云南大学学报（自然科学版），2012，34（2）：159～164　　CN 53－1045/N　ISSN 0258－7971
Journal of Yunnan University　　http://www.yndxxb.ynu.edu.cn

基于信息熵的软件维护风险度量*[1]

王　佳，周　华，梁志宏，代　飞，白丽瑞
（云南大学 软件学院，云南省软件工程重点实验室，云南 昆明 650091）

摘要：针对软件维护过程中不确定信息难以量化的问题，使用信息熵定量度量软件的维护风险. 基于信息熵，引入信息熵定量分析算法，提出了软件维护风险模型，使用信息熵算法定量计算软件维护过程中的不确定程度和损失度. 仿真结果表明，基于软件维护风险模型，使用信息熵算法能够定量度量软件的维护风险.

关键词：软件维护风险；信息熵；维护风险模型；层次分析；定量度量
中图分类号：TP 311　　**文献标识码**：A　　**文章编号**：0258－7971(2012)02－0159－06

软件项目成功完成开发以后，在用户投入使用过程中，随着时间的推移，常常需要对软件作一些变更. 通常把软件在运行/维护阶段对软件产品所进行的修改叫做维护[1]. 有效地进行软件维护是非常重要的. 软件项目不同于一般的工程项目，软件项目它是一种概念项目，是一种逻辑产品，因此对软件项目的维护就比一般的工程项目维护更复杂. 通常选择软件项目的维护风险作为度量指导软件维护工作的一个重要参考依据. 显然，修改一个维护风险较高的软件项目比修改一个维护风险较低的软件项目更容易，而且可以降低维护开支，提高软件项目维护成功的可能性. 因此预测软件项目的维护风险的度量模型对软件企业具有重要的意义.

目前，有基于程序信息的软件可维护性度量[2]，定义软件系统的可维护性是软件开发阶段各个时期的关键目标，进行软件维护时需要大量统计程序代码的信息，工作量大而且费时；基于支持向量机的面向对象软件可维护性预测[3]，主要是针对面向对象的软件进行预测，使用的范围有局限性；基于 SVM（support vector machine，支持向量机）的软件可维护性评价模型研究[4]是一种新的基于统计学习理论的机器学习算法，需要大量的数据进行统计分析；基于 MIM（Measurement Information Model，度量信息模型）的软件度量扩展模型对软件可维护性的度量[5]主要使用目标驱动问题，再去解决具体问题，在维护阶段有的问题是没有答案的；基于层次分析法的软件可维护性评价[6]针对软件可维护性定量研究工作较少的情况，提出了利用层次分析法对其进行了定性和定量评估的方法，但是没有度量单元的共享和数据的共享.

本文依据 ISO/IEC 25010—2011 标准[7]中度量的方法和标准，设计了分析基础数据信息的统计表，使用系统科学中的信息熵对软件系统维护的不确定性和软件系统的维护风险进行度量，提出了软件项目维护风险模型，通过实践验证，该模型能有效度量软件维护风险.

1　维护风险的定义与信息熵

1.1　维护风险的定义　软件项目的维护风险就是项目维护过程中所出现的不确定程度. 不确定性程度越高，项目维护失败的可能性越大，给企业带来的损失也就越大. 因此，在软件项目的维护过程中，必然涉及不确定信息和维护损失.

定义 1　维护风险包括软件维护因素、软件维护因素发生的概率和软件维护因素产生的损失度，所以

* **收稿日期**：2011－10－19
基金项目：云南省软件工程重点实验室开放基金资助项目(2011SE04).
作者简介：王　佳(1985－)，男，四川人，硕士生，主要从事软件体系结构方面的研究.
通讯作者：周　华(1963－)，男，云南人，研究员，主要从事软件体系结构及企业应用集成方面的研究. E－mail：hzhou@ynu.edu.cn.

把维护风险表示为一个三元组 $X=(M,P,C)$,其中 M 表示软件维护因素的集合,P 表示软件维护因素发生的概率,C 表示软件维护因素产生的损失度.

