# 一种基于 JM 模型的软件安全性测试方法研究

李 娟,陈 斌
(海军工程大学,湖北 武汉 430033)

**摘 要**:为保证和提高软件安全性水平,针对软件安全性与软件失效后果和发生可能性密切相关,提出一种有效的软件安全性测试方法。在 JM 模型中注入软件失效严重度参数,对软件失效后果严重度进行降级处理;根据软件失效对系统安全影响程度,推导了软件安全可靠度计算公式;以软件错误严重度和发生概率为核心,建立了软件风险计算公式,更直观地反映软件安全性能,其中所定义的含权软件缺陷严重度变化矩阵,直接反映软件安全改善力度。改进后的 JM 模型以降低软件风险为目的开展测试过程,更符合软件安全性特征,为软件安全性测试的工程实践提供了一种可行、可信的方法。

**关键词**:JM 模型; 软件安全性; 测试;软件失效
**中图分类号**:TP309 **文献标识码**:A **文章编号**:1673-629X(2012)09-0246-04

# Study on JM Model Based Software Safety Test

LI Juan, CHEN Bin
(Naval University of Engineering, Wuhan 430033, China)

**Abstract**: In order to ensure and enhance the level of software safety, according to close relation of effects and likelihood software failures to software safety, an effective method is propounded for software safety test. Being engaged in the JM model, severity of software failure will degrade through testing process. Based on degree of software failure affecting on system safety, the formula for calculating software safety reliability is deduced. Further, risk formula is built focusing on severity and probability of software error. It is more intuitive to respond the performance of software safety. Weighting severity changed matrices is defined to express the improvement directly. With the improved JM model software safety test is processed for the purpose of risk reducing, which is more fit to character of software safety. It provides a feasible and credible method for practice of software safety test.

**Key words**: JM model; software safety; test; software failure

## 0 引 言

自上个世纪中期以来,系统安全性(System Safety)研究已经发展成为系统工程领域的一门独立学科。随着软件在安全关键领域的广泛应用,人们对软件安全性(Software Safety)[1,2]展开了深入的研究。软件安全性的出发点及目标点是系统安全性。其中,软件安全测评则是保证软件安全性的重要方面,也是软件安全性研究的重要课题之一。

软件安全性是指“软件运行不引起系统事故的能力”(GJB/Z 102-97)[3]。虽然安全性和可靠性[4]之间存在着一定的共性,安全性需要可靠性的支持,在实践中,人们往往将两者混为一谈,采用软件可靠性方法和技术实施软件安全性测试。但是,软件安全性与软件可靠性二者在具体含义及关注目标等方面表现出不同的特征,不存在等价关系。安全性描述的是系统在规定时间内、在特定环境下,不会发生事故或造成灾难性事故的能力。它更关注于导致系统进入不安全状态的、风险较大的软件缺陷。其中,事故(Accident)是造成人身安全伤害、重大财产损失、系统外界危害、关键任务失败的意外事件。

对软件可靠性模型(论文选用 JM 模型)及测试过程进行适当的修改,通过软件安全可靠性和软件风险指标计算,可以为软件安全性测试提供更为合适的方法和有力依据。

## 1 软件安全性影响因素

在系统这个层面上,软件被视为系统的组成部分,软件的失效有可能成为系统故障的一个诱发条件,这个诱发条件最终可能会导致系统事故的发生。软件失效[5]是指软件不能在既定约束条件下执行期望的功能。而软件失效的根本原因可以归结到软件的不正确性,即软件存在缺陷或错误,而且没有方法可以保证软件不存在缺陷。可能导致事故的状态称之为危险

**收稿日期**:2011-12-28;**修回日期**:2012-03-30
**基金项目**:海军工程大学青年基金(HGDQNJJ11026)
**作者简介**:李 娟(1977-),女,博士生,CFF 会员,副教授,研究方向为计算机应用技术。

(Hazard),每一个软件失效都是一种潜在的危险,可能导致系统事故的发生。

用集合 $F$ 来表示系统中的软件错误:

