文章编号:1671-4598(2020)03-0260-07 DOI:10.16526/j.cnki.11—4762/tp.2020.03.054 中图分类号:TP311.5 文献标识码:A

基于改进 AHP—Fuzzy 的面向测试的软件质量评价模型

王勇利，王 鑫
（中国人民解放军 91404 部队，河北 秦皇岛 066001）

摘要：通用软件质量评价模型过于庞大不便于其在软件测试活动中直接应用，软件测试评价空间有限不足以消除传统 AHP 等方法中人为主观影响，以上诸多因素限制了面向测试的软件质量评价在工程实践中的有效开展；鉴于此，为解决面向测试的软件质量评价难以有效开展的问题，基于对传统 AHP—Fuzzy 模型的适当改进，提出一种面向测试的软件质量评价模型——SFSE（system failed state evaluating model，系统失效评估模型）；SFSE 模型以软件测试活动实测数据为输入，首先对各叶子节点进行单因素实测评价，之后采用从下至上、逐层迭代的方式对各级评估原子集实施评价，最终实现对被测软件质量状态的整体评估；实验结果表明，拓展综合评价集向量提供了可调节的量化评估参数，利用基于实测问题规模构建评估模糊评价矩阵、将权重投射至评价因素集上的模糊子集等方式，SFSE 模型可在无需人工干预的情况下实现对被测软件质量状态的定量评价。

关键词：软件测试；软件质量评价；层次分析；模糊综合评价

A Software Quality Evaluation Model for Software Testing Based on Improved AHP—Fuzzy

Wang Yongli, Wang Xin
(No. 91404 Unit of PLA, Qinhuangdao 066001, China)

Abstract: To solve the problem that the software quality evaluation for software—testing cannot be effective in engineering practice, combining the characteristics of software testing implementation, SFSE Model (system failed state evaluating model), a new kind of software quality evaluation model for software testing, is proposed. The SFSE Model, well adapted for the characteristics of software testing implementation, is based on the appropriate modification of the traditional AHP—Fuzzy model. On the basis of proposing concepts such as System Failed State Evaluating, Synthetic Evaluating Set and so on, the SFSE Mode evaluates all leaf node based on the information of software—testing executing firstly. Then the SFSE Mode implements evaluation for every Atomy—Evaluating—Scope at all levels in the form of iteration step by step. The experimental results show that the Expanded—Synthetic—Evaluating—Set provides a quantitative assessment of the adjustable parameters and an adjustable mechanism for quantitative evaluating, based on the measured problem size building assessment fuzzy evaluation matrix and a weighted projection to the evaluation factor set of fuzzy subset, and the SFSE Model can automatically implement each node and whole software quality evaluation in the form of without human intervention.

Keywords: software testing; software quality evaluation AHP; fuzzy synthetic evaluation

0 引言

随着信息化程度的不断提高，软件在各行业、领域、系统中所占的比重逐渐增大，所发挥的作用日益重要，越来越多系统的功能界面主要通过软件形态呈现，软件质量水平的高低在某种程度上可直接影响或间接反映出系统整体质量状态[1]。软件度量学的概念最早由 Hurtwick 和 Rubey 于 1968 年提出[2]，历经 50 多年发展，软件质量评价理论及相关技术的研究取得了长足进步。一方面多种效能分析或评估方法被大量借鉴和应用，例如：软件质量评价研究人员通常使用层次分析法（analytic hierarchy process，AHP）确定子特性在特性及子特性线性函数关系式中的权重[3]，同时也有一部分研究人员采用模糊综合评价法（fuzzy synthetic evaluation）确定特性与子特性之间的关系；另一方面形成了由 ISO/IEC 25010：2011、ISO/IEC 25051：2014 等各类标准规范所构成的多个通用软件质量评价体系，不仅描述了软件产品的质量模型，还对软件产品质量评价的方法和过程进行了规定[4-5]。

软件测试是在有限的规定条件下，执行程序操作或触发程序运行，以尽可能暴露程序错误，评估软件能否满足需求的过程[6]，故一定程度上可基于软件测试的执行对被测软件或系统实施质量评价。目前软件测试早已跳出最初仅对程序进行“调试”的局限，且成功借鉴了工程化的理念和方法，其在内涵上逐步引入“质量”的概念[7]，出现了类似于“软件测试是实现对软件质量的度量，是将对某一程序或系统的相关属性进行评价作为目标的任一活动”的定义[8]，开展“基于软件测试的软件质量评价（又称为面

收稿日期:2019-11-25； 修回日期:2020-01-15。

作者简介:王勇利(1982-),男,安徽巢湖人,硕士,工程师,软件测试人员,主要从事软件测试及其工程化方向的研究。

向测试的软件质量评价)”的意义和必要性得到普遍认可并不断强化。

然而与各类软件质量度量及评价的理论和模型不断推陈出新、受关注度不断提高形成鲜明对比的是，当前在工程实践中软件测试仍然以暴露、定位、分析并客观呈现各类具体问题为主，测试的首要任务和主要目的仍旧是验证软件是否满足规定的需求。正如 IEEE 在其软件工程术语的行业标准中对软件测试所下的定义“软件测试是使用人工或自动的手段来运行或测定某个软件系统的过程，其目的在于检验它是否满足规定的需求或弄清预期结果与实际结果之间的差别”[9]，软件测试“测而不评”、“以测代评”的现象普遍存在，究其原因，主要包括：

