

人工走查数据在软件可靠性度量中的应用

左万娟, 杨孟飞, 段永颢

(北京控制工程研究所, 北京 100190)

摘要: 为了充分利用第三方评测数据开展航天软件可靠性度量, 研究人工走查数据在软件可靠性度量中的应用。分析度量实体的确定、人工走查数据向动态失效数据的转换、数据融合和数据应用的合理性等问题, 提出相应解决方法。应用结果表明, 该方法具有可行性。
关键词: 航天软件; 人工走查; 第三方评测; 软件可靠性; 度量

Application of Walking Through Data in Software Reliability Measure

ZUO Wan-juan, YANG Meng-fei, DUAN Yong-hao

(Beijing Institute of Control Engineering, Beijing 100190)

【Abstract】 To conduct space software reliability measure by using the third party test data fully, this paper studies the application of walking through data in software reliability measure. It analyzes some problems such as identification of measure entity, the data transformation from walking through to dynamic failure, and the data integration and the rationality of data application. The corresponding solutions are proposed. Application results indicate that this approach is feasible.

【Key words】 space software; walking through; the third party test; software reliability; measure

1 概述

航天软件具有分阶段、多版本、不同研制单位、不同编程语言、不同规模等特点。在其研制与设计过程中, 为确保软件质量, 需要对不同研制阶段的不同版本进行多轮次第三方评测, 即独立于开发方和用户的第三方测试。

评测过程涉及人工走查数据、动态测试数据和回归测试数据。其中, 人工走查数据较多, 动态失效数据较少。因此, 通过动态失效数据驱动可靠性模型进行软件可靠性度量的传统方法不适用于航天软件, 其原因是动态失效数据太少, 难以得到稳定的度量结果。为了解决上述问题, 需要研究利用人工走查数据进行软件可靠性度量的方法。

2 航天软件第三方评测失效数据集的获取

2.1 度量实体的确定

度量实体是指一个独立的、完整的度量对象。多版本软件是指具有同一软件标识、同一软件名称、不同版本号的文件。确定度量实体是开展航天软件可靠性度量工作的前提。

通过对第三方评测报告的调研, 制定如图 1 所示的度量实体确定方法。

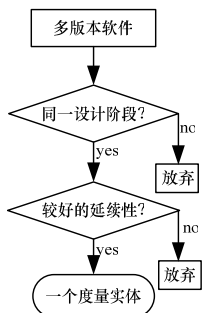


图 1 度量实体的确定方法

先考察多版本软件是否属于同一设计阶段, 只有同一设计阶段的软件才可以作为一个度量实体进行度量。然后考察多版本软件是否具有较好的延续性, 延续性的基本判定准则如下: 软件经历 1 次全面人工走查、1 次全面动态测试和若干次回归测试, 并考虑软件的更动与此前暴露问题在后续版本中的可追踪性。最后将满足上述 2 个条件的多版本软件作为一个度量实体进行度量。

2.2 人工走查数据向动态失效数据的转换

人工走查数据是指对软件进行人工走查过程中发现的问题。失效^[1]是指程序的功能在某方面没有达到用户需求, 是系统运行行为对用户要求的偏离。

2.2.1 人工走查数据转换成动态失效数据的合理性

单纯从失效定义来看, 由于在人工走查数据的获取过程中系统没有运行, 因此人工走查数据不等同于失效数据。但分析人工走查数据可以得到如下事实: 某些人工走查数据在系统运行中会以失效形式表现出来。基于该事实, 可得如下结论: 人工走查数据可以转换成动态失效数据。关于人工走查数据转换成动态失效数据的合理性, 文献[2]有相关论述。

2.2.2 人工走查数据向动态失效数据的转换规则

在制定转换规则的过程中, 一个关键标准如下: 人工走查发现的问题在程序运行中必须能以失效的形式表现出来。

分析来自第三方评测的全部问题记录, 对所提交问题的处理方式主要包括修改程序、修改文档、问题被取消、不处理、其他方式等 5 种。对程序本身的问题, 一般采取修改程序的方式进行处理。对与需求不一致的问题, 一般采取修