1.2 信息熵定量分析算法 香农的狭义信息论最先定义信息量是人们对事物了解的不确定性的消除或减少. 美国科学家冯·诺依曼把香农定义的信息量称作熵. 信息熵就是在通信前后消除的信息不确定性[8]. 参考文献[9-10]对信息熵定义.

定义2 如果设从某个信息 X 中得知的可能结果为 $x_i,i=1,2,\cdots,n$,记 $X=\{x_1,x_2,\cdots,x_n\}$,而每种结果 x_i 出现的概率分别为 $P_i,i=1,2,\cdots,n$,则信息熵 $H(X)=-k\sum_{i=1}^{n}p_i\cdot\log_2P_i$,其中,$0\leqslant P_i\leqslant 1(i=1,2,\cdots,n)$,且有 $\sum^{n=i=1}P_i=1$,k 为比例系数. 若概率系统为连续系统,其概率分布为 $P(x)$,则信息熵由 $H=-\int_{x_0}^{x_1}P(x)\log_2P(x)\mathrm{d}x$,其中 $x\in[x_0,x_1]$ 所表示.

最大信息熵原理从理论上说明,在信息熵取得极大值时,对应的一组信息状态出现的概率占有绝对优势,这就为维护风险的熵权系数提供了数理依据.

信息熵 $H(X)=-\sum_{i=1}^{n}p_i\cdot\log_2P_i$ 表示系统信息的有序程度. 对于某种信息结果 x_k 有 $x_k=1$,那么其他结果 x_i 的 $P_i=0(i\neq 0)$,则 $H(X)=0$ 取得最小值. 如果 $X=\{x_1,x_2,\cdots,x_n\}$,$P_i=\frac{1}{n}$,则 $H(X)=\log_2n$ 取得最大值. 所以,$0\leqslant H(X)\leqslant\log_2n$. P_i 越接近相等,熵值越大,影响系统维护的因素 M_i 产生的不确定性就越大. 因此,可用信息熵计算各系统维护因素的权值.

$P_{i,j}$ 为第 i 个基本特征要项下的第 j 项子项目标的不确定程度,$P_{i,1},P_{i,2},\cdots,P_{i,n};0\leqslant P_{i,j}\leqslant 1$. $c_{i,j}$ 为第 i 个基本特征要项下的第 j 项子项目标的损失度,$c_{i,1},c_{i,2},\cdots,c_{i,n};0\leqslant c_{i,j}\leqslant 1$. 可维护量化值 $m_{i,j}$ 由软件项目的不确定程度 $P_{i,j}$ 和损失度 $c_{i,j}$ 两个重要的特性决定. 可维护量化值 $m_{i,j}$ 为:

$$m_{i,j}=\sqrt{P_{i,j}c_{i,j}}\quad i=1,2,\cdots,n;j=1,2,\cdots,n. \tag{1}$$

对项目维护的因素 M_i 的相对重要性进行归一化处理,信息熵表示为:

$$e_i=-\frac{1}{\log_2n}\sum_{j=1}^{n}m_{i,j}\cdot\log_2m_{i,j}. \tag{2}$$

当 $m_{i,j}(j=1,2,\cdots,n)$ 取值相等时,熵值 e_i 最大为1. 所以 e_i 的值也满足 $0\leqslant e_i\leqslant 1$. 熵值取得最大时,项目的维护因素对项目维护风险贡献最小,所以确定项目维护因素 M_i 的权值由 $1-e_i$ 度量. 因此,对其进行归一化处理得到项目维护因素 M_i 的权值 w_i 为:

$$w_i=\frac{1-e_i}{\sum_{i=1}^{n}(1-e_i)}. \tag{3}$$

式中 $0\leqslant w_i\leqslant 1$,且有 $\sum_{i=1}^{n}w_i=1$.