1）通用软件质量评价标准体系不能直接应用于基于测试的软件质量评价。面向测试的软件质量评价主要面向软件测试过程，具有其自身特殊性，然而以 ISO/IEC 25000 系列标准为代表的通用软件质量评价体系均基于软件全寿命周期等全局视角对软件质量评价活动所进行的系统性规范。通用软件质量评价标准一方面不能满足面向测试的软件质量评价活动的特定要求，另一方面又显得过于庞大和复杂，无法直接使用。例如：GB/T 25000．10－2016 就将软件的产品质量属性从“正向”视角划分为功能性、可靠性、易用性、性能效率、维护性、可移植行等 8 个特性，并将其进一步细化为多个子特性[10]，而测试先天所具有的证伪性决定了其只能从“反向”视角检测出软件实现在各特性中是否存在问题，以及存在什么样的问题。

2）现有系统效能分析或评估方法也不能直接应用于基于测试的软件质量评价。例如：传统 AHP 虽然采用九级标度法（9 标度法）实现了多个元素针对上层元素的定量排序且易于操作[11]，但仅基于 1～9 级的单向标度无法满足面向测试的软件质量评价须同时涵盖正、反两个评价维度的特定要求；经典模糊综合评价方法虽然基于模糊数学理论实现了对定性问题域的定量分析及评价，但其通常首先需要利用人工评判打分或专家系统等方式定性地确定各评价因素在评判等级论域（评判等级集合）上的隶属度，且还需要采用专家直接赋值等方式确定权重以实现后续的模糊综合评价[12]，因此受参与者主观因素的干扰较为显著，然而软件测试评价空间极为有限，不足以消除上述人为主观因素干扰（AHP 方法在权重确定过程中也存在类似问题[13]），易导致评价过程受主观因素干扰从而降低评价的客观性和准确性。

为解决面向测试的软件质量评价不能有效开展的实际问题，在引入失效度、系统失效评估等概念的基础上，结合软件测试工程实践特点，对层次分析、模糊综合评价进行适当改进和拓展，提出一种可行性较强的面向测试的软件质量评价模型——SFSE 模型（system failed state evaluating model，系统失效评估模型），该模型以软件测试执行所发现的问题规模等实测信息为输入，通过分析被测系统或软件的失效程度，从“反向”视角间接实现对被测系统及软件质量状态的量化评价。

1　相关定义说明

定义 1（失效度 the Failure Level）：失效度是评价因素或评估对象失效程度的量化表述，失效度的量化形式表述参见 3.3.2 所述。

定义 2（系统失效评估 system failed state evaluating）：系统失效评估是基于系统运行过程中出现的各类缺陷或错误的严重等级、规模的分布特性，对系统运行失效性进行定量的分析和评价，并实现对系统质量状态的间接评估。

定义 3（测试评价因素层次结构 layered structure for evaluating）：基于对被测系统的理解与分析，参考层次分析法，从软件测试角度对被测系统功能、性能、接口等各项指标或特性，以及安全性、恢复性、边界、强度等软件测试需求进行逐层分解，抽取评价因素，建立与之对应的树状结构模型，即为被测系统的测试评价因素层次结构。可采用层次结构图图形化表述测试评价因素层次结构，其中节点表示评估对象或评价因素，由连接线表明从属关系。

测试评价因素层次结构分为两大部分：第一部分为顶层，仅包含单一的根节点，表示系统本身；第二部分为评价因素层，可包含多个子层，每个子层可包含多个节点（评价因素），为简化问题处理，本文将测试评价因素层次结构限制为二维树结构，即任意下层节点（子节点）仅归属于单一特定上层节点（母节点），如图 1 所示。

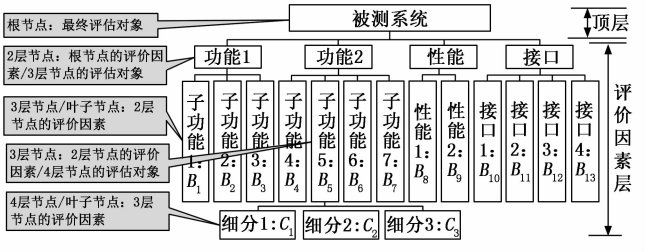

图 1　测试评价因素层次结构示意图

定义 4（评估对象 evaluating target）：评估对象为对其实施失效评估的具体对象（可以是实体，也可以是属性、特性等抽象概念），评估对象的概念具有相对性，在测试评价因素层次结构中，除叶子节点（某一路径的最底层节点，该节点无任何子节点）外的各层中的各节点均可视为其下层子节点的评估对象，系统失效评估最终的评估对象为根节点即被测软件或系统，如图 1 所示。

