](#)[ISTIC](#)

英文刊名：[Computer Technology and Development](#)

年，卷(期)：[2015\(6\)](#)

引用本文格式：[赵瑞曼](#).[潘冠华](#).[方建勇](#).[王蕾](#).[ZHAO Rui-man](#).[PAN Guan-hua](#).[FANG Jian-yong](#).[WANG Lei](#) [考虑时变学习效应的软件可靠性模型](#)[期刊论文]-[计算机技术与发展](#) 2015(6)