$F = \{f_1, f_2, \cdots, f_i, \cdots, f_N\}$

$f_i$ 对应系统中某一个软件缺陷或错误, $N$ 表示软件中所有缺陷(错误)总数。

GJB900《系统安全性通用大纲》把软件事故分为四个等级:灾难性的(人员死亡或系统报废)、严重的(人员严重受伤、严重职业病或系统严重损坏)、中等的(人员轻度受伤、轻度职业病或系统轻度损坏)、轻微的(人员受伤和系统损坏次于中等量级)[6]。再考虑到有些软件失效不可能产生事故,可以将软件失效造成的事故后果分为 5 个等级。

因此,对于每一个软件缺陷 $f_i$ ,将其可能引起事故的严重程度定义为危险严重性(Severity)。危险严重性级别从高到低分别定义为 S4(灾难),S3(严重),S2(中等),S1(轻微)和 S0(无危险)。每一个软件缺陷 $f_i$ 所对应的危险可定义为:

$h_i \in H = \{S0, S1, S2, S3, S4\}$

除了危险的严重性外,各种危险的发生概率也是影响安全性的一个重要因素。产生某种危险总的可能性,定义为:

$$p(H) = \{p_{S0}, p_{S1}, p_{S2}, p_{S3}, p_{S4} \mid \sum_{i=0}^{4} p_{Si} = 1\}$$

## 2　软件安全可靠性测试

### 2.1　JM 模型局限性

JM(Jelinski-Moranda)模型是 Z. Jelinski 和 P. B. Moranda 于 1972 年提出的软件可靠性模型[7,8],是最具有代表性的马尔可夫过程模型。

该模型是提出最早且应用较为广泛的可靠性模型,测试过程中,软件的运行方式与预期的运行方式相似,系统基本假设如下:

(1)开始测试时(前),软件的初始失效数,即软件中固有的错误数 $N$ 是一个未知但固定的常数;

(2)软件中的所有错误(失效)相互独立,以相同概率发生,失效严重程度相同;

(3)软件错误(失效)一旦被发现后立即被完全排除,且不引入新的错误,排除错误的时间忽略不计。即每次排除软件错误后, $N = N - 1$;

(4)软件失效率在错误(失效)发生的时间间隔内保持不变,其数值与软件中残余的错误(失效)数成正比,用正比例常量 $\varphi$ 表示。则,在第 $i$ 个测试区间,即发生第 $i-1$ 个软件错误(失效)到发生第 $i$ 个软件错误(失效)之间,失效率函数为:

$$\lambda_i = \varphi(N - i + 1) \tag{1}$$

其可靠度函数为:

$$R_i(t) = \exp[-\varphi(N - i + 1)t] \tag{2}$$

关于 $t_i$ (发生第 $i$ 次软件错误的时间)的概率密度函数为:

$$p(t_i) = \varphi(N - i + 1)\exp[-\varphi(N - i + 1)\, t_i] \tag{3}$$

可靠性模型关注于软件失效对系统运行能力的影响[9],这和软件安全性所关注的问题是一致的。安全性作为可靠性的一个子集,首先要保证系统的运行能力。

但是,可靠性强调的是持续服务,系统的所有失效是等价的,不考虑其危害性。而安全性的着重点在于系统风险,这与软件失效的严重性密切相关,在 JM 模型中却没有任何体现。同时,软件错误发现后立即排除,只是一种理想状态,由于开销及系统本身原因,有时是做不到的[10,11]。在这种情况下,设法降低软件失效的严重性也是改善软件安全性的一种有效措施。由于软件失效所产生后果是不一样的,所以对软件错误的处理也不应该一概而论。

### 2.2　JM 模型改进

针对 JM 模型应用于软件安全性方面的局限性,对模型的假设作如下更改:

(1)软件错误被发现后立即可以对其严重性进行评判;

(2)在错误处理中不引入新的错误,错误处理的时间忽略不计。对错误的处理,可以分为 3 种情况:

a. 完全排除一个错误;

b. 错误对系统安全无影响,不排除;

c. 无法完全排除错误,只能降低错误的严重性,将改软件失效严重性由 $h_i$ 降为 $h_i'$ 。

按改进后的模型,对于每一次软件失效 $f_i$ 处理,应该记录以下数据:

$t_i$ ,软件失效发生时间;

$h_i$ ,软件失效严重度;

$e_i$ ,错误是否完全排除, $e_i \in \{0,1\}$ (0:未完全排除,1:完全排除);