作者简介: 左万娟(1971—), 女, 硕士研究生, 主研方向: 软件可靠性度量; 杨孟飞, 研究员、博士生导师; 段永颢, 高级工程师、硕士
收稿日期: 2008-11-16 **E-mail:** spacecraft001@yahoo.com.cn

改程序或修改文档的方式。另外 3 种处理方式的处理对象经反复确认后, 判定其不是问题。

根据失效定义并结合上述分析, 制定转换规则如图 2 所示, 具体说明如下:

(1)剔除规则。根据问题的处理方式, 先做剔除处理, 即处理方式为修改程序、修改文档的问题初步入选失效计数。处理方式为问题被取消、不作处理和其他方式的问题不作失效计数考虑。

(2)筛选规则。对处理方式为修改程序和修改文档的问题, 按图 2 进行筛选处理。

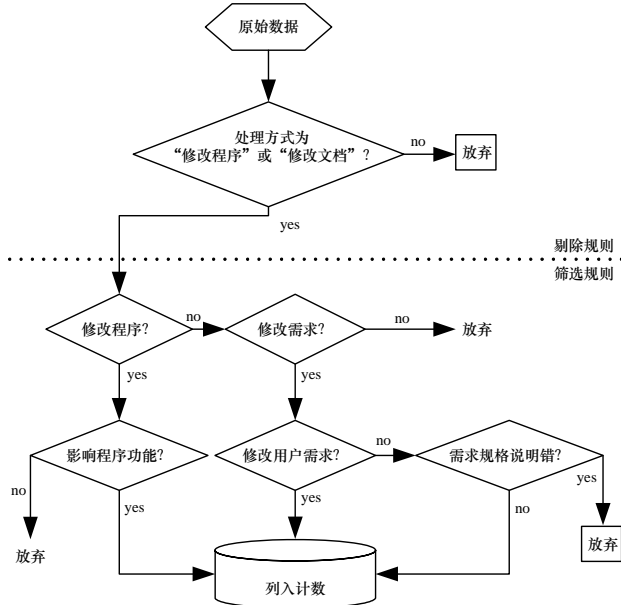


图 2 转换规则

上述剔除和筛选规则适用于动态与回归测试数据的失效计数整理。

2.3 人工走查、动态测试及回归测试数据的融合

进行数据融合的目的是为了更全面地反映软件的可靠性水平。

2.3.1 人工走查数据的进一步处理

在第三方评测期间, 人工走查数据一般经集中确认后成批提交, 由此带来了数据集中且数据点过少的问题, 不利于可靠性度量工作的开展。由于大规模软件的人工走查通常由若干人分工完成, 因此需要考虑多人人工走查数据的整合问题。

针对上述问题, 在数据整理过程中, 先对不同测试人员的走查数据按上述转换规则分别进行失效计数整理; 然后采用取随机数的方案解决集中提交问题, 即对于各人次走查数据中存在集中提交问题的数据, 分别对其在本次提交时间与上次提交时间之间取随机数, 以确定各个失效的发生时间; 最后对不同人次人工走查数据按时间顺序进行整合, 得到完整的人工走查失效数据集。

2.3.2 数据融合

采取时间规范化处理方法, 并根据回归测试是否发现问题分别采用以下 2 种算法:

(1)回归测试未发现问题的数据融合。就时间而言, 人工走查与动态测试的开展有串行和并行 2 种。在考虑人工走查与动态测试数据的融合问题时, 采取统一处理的方法来解决数据融合问题。先将人工走查工作日 T_1 与动态测试工作日

T_2 相加, 得到总工作日 $T=T_1+T_2$ 。然后对人工走查数据进行由 T_1 至 T 的规范化处理, 对动态测试数据进行由 T_2 至 T 的规范化处理。最后对上述规范化处理结果按时间顺序进行纵向相加融合处理, 对总工作日增加回归测试工作日 T_3 , 并对失效数进行累加处理。

对第(1)种算法举例说明如下:

对原始数据按上述方法进行处理后, 得到人工走查失效数据为 $(t_1, a), (t_2, b), (t_3, c)$, 动态测试失效数据为 $(t_4, d), (t_5, e), (t_6, f)$, 其中, 括号内的 2 个数分别代表提交时间和相对于上一个时间间隔内的失效数。