2 建立软件项目维护风险模型

2.1 建立维护风险的层次图 依据 ISO/IEC 25010-2011 标准[7]做适当的扩展,建立维护风险的层次模型,如图1所示. 软件项目维护风险为第1层(用 α 表示). 维护风险包括测试风险、理解风险、修改风险和复审风险4种基本特性[11]. 这4种基本特性为第2层(用 β 表示),每一类基本特性又和若干个维护因素指标相互作用,维护因素指标为第3层(用 γ 表示). 测试风险主要由可访问性、可通信性、结构化和自描述性等特性决定;理解风险主要由结构化、自描述性、简洁性、合法性、可扩展性和一致性等特性决定;修改风险主要由结构化、可扩展性和环境无关性等特性决定;复审风险主要由自描述性、简洁性、维护计划可复审性和维护日志可复审性等特性决定. 通过对这些子特性进行度量,可以量化软件项目维护风险.

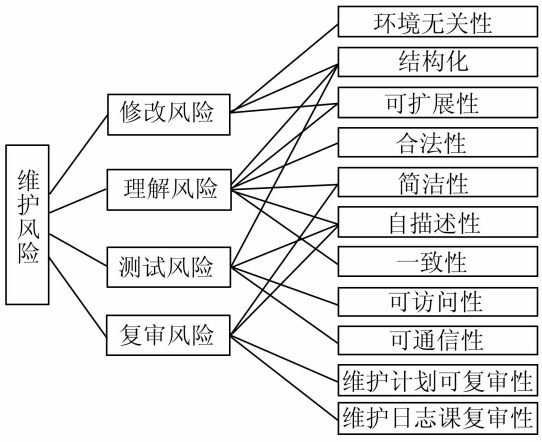

图 1　系统维护风险层次图

Fig. 1 The level diagram of system maintenance risk

将项目总体维护风险作为顶级事件,通过分析造成顶级事件的各种可能原因及彼此间的关系,构造逻辑层次图. 再由分析结果确定软件项目的关键部位、薄弱环节、对安全性的要求等. 最终评估软件项目的总体维护风险.

2.2　熵权系数及符合度说明　在系统维护风险层次图的基础上,跟据 ISO/IEC 25010—2011 标准的扩展,首先对每层影响项目维护的因素 M_i 的熵权系数进行规定. 层次图的第 1 层有 4 个基本特征子项,体现了目标项目总体维护风险包含的若干基本特征维护要素,其熵权数值用 α_i 表示;层次图的第 2 层说明每个基本特征维护要素要通过若干个具体子项,第 i 个基本特征要项下的第 j 项子项目标的熵权系数用 $m_{i,j}$ 表示. $m_{i,j}$ 由软件项目维护包含过程中的不确定程度 $P_{i,j}$ 和损失度 $c_{i,j}$ 两个重要的特性决定. 把符合度的量化定义划分成5 个级别,具体的划分如表 1 和表 2 所示.

在进行软件项目实际维护时,对软件系统的不确定程度和损失度 2 个重要的特性通过相关的专家问卷调查等方式进行收集整理,比较不确定程度和损失度这 2 个重要性质的现有程度与维护基线的差距,从而确定出符合度的量化值. 于是,这个符合度能客观地反映软件项目的维护风险的现状.

2.3　使用信息熵算法计算维护风险　对软件项目总体维护风险的量化具体计算过程如下.

第 1 步　取得符合度量化值. 假设在层次图中第 i 个基本特征要项下的第 j 项子项目标的熵权数值用 $m_{i,j}$ 表示,$m_{i,j}$ 由软件项目维护包含不确定程度 $P_{i,j}$ 和损失度 $c_{i,j}$ 两个重要的特性决定($P_{i,j}$ 和 $c_{i,j}$ 来源软件项目实践考核的评分,图 1).