$h_i'$ ,错误处理(降低错误的严重性)后,软件失效严重度。

### 2.3　测试公式

在第 $i$ 个测试区间(即发生第 $i-1$ 个软件错误到发生第 $i$ 个软件错误之间),已排除软件错误个数为: $\sum_{j=1}^{i-1} e_j$ ;

其失效率函数为:

$$\lambda_i = \varphi(N - \sum_{j=1}^{i-1} e_j) \tag{4}$$

其可靠度函数为：

$$R_i(t) = \exp[-\varphi(N - \sum_{j=1}^{i-1} e_j)t] \tag{5}$$

关于 $t_i$（发生第 $i$ 次软件错误的时间）的概率密度函数为：

$$p(t_i) = \varphi(N - \sum_{j=1}^{i-1} e_j)\exp[-\varphi(N - \sum_{j=1}^{i-1} e_j)\, t_i]$$

模型中含有两个未知参数 $N$ 和 $\varphi$，这两个参数可以通过最大似然法求出相应的估计值。由以下超越方程：

$$\sum_{i=1}^{n} \frac{1}{\hat{N} - \sum_{j=1}^{j=i-1} e_i} = \frac{n}{\hat{N} - \frac{1}{\sum_{i=1}^{n} t_i}[\sum_{i=1}^{n}(\sum_{j=1}^{i-1} e_j)\, t_i]}$$

可以求得 $N$ 的估计值 $\hat{N}$。

求得 $\hat{N}$ 后，代入下面方程：

$$\hat{\varphi} = \frac{n}{\hat{N}(\sum_{i=1}^{n} t_i) - \sum_{i=1}^{n}(\sum_{j=1}^{i-1} e_j)\, t_i}$$

可以求得 $\varphi$ 的估计值 $\hat{\varphi}$。

### 2.4 软件安全可靠度(Software Safety Reliability)

类似于软件可靠度定义，可以定义软件安全可靠度 $RS$ 为在规定条件下和规定时间内，不发生大于等于危险级别为 $Sk(k > 0)$ 事故的概率。根据软件安全可靠度定义，需要重新整理测试数据，仅需关注测试过程中发现的严重度大于等于 $Sk$ 的软件错误，总数为 $M$，记为：$Fs = \{fs_1, fs_2, \cdots, fs_i, \cdots, fs_M \mid fa_i \in F, h_i \geqslant Sk\}$；

并记录相关数据，包括：

$t_i'$，影响系统安全的软件失效发生时间；

$e_i'$，危险是否完全排除，$e_i' \in \{0,1\}$（1：危险排除或降级到安全范围，0：危险未被降级到安全范围）。

依据式 5 可得：

安全可靠度函数为：

$$RS_i(t) = \exp[-\varphi'(M - \sum_{j=1}^{i-1} e_j')t] \tag{6}$$

其中，$\varphi'$ 为比例常数。关于 $t_i'$（发生第 $i$ 次严重度大于等于 $Sk$ 的软件错误的时间）的概率密度函数为：

$$p(t_i') = \varphi'(M - \sum_{j=1}^{i-1} e_i')\exp[-\varphi'(M - \sum_{j=1}^{i-1} e_i')\, t_i'] \tag{7}$$

## 3 软件风险

用危险可能性和危险严重性表示的事故损失可能程度叫做风险(Risk)。即：

$$\text{Risk} = N\sum_{i=1}^{n} w(h_i)\, p_i \tag{8}$$

其中，$N$ 为系统中软件错误(失效)总数，可以由前述改进的 JM 模型进行估算。$w(h_i)$ 表示严重度 $h_i$ 所对应的风险权值，最小值为 $w(S0) = 0$，表示该软件错误的发生不会(或极小可能，可忽略)引发系统危险，不存在事故风险。$w(h_i)$ 的不同取值，代表了软件错误(失效)可能引发的事故损失的不同等级，具体数值视系统具体需求而定，目的是要将事故后果的严重程度及代价划分不同等级和档次。$p_i$ 代表某一个软件错误出现概率，根据 JM 模型假设，系统中 $N$ 个错误的出现概率是相等的，即 $p_i = \frac{1}{N}$。