先对人工走查数据进行由 T_1 至 T 的规范化处理, 得

$$(t_1/T_1, a), (t_2/T_1, b), (t_3/T_1, c)$$

对动态测试数据进行由 T_2 至 T 的规范化处理, 得

$$(t_4/T_2, d), (t_5/T_2, e), (t_6/T_2, f)$$

假设 $t_1/T_1 < t_2/T_1 < t_3/T_1 < t_4/T_2 < t_5/T_2 < t_6/T_2 < T$, 对上述规范化结果按时间顺序进行纵向相加融合处理, 得到

$$(t_1/T_1, a), (t_2/T_1, b+d), (t_3/T_1, c), (t_4/T_2, d), (t_5/T_2, e), (t_6/T_2, f), (T, 0)$$

然后增加回归测试工作日 T_3 , 得到

$$(t_1/T_1, a), (t_2/T_1, b+d), (t_3/T_1, c), (t_4/T_2, d), (t_5/T_2, e), (t_6/T_2, f), (T, 0), (T+T_3, 0)$$

最后对失效数进行累加, 得到实现融合后完整的失效数据集, 即

$$(t_1/T_1, a), (t_2/T_1, a+b+d), (t_3/T_1, a+b+d+e), (t_4/T_2, a+b+d+e+c), (t_5/T_2, a+b+d+e+c+f), (T, a+b+d+e+c+f), (T+T_3, a+b+d+e+c+f)$$

(2)回归测试发现问题的数据融合。基本算法同上, 令人工走查、动态测试和发现问题的回归测试工作日分别为 T_1, T_2, T_3 , 则总工作日 $T=T_1+T_2+T_3$ 。对人工走查、动态测试和回归测试数据分别进行 T_1 至 T, T_2 至 T, T_3 至 T 的规范化处理。对规范化处理结果进行纵向相加融合处理和失效数累加处理。

2.4 第三方评测数据的整理步骤

综上所述, 得到第三方评测数据的整理步骤如下:

- (1)确定度量实体。
- (2)按日期进行失效计数整理。完成数据转换, 并按日期对人工走查、动态测试和回归测试数据分别进行失效计数整理。
- (3)按工作日进行失效计数整理。对第(2)步得到的数据按有效工作日分别进行整理。
- (4)人工走查数据的进一步处理。
- (5)数据融合, 得到完整的失效数据集。

3 数据应用

得到失效数据集后, 可以对软件开展可靠性度量。但该失效数据集与传统意义上完全来自动态测试的数据集存在很大不同, 因此, 需要进一步研究并论证以下问题: 利用该失效数据集开展软件可靠性度量是否合理; 对于此类失效数据集是否能通过数学建模方法对其开展软件可靠性度量; 应怎样使用其度量结果才能充分发挥其在工程上的实用价值。

3.1 合理性

从软件可靠性定义出发, 探讨应用本文失效数据集开展软件可靠性度量的合理性。

软件可靠性的 2 个定义如下^[3]: (1)在规定条件下、在规定时间内, 软件不引起系统失效的概率。该概率是系统输入和系统使用的函数, 也是软件中所存在缺陷的函数。系统输入将确定是否会遇到已存在的缺陷(如果有缺陷存在的话)。(2)在规定时间周期内, 在所述条件下程序执行所要求功能的能力。

第(1)个定义是定量定义,第(2)个定义是定性定义。

由上述定义可知,应用本文失效数据集开展软件可靠性度量是否合理的焦点在于该数据集的获取过程是否满足规定条件。

经典的软件可靠性增长模型均假设软件测试用例的选取代表软件实际的操作剖面,操作剖面指操作的集合及其出现的概率^[1]。规定条件即操作剖面,根据经典理论,失效数据集是依据操作剖面进行测试所得失效数据的集合。基于上述假设条件的软件可靠性度量结果应当是软件交付用户后,实际使用中的可靠性。