表 1　维护不确定程度定义

Tab. 1 The definition of the maintenance the uncertainty conformity

符合度	量化值	符合度描述
$P_{i,j,1}$	0	系统维护的不确定程度能很好的控制,维护的风险非常小
$P_{i,j,2}$	0. 25	系统维护的不确定程度能比较好的控制,维护的风险较小
$P_{i,j,3}$	0. 5	系统维护的不确定程度能较好的控制,维护风险一般
$P_{i,j,4}$	0. 75	系统维护的不确定程度能一部分的控制,维护风险一般大
$P_{i,j,5}$	1	系统维护的不确定程度不能控制,维护风险最大

表 2　维护损失度定义

Tab. 2 The definition of the maintenance the loss degree conformity

符合度	量化值	符合度描述
$c_{i,j,1}$	0	系统维护的资金能很好的控制,损失非常小
$c_{i,j,2}$	0. 25	系统维护的资金能比较好的控制,损失较小
$c_{i,j,2}$	0. 5	系统维护的资金能较好的控制,损失一般
$c_{i,j,3}$	0. 75	系统维护的资金能一部分的控制,损失一般大
$c_{i,j,4}$	1	系统维护的资金能不能控制,损失最大

第 2 步 首先根据公式(1),(2)和(3),计算第 2 层各个子项目标的熵权系数得系数 $m_{i,j}$ 和维护风险权值 w_i,则计算公式分别为:

不确定程度 $P_{i,j}$ 和损失度 $c_{i,j}$ 两个目标统一:$m_{i,j} = \sqrt{P_{i,j}c_{i,j}}$, (4)

进行归一化处理:$\beta_i = -\frac{1}{\log_2 n}\sum_{j=1}^{n} m_{i,j} \cdot \log_2 m_{i,j}$, (5)

平均一致性指标:$w_i = \frac{1-\beta_i}{\sum_{i=1}^{n}(1-\beta_i)}, 0 \leqslant w_i \leqslant 1, \sum_{i=1}^{n} w_i = 1.$ (6)

第 3 步 同理计算软件项目第一层的总体维护风险的熵权系数 α 和总体维护风险 w,计算公式分别为:

进行归一化处理:$\alpha = -\frac{1}{\log_2 n}\sum_{j=i}^{n} \beta_i \cdot \log_2 \beta_i$, (7)

总体维护风险:$w = 1 - \alpha, 0 \leqslant w \leqslant 1, 0 \leqslant \alpha \leqslant 1.$ (8)

第 4 步 对照软件系统总体维护风险 w 与表 3 所示的维护级别,评价软件系统的维护风险.

2.4 维护风险符合度说明 项目可维护级别定义为 5 个级别,分别为:极为关键、较大、程度适中、较小和可以忽略. 而使用信息熵的原理进行建模,使得计算的维护风险不是线性变化的,而是对数级的变化,所以分级之间的间隔值需要符合对数级. 每一级的具体定义如表 3 所示.

表 3 系统维护级别定义

Tab. 3 The definition of system maintenance level

等级	值	影响描述
w_1 极为关键	0	可能引起灾难性的损失,几乎可以考虑不用维护
w_2 较大	0.10	维护会导致大量的损失,付出很大的努力也无济于事
w_3 程度适中	0.30	直接影响系统的运行,会降低系统的运行效率
w_4 较小	0.60	维护失败的影响很小,只需较小的努力就可弥补
w_5 可以忽略	1	维护失败微乎其微,几乎可以不考虑维护失败

3 模型实例分析

3.1 实例简介 财务软件项目简介,该项目主要由会计核算模块和管理模块两大模块组成. 其中会计核算部分除可以按分户设分户明细账外,还可以按会计科目设科目明细账,并均可以对其进行经费计划管理及暂付款管理. 该软件允许多用户同时操作,也可以安装服务器版软件后进行单机操作,经过扩充后,可以实现联网实时操作.