软件风险评价中，所关注的是软件错误(失效)可能引发事故的后果或代价，即事故严重程度，而并不是事故本身。所以，式 8 可用以下等价形式表示：

$$\text{Risk} = N\sum_{i=1}^{4} w(Si)\, p_{si} \tag{9}$$

而软件测试过程中，由于有些失效已消除，软件剩余错误为 $N - \sum_{i=1}^{n} e_i$，有些失效的严重度得到降级，故失效严重度分布的概率函数会发生变化。测试过程中失效严重度变化可以表示为：

$$\boldsymbol{D} = \begin{bmatrix} d_{0,0} & d_{0,1} & d_{0,2} & d_{0,3} & d_{0,4} \\ d_{1,0} & d_{1,1} & d_{1,2} & d_{1,3} & d_{1,4} \\ d_{2,0} & d_{2,1} & d_{2,2} & d_{2,3} & d_{2,4} \\ d_{3,0} & d_{3,1} & d_{3,2} & d_{3,3} & d_{3,4} \\ d_{4,0} & d_{4,1} & d_{4,2} & d_{4,3} & d_{4,4} \end{bmatrix}$$

其中，

$$d_{i,j} = \begin{cases} -\text{Count}(f_k \mid (h_k = Si) \bullet (h_k' = Sj)), i < j \\ \text{Count}(f_k \mid (h_k = Sj) \bullet (h_k' = Si)), i > j \\ \sum e_i \bullet (h_k' = Si), i = j \end{cases}$$

矩阵下三角元素 $d_{i,j}(i > j)$ 为负数，其绝对值为从严重度 $Si$ 降级为严重度 $Sj$ 的软件失效个数；上三角元素 $d_{i,j}(i < j)$ 为正数，与上三角反对称，值为从严重度 $Sj$ 降级为严重度 $Si$ 的软件失效个数；对角线元素 $d_{i,j}(i = j)$ 为负数，其绝对值表示被完全排除的、严重度为 $Si$ 的软件错误数。以 $c_i$ 严重度为 $Si$ 类型的软件失效增量，那么表示 $c_i$ 即为矩阵第 $i(i \in \{0,1,2,3,4\})$ 行元素和，如下式所示：

$$\Delta c_i = \sum_{j=0}^{4} d_{i,j} \begin{cases} > 0, \text{严重度为 } Si \text{ 的软件缺陷个数有增加}; \\ < 0, \text{严重度为 } Si \text{ 的软件缺陷个数有减少}; \\ = 0, \text{严重度为 } Si \text{ 的软件缺陷个数无增减}; \end{cases}$$

测试过程中，会对软件失效进行处理，排除软件失效或降级严重度，故：$i > j \Rightarrow \Delta c_i < \Delta c_j$。

测试后的软件失效严重度的概率为：

$$p_{Si}' = \frac{N \times p_{Si} + \sum_{j=0}^{4} d_{i,j}}{N + \sum_{j=0}^{4} d_{j,j}} \tag{10}$$

而测试后的软件失效为:

$$\mathrm{Risk}' = \sum_{i=1}^{4} w(Si)(N \times p_{Si} + \sum_{j=0}^{4} d_{i,j}) \tag{11}$$

经过软件安全性测试,软件安全性改善可用风险降低值来表示,为:

$$\Delta\mathrm{Risk} = \sum_{i=1}^{4}(w(Si)\sum_{j=0}^{4} d_{i,j}) = \sum_{i=1}^{4}\sum_{j=0}^{4} w(Si)d_{i,j} \tag{12}$$

定义带权值的软件缺陷严重度变化矩阵为:

$$\boldsymbol{DW} = \begin{pmatrix} w(S0) & 0 & 0 & 0 & 0 \\ 0 & w(S1) & 0 & 0 & 0 \\ 0 & 0 & w(S2) & 0 & 0 \\ 0 & 0 & 0 & w(S3) & 0 \\ 0 & 0 & 0 & 0 & w(S4) \end{pmatrix} \times \boldsymbol{D} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -w(S1)d_{0,1} & w(S1)d_{1,1} & w(S1)d_{1,2} & w(S1)d_{1,3} & w(S1)d_{1,4} \\ -w(S2)d_{0,2} & -w(S2)d_{1,2} & w(S2)d_{2,2} & w(S2)d_{2,3} & w(S2)d_{2,4} \\ -w(S3)d_{0,3} & -w(S3)d_{1,3} & -w(S3)d_{2,3} & w(S3)d_{3,3} & w(S3)d_{3,4} \\ -w(S4)d_{0,4} & -w(S4)d_{1,4} & -w(S4)d_{2,4} & -w(S4)d_{3,4} & w(S4)d_{4,4} \end{pmatrix}$$