结合第三方评测的工程实践,将规定条件定义为软件经历全面的第三方评测。在上述规定条件下,本文开展的软件可靠性度量的结果是软件在第三方评测条件下的可靠性。

综上所述,如果对本文要开展的软件可靠性度量的规定条件及其度量结果有正确认识,则应用本文失效数据集开展航天软件可靠性度量具有合理性。

3.2 度量指标的选取

在选择度量指标时,应综合考虑工程实践条件和所选指标的普遍可得性问题。

第三方评测是以查错为目的的测试,因此,在此工程条件下,残余失效数是一个敏感项。以残余失效数作为度量指标有利于建模工作的开展,只要对累积失效数建立数学模型,即只提出均值函数假设即可,无须附加其他假设条件,增强了模型的适应性。因此,本文选择残余失效数作为度量指标。

3.3 建模

以残余失效数作为度量指标,要求数学模型在单调递增的基础上有极限,即能得到预计的总失效数。通过预计总失效数与实测失效数的差值,可以得到度量结果——残余失效数。综合考虑上述因素,建立的数学模型如下:

MGO 模型: $m(t) = a(1 - e^{-bt})$, 参考 G-O NHPP 模型^[4-5], 只取其均值函数假设。

MS 模型: $m(t) = a[1 - (1 + bt)e^{-bt}]$, 参考延迟 S-Shape 模型^[5], 只取其均值函数假设。

MW 模型: $m(t) = a[1 - \exp(-bt^c)]$, 参考 Weibull 模型^[5], 只取其均值函数假设。

ZJ 模型: $m(t) = a[1 - \exp(-bt^2/2)]$, 参考 Rayleigh 函数^[6]。其中, $m(t)$ 为累积失效数期望值(即均值函数); $a, b, c > 0$ 。

3.4 度量结果分析

应用本文数据整理方法,整理得到 8 套航天软件第三方评测失效数据集,分别对其开展可靠性度量分析。度量过程应用的参数估计方法、数值解法分别为最大似然法^[4-5]、割线法^[7],得到的拟合曲线如图 3~图 6 所示。根据拟合曲线可以看出,累积失效数的估计值与实测值拟合较好,说明建立的数学模型对本文整理得到的航天软件第三方评测失效数据集的适应性较好。

残余失效数是本文选取的度量指标,其度量结果如表 1 所示。残余失效数普遍较低,说明针对航天软件开展的多轮次第三方评测效果较好。各模型的度量结果存在一定差距,但在度量结果的相对性上基本一致,这为度量结果的横向对比提供了依据。本文对度量结果没有进行取整处理,主要是考虑用小数表示失效数能更好地反映其程度上的变化。