软件主要功能有:完成记账凭证录入和修改、计算机自动复核、凭证打印、个人记账的工作;完成全体凭证自动复核、当日结记账、打印当日记账、打印当日报表、当日处理初始的工作;完成打印总账、打印明细账、打印多栏账、打印月报表、通用电子报表的工作;完成余额查询、明细账查询、暂付款查询、记账凭证查询的工作;完成本年计划录入、本年计划调整、计划项目汇总表、计划执行明细表、计划执行对账单、经费数据维护的工作;完成借款打印、催款打印的工作;完成月终结转、年结模型设置、年终结转的工作;完成银行对账、支票号码管理的工作;完成科目、分户账的设置及起始余额的输入、经费项目的设置及起始余额输入、暂付款起始余额输入、系统参数设置、建账审计的工作;完成自动校验账户、增加用户、删除用户、修改密码、设置权限、数据库恢复的工作.

3.2 实例计算 依据 ISO/IEC 25010—2011 标准的扩展对财务管理软件项目建立如图 1 的维护风险层

次图,再按第 2 节中给出的信息熵的维护风险度量模型计算,分析评价软件项目的维护风险. 为计算方便,本实例简化了子项指标的选择数目.

第 1 步　根据表 1 和表 2 的符合度定义以及财务管理软件项目的专家评分的均值,得到层次图的第三层的 11 个子项指标的不确定性符合度不确定程度 $P_{i,j}$ 和损失度 $c_{i,j}$,他们分别为(0.65,0.75,0.85,0.60,0.70,0.80,0.45,0.95,0.90,0.40,0,80),(0.70,0.75,0.75,0.40,0.80,0.75,0.50,0.60,0.55,0.90,0.85))

第 2 步　根据公式(4),(5)和(6),计算第 2 层各个子项目标的熵权系数,得系数 $m_{i,j}$ 和总体维护风险值 w_i,分别为:

$m_{1,1}=0.6745, m_{1,2}=0.7500, m_{1,3}=0.7714;$

$m_{2,1}=0.7500, m_{2,2}=0.7714, m_{2,3}=0.4899, m_{2,4}=0.7483, m_{2,5}=0.7746, m_{2,6}=0.4743;$

$m_{3,1}=0.7500, m_{3,2}=0.7746, m_{3,3}=0.7540, m_{3,4}=0.7036;$

$m_{4,1}=0.7483, m_{4,2}=0.7746, m_{4,3}=0.6000, m_{4,4}=0.8246;$

$\alpha_1=0.6208, \alpha_2=0.8564, \alpha_3=0.6303, \alpha_4=0.6350;$

$w_1=0.3016, w_2=0.1142, w_3=0.2940, w_4=0.2902.$

第 3 步　根据公式(7)和(8),计算软件项目的总体维护风险的熵权数值 β 和总体维护风险值 w,分别为:

$\beta=0.9582, w=1-\beta=0.0418.$

需要说明的是,初始的符合度量化值采用 2 位精度就可以说明实际情况,但是在仿真计算中需要采用 4 位精度. 这样可以避免和降低计算过程中的误差和误差积累.

3.3　维护风险分析　对软件项目的总体维护风险的分析,对照维护风险等级定义,可知实际软件项目的总体维护成功的风险 $w=0.0418$,系统的维护风险值 $0<w<0.10$,并且接近于 0.05,属于维护成功的可能性很小的范围,故软件项目维护成功的风险很大.

第二层的理解风险的总体维护风险为 0.1142,属于风险较大的级别,是影响软件总体维护的主要因素. 而修改风险、测试风险和复审风险则比较平均,都在 0.3 左右,对系统的总体维护的风险的影响也比较平均,相对理解风险就更小. 修改风险的总体维护风险值为 0.3016,数值最大,对总体维护风险的影响最小.

3.4　维护风险比较　实际数据的仿真结果表明,采用信息熵对软件项目的维护风险进行分析与实际情况比较吻合,证明该度量算法比较合理.

既满足以 ISO/IEC 25010—2011 标准测量与分析软件项目维护风险的要求,又克服了 ISO/IEC 25010—2011 标准中不区分度量单元的共享和数据的共享的缺点,解决了度量单元测量的复杂性问题,二者兼顾,有利于进行量化.