则,软件安全性改善值 ΔRisk 为矩阵 ***DW*** 所有元素和的绝对值。

## 4　结束语

将 JM 模型用于软件安全性测评,对其作如下改进:注入软件失效严重度参数;测试过程中,增加了对已发现的软件错误(失效)进行严重度降级的处理方式。改进后的 JM 模型更符合软件安全性的定义,并易于工程实践。在改进的 JM 模型基础上所计算的软件安全可靠度,是专门针对软件安全性的一个度量指标,该指标体现了软件安全的可能性。

利用 JM 模型中估算的软件错误总数,及相关测试数据,可以计算软件风险值。软件风险定义不但刻画了导致事故的危险发生的可能性(概率),还考虑了不同的事故代价(事故后果的严重程度)。这与软件安全性增长测试的目的,减少软件事故风险是相一致的。该项指标更为直观地反映软件的安全性。软件安全性设计和软件安全性测试的最终目标便是减少因为软件因素而导致灾难性事故的风险[12]。带权值的软件缺陷严重度变化矩阵 ***DW*** 直接反映了系统安全性测试的能力,即通过测试软件安全是否得到改善以及改善力度。

**参考文献:**


[1] NASA-GB-8719.13B. Software Safety[S]. Washington: NASA,2004.

[2] 樊晓光,褚文奎,张凤鸣. 软件安全性研究综述[J]. 计算机科学,2011(5):14-19.

[3] GJB/Z 102-97. 软件可靠性和安全性设计准则[S]. 北京:国家技术工业委员会,1997.

[4] 李烈彪,李　仙. 计算机系统的可靠性技术[J]. 计算机技术与发展,2007,17(11):148-150.

[5] 胡海宏,沈元隆. 基于用户要求并考虑软件失效的费用模型[J]. 计算机技术与发展,2011,21(7):91-95.

[6] GJB-900-90. 系统安全性通用大纲[S]. 北京:国家技术工业委员会,1991.

[7] Jelinski Z, Moranda P B. Software reliability research[M]// Statistical computer performance evaluation. New York: Academic Press,1972:465-484.

[8] John D M. Software Reliability Engineering[M]. 北京:机械工业出版社,2003.

[9] 楼俊钢,江建慧,帅春燕,等. 软件可靠性模型研究进展[J]. 计算机科学,2010(9):19-25.

[10] 谈维新,沈元隆. 考虑测试效率的软件可靠性模型研究[J]. 计算机技术与发展,2011,21(8):73-76.

[11] 郑　垒,沈元隆. 考虑非理想排错过程的软件可靠性模型[J]. 计算机技术与发展,2011,21(8):118-122.

[12] Ericson C A. Hazard analysis technique for system safety[M]. Hoboken(USA): John Wiley & Sons Inc,2005.


(上接第 245 页)


档 4.0.1 [EB/OL]. [2007-07-23]. http://www.WinPcap.org.cn/.

[7] 徐培文. 软交换与 SIP 实用技术[M]. 北京:机械工业出版社,2007.

[8] Sisalem D, Kuthan J, Ehlert S. Denial of sevice attacks targeting a SIP VOIP infrastructure: attack scenavius and prevention mechanisms[J]. Network IEEE, 2006,20(5):26-31.

[9] 吴坤鸿,乐宏彦. 反 rootkit 的内核完整性检测与恢复技术[J]. 计算机工程,2008(21):129-131.

[10] 胡　影,郑康锋,杨义先. 一种基于原子功能的网络攻击效果评估指标体系[J]. 计算机工程与科学,2008,30(10):1-4.

[11] Zhang Lijuan, Cao Yan, Wang Qingxian. Fuzzy-AHP 法在网络攻击效果评估中的应用[J]. 北京邮电大学学报,2006,29(1):124-127.

[12] Choudhary A R. Security-auditing in a Softswitch[C]//Information Assurance Workshop, 2003. [s. l.]: IEEE Systems, Man and Cybernetics Society,2003:292-293.

# 一种基于JM模型的软件安全性测试方法研究



作者：李娟， 陈斌
作者单位：海军工程大学,湖北 武汉 430033
刊名：计算机技术与发展
英文刊名：Computer Technology and Development
年，卷(期)：2012(9)