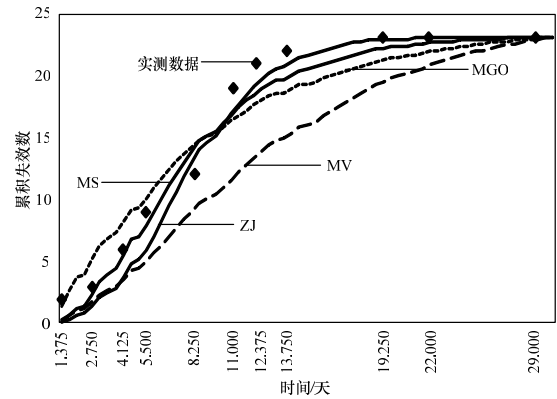


图 3 数据集 1 的拟合曲线

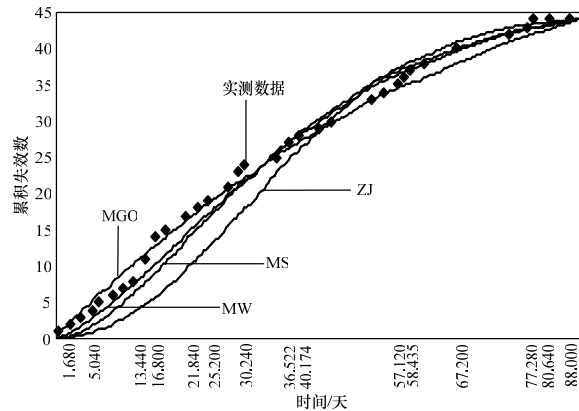


图 4 数据集 2 的拟合曲线

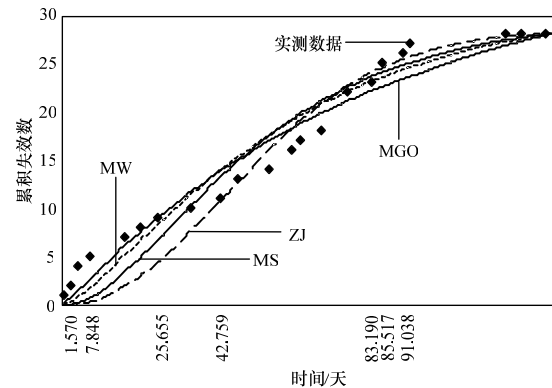


图 5 数据集 3 的拟合曲线

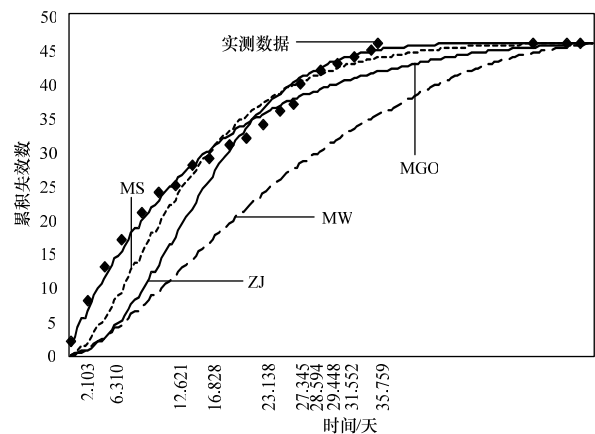


图 6 数据集 4 的拟合曲线

表1 残余失效数

| 模型 | 数据集 | | | | | | | |
|-----|---------|----------|---------|---------|----------|---------|---------|---------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| MGO | 0.672 8 | 14.153 5 | 7.237 9 | 0.764 8 | 16.021 9 | 0.757 6 | 1.124 4 | 0.545 7 |
| MS | 0.079 9 | 2.630 5 | 1.361 0 | 0.070 6 | 2.413 9 | 0.125 0 | 0.169 3 | 0.083 9 |
| MW | 1.078 5 | 2.689 3 | 2.282 0 | 1.504 2 | 1.715 8 | 0.375 9 | 1.569 1 | 0.925 1 |
| ZJ | 0.000 1 | 0.516 9 | 0.214 7 | 0.000 1 | 0.581 0 | 0.003 7 | 0.008 7 | 0.004 5 |

误差分析结果如表2~表5所示。

表2 数据集1的误差分析结果

| 误差项 | 模型 | | | |
|----------|---------|---------|----------|----------|
| | MGO | MS | MW | ZJ |
| MSE | 5.824 1 | 1.711 6 | 1.462 7 | 0.939 6 |
| RE | 0.331 1 | 0.133 5 | 0.075 7 | 0.126 6 |
| Bias | 0.493 2 | 0.074 6 | -0.044 1 | -0.206 3 |
| Variance | 2.685 0 | 1.378 8 | 1.230 0 | 1.083 5 |
| RMSPE | 2.728 0 | 1.380 8 | 1.230 8 | 1.102 9 |
| MEOP | 2.155 4 | 1.056 3 | 0.960 8 | 0.664 4 |

表3 数据集2的误差分析结果

| 误差项 | 模型 | | | |
|----------|---------|----------|----------|----------|
| | MGO | MS | MW | ZJ |
| MSE | 1.764 2 | 2.888 5 | 12.424 3 | 12.303 4 |
| RE | 0.135 3 | 0.148 0 | 0.221 7 | 0.271 9 |
| Bias | 0.264 0 | -0.323 5 | -2.787 3 | -1.766 1 |
| Variance | 1.425 8 | 1.816 5 | 6.269 4 | 4.724 1 |
| RMSPE | 1.450 1 | 1.845 1 | 6.861 1 | 5.043 4 |
| MEOP | 0.999 9 | 1.449 6 | 2.787 3 | 2.834 9 |

表4 数据集3的误差分析结果

| 误差项 | 模型 | | | |
|----------|---------|----------|---------|---------|
| | MGO | MS | MW | ZJ |
| MSE | 3.547 2 | 5.487 5 | 4.357 2 | 6.293 4 |
| RE | 0.153 9 | 0.278 2 | 0.210 1 | 0.300 6 |
| Bias | 0.064 2 | -0.062 1 | 0.262 6 | 0.698 4 |
| Variance | 1.931 1 | 2.400 2 | 2.186 6 | 2.850 6 |
| RMSPE | 1.932 1 | 2.401 0 | 2.202 4 | 2.934 9 |
| MEOP | 1.554 9 | 1.981 0 | 1.715 7 | 1.942 6 |