信息熵维护风险分析方法是扩展 ISO/IEC 25010—2011 标准的体系结构建立层次图,依据测评实践获取原始数据,使用信息熵分析算法计算各个风险因素的风险值,将定性分析与定量计算相结合,克服了标准中控制措施对风险的影响程度分析不足的缺点,保证风险分析的结论可信度和可重复性.

该模型仿真数据直接来源于测评实践,解决了文献[2]软件维护时需要大量统计程序代码的信息的缺憾;同时,解决了文献[3]只是针对以面向对象软件的度量准则作为预测因子的缺陷,而是对所有的软件项目都实用;文献[4,6]虽然使用了定量和定性的分析方法,但是模型底层目标只决定一个上层目标,模型的低层之间没有交叉,该模型解决了底层指标影响多个上层目标的情况.

克服了文献[5]方法的局限性,对软件项目的维护风险预测分析具有普遍意义,如在实践中依据 ISO/IEC 25010—2011 标准的要求对目标软件项目进行维护风险测量与分析;使用层次图建立目标软件项目进行维护风险测量与分析的层次结构模型;用信息熵分析算法不仅可计算分析目标软件项目的总体维护风险,而且可分析各层次维护风险因素的风险大小.

4 结束语

依据ISO/IEC 25010—2011标准对软件项目的维护风险因素进行分析,在建立层次图的基础上,提出了一种基于信息熵的软件项目维护风险的定量计算方法,并形成了比较完整的分析模型.模型的初始数据是真实可靠的数据,通过信息熵分析模型计算软件项目的维护风险,事先评估软件项目的维护风险水平,从而更加科学地进行维护风险的管理和应对.

参考文献:

[1] 郑人杰.软件工程(高级)[M].北京:清华大学出版社,1999:105-107.
[2] 孙晓雅,陈静.基于程序信息的软件可维护性度量[J].山东科学,2010,23(4):68-71.
[3] 王李进,胡欣欣.基于支持向量机的面向对象软件可维护性预测[J].北华大学学报,2010,11(3):282-285.
[4] 陈雪娟,潘梅森,雷超阳.基于SVM的软件可维护性评估模型研究[J].计算机工程与设计,2008,29(3):566-570.
[5] 赵金伟,郝克刚,葛玮.基于MIM的软件度量扩展模型对软件可维护性的度量[J].计算机应用,2007,27(6):1 430-1 433.
[6] 刘万远,张卫东,王伟.基于层次分析法的软件可维护性评价[J].四川兵工学报,2011,32(7):96-98.
[7] ISO / IEC 25010—2011. Systoms and software engineering – systoms and software quality requirements and evaluation – system and sofe ware quality models[S/OL]. 2011 – 10 – 1. http://www.chinaios.com.
[8] 孙东川,林福永.系统工程引论[M].北京:清华大学出版社,2004:97-98.
[9] ROBERT M Gray. Entropy and information theory[M]. Springer,2011.
[10] JAYNES E T. Information theory and statistical mechanics [J]. Phy Rev,1957,106 (5) :620-630.
[11] 王洋,赵宗敏,吴海涛.重构改善软件可维护性的量化研究[J].微计算机应用,2009,30(10):36-42.

Measurement of a software that based on comentropy and maintained risk

WANG Jia, ZHOU Hua, LIANG Zhi-hong, DAI Fei, BAI Li-rui
(School of Software, Yunnan University, Key Laboratory of Software Engineering of Yunnan Province, Kunming 650091, China)

Abstract: In order to solve the problem that the uncertain information is difficult to quantitative analysis in the process of software maintenance, this paper uses the comentropy to quantitative measure the risk in software maintenance. A quantitative analysis algorithm about the comentropy is proposed, and the software risk maintenance model is put forward based on the comentropy, in addition, this paper uses the comentropy algorithm quantitative calculate the uncertain degree and loss degree in the process of software maintenance. The emulational result demonstrates that based on the software risk maintenance model, using the comentropy algorithm is able to quantitative measure the risk in software maintenance.

Key words: software maintenance risk; comentropy; risk maintenance model; level analysis; quantitative measure