表5 数据集4的误差分析结果

| 误差项 | 模型 | | | |
|----------|---------|---------|----------|----------|
| | MGO | MS | MW | ZJ |
| MSE | 5.123 7 | 6.722 1 | 80.881 7 | 18.072 6 |
| RE | 0.148 3 | 0.101 6 | 0.329 5 | 0.184 7 |
| Bias | 0.655 6 | 0.336 6 | -8.165 0 | -1.097 4 |
| Variance | 2.589 3 | 2.717 2 | 17.136 2 | 4.761 5 |
| RMSPE | 2.671 1 | 2.737 9 | 18.982 0 | 4.886 3 |
| MEOP | 2.012 2 | 2.038 2 | 8.165 0 | 3.100 4 |

$$MSE = \frac{1}{n} \sum_{i=1}^n (m(t_i) - N(t_i))^2 \quad [6]$$

$$RE_i = \frac{|N(t_i) - m(t_i)|}{N(t_i)} \quad [4]$$

$$RE = \sum_{i=1}^n RE_i / n$$

$$Bias = \sum_{i=1}^n (m(t_i) - N(t_i)) / n \quad [6]$$

$$Variance = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (N(t_i) - m(t_i) - Bias)^2} \quad [6]$$

$$RMSPE = \sqrt{Variance^2 + Bias^2} \quad [6]$$

$$MEOP = \frac{1}{n} \sum_{i=1}^n |m(t_i) - N(t_i)| \quad [6]$$

其中, t_i 是工作日(天); $N(t_i)$ 是实际累积失效数; $m(t_i)$ 是估计累积失效数; n 是数据集样本大小。

将误差分析结果与文献[6]结果进行比对可以发现,应用MGO, MS, MW及ZJ模型对本文失效数据集开展度量得到的各项误差较小。

3.5 经典模型的工程扩展性应用方法

由于模型的假设极大限制了模型的使用和适用性,因此本文通过研究,提出经典模型的工程扩展性应用方法如下:在工程应用上,只保留其数学模型部分(即均值函数假设),不使用模型原型的其他假设,根据工程实践需要对规定条件给出具体定义,通过对模型的综合分析选择最佳模型并提取度量结果。

4 结束语

本文提出以下方法:(1)以人工走查数据为主的航天软件第三方评测失效数据集的获取方法;(2)度量指标的选取、建模和经典模型的工程扩展性应用方法。

应用上述方法可以形成统一的航天软件第三方评测数据整理机制和可靠性度量机制。以上述统一机制为基础,可以将度量结果纳入软件质量及其可靠性属性的量化管理过程。数据应用结果表明,本文提出的人工走查数据在航天软件可靠性度量中的应用方法是可行的,且效果良好,为充分利用第三方评测数据开展航天软件可靠性度量奠定了基础。

参考文献

- [1] Musa J D. 软件可靠性工程[M]. 韩柯,译. 北京:机械工业出版社,2003.
- [2] Musa J D, Iannino A, Okumoto K. 软件可靠性——度量、预计和应用[M]. 姚一平,译. 北京:机械工业出版社,1992.
- [3] 中华人民共和国国家技术监督局. GB/T11457-1995 软件工程术语[S]. 1995.
- [4] 蔡开元. 软件可靠性工程基础[M]. 北京:清华大学出版社,1995.
- [5] Lyu M R. Handbook of Software Reliability Engineering[M]. New York, USA: McGraw-Hill, 1996.
- [6] Lin Chuti, Huang Chinyu, Chang Junru. Integrating Generalized Weibull-type Testing-effort Function and Multiple Change-points into Software Reliability Growth Models[C]//Proc. of APSEC'05. Taiwan, China: [s. n.], 2005: 431-438.
- [7] 颜庆津. 数值分析[M]. 北京:航空航天大学出版社,1999.

编辑 陈晖