定义 5（综合评价集 synthetic evaluating set）：实施模糊综合评价时任一评价者针对若干（单个或多个）因素实施模糊评价所能选用的评价结果全集构成的评判等级论域，称为评价集，设 V 为评价集：

$$V=\{v_1,v_2,\cdots,v_j,\cdots,v_m\} \tag{1}$$

评价集 V 中包含的评价结果（评价分量）的规模（数量）称为该评价集的度，表示为 $deV(V)$。例如：针对式（1）所表示的评价集 V 有 $deV(V)=m$，称 V 为 m 度评价集，表示 V 的度为 m，即 V 共包含 m 个（级）评价结果。

软件测试通常根据所发现问题的危害程度、整改开销等因素，将问题划分为多个严重性等级，可直接使用有关成熟的工程实践构建相应评价集，例如 GJB2786/A 从开发方视角根据软件问题对研发过程的影响程度、造成的危害性等方面将软件问题划分为 4 个严重性等级[14]，基于此种软件问题严重性等级划分可构建通用软件问题评价集如下：

$$V_{2786/A}=\{1\text{ 级(致命)},2\text{ 级(严重)},3\text{ 级(一般)},4\text{ 级(轻微)}\} \tag{2}$$

如式（2）的通用软件问题评价集适用于对软件问题的分析和评价，但由于缺少对未发现问题情形的评判，故无法实现对被测系统或软件失效程度以及质量状态的评估，鉴于此，本文在系统失效评估中引入“综合评价集”的概念：

综合评价集（synthetic evaluating set）在通用软件问题严重性等级评价集的基础上，引入对“无问题（0 问题）”状态的标定，构成系统失效评估所需评价分量的完整集，即为综合评价集，其一方面仍可用于标定软件问题严重等级，另一方面亦适用于对系统失效程度、质量状态实施评判。

设：V_s 为基于通用软件问题严重性等级评价集 V 所构建的综合评价集，v_∞ 为标定“无问题（0 问题）”状态的评价分量（称为 0 项增量），则将 V_s 表示为：

$$V_s=V\bigcup\{v_\infty\}=\{v_1,v_2,\cdots,v_j,\cdots,v_m,\mid v_\infty\} \tag{3}$$

式（3）中“∣”用于将 V 中各元素与 v_∞ 分量进行分割，以显性区分 0 项增量。为便于形式化描述，本文引入分量函数，将综合评价集 V_s 进一步简化表述为：

$$V_s=V\bigcup\{v_\infty\}=(v_{(j)}、v_\infty)\quad 1\leqslant j\leqslant deV(V) \tag{4}$$

式（4）中 $v_{(j)}$ 为集合 V 的分量函数形式，既可表示 V 中的分量全集所构成的向量，亦可表示 V 中任意分量，$(v_{(j)}、v_\infty)$ 为 V_s 的分量函数形式，用于表示 V_s 所含的全部分量（元素）。

综合评价集 V_s 的度定义为：

$$deV(V_s)=deV(V)+deV(\{v_\infty\})=deV(V)+1=m+1 \tag{5}$$

称式（3）所述综合评价集 V_s 的度为 $m+1$ 度。

综合评价集 V_s 可同时作为对测试评价因素层次结构中各评估对象进行定量评价、对全系统的失效程度进行评判的等级标尺，将测试过程中发现的具体软件问题或缺陷的严重性标定与系统整体失效程度及质量状态的评判统一至同一个评判维度。

定义 6（评价因素集 set－of－evaluating－elements）：评价因素（evaluating element）为影响或制约评估对象的某一具体要素或特性。影响和制约某一评估对象的所有评价因素的全集构成评价因素集。

设 U 表示包含 n 个评价因素的评价因素集，则：

$$U=\{u_1,u_2,\cdots,u_i,\cdots,u_n\} \tag{6}$$

评价因素集 U 中所包含的评价因素的数量称为该评价因素集的度，表示为 $deU(U)$，称 U 为 $deU(U)$ 度评价因素集，以式（6）为例，其表示 U 为 n 度。

评价因素在测试评价因素层次结构中同样具有相对性，测试评价因素层次结构中除顶层根节点外的任一节点均可视为其紧邻上一层母节点的评价因素（该母节点为该因素的评估对象），如图 1 所示。

定义 7（评估原子集 atomy evaluating scope）：评估原子集定义了实施失效评估的最小分析范围（分析对象），包括评估对象及该其对应的评价因素集。

令 ES 为评估原子集，则 ES 可表示为序偶形式：

$$ES=[et,U] \tag{7}$$

其中：et 为评估对象，集合 U 为 et 对应的评价因素集。在测试评价因素层次结构中，任意评估原子集表示某一非叶子节点 et（评价对象）以及其下一层所有子节点集合 U（评价因素集）所构成的有序偶。

2 单因素实测评价

单因素实测评价是基于实测过程中所发现的问题规模，对测试评价因素层次结构中单一节点所实施的模糊评价。设有给定的综合评价集 $V_s=V\bigcup\{v_\infty\}$，U 为任意评价因素集，令 u_i 为 U 中的任一评价因素，基于采集 u_i 在实际测试过程中所发现的各类严重性等级问题的规模（数量）构建模糊子集 $\underset{\sim}{RT}_i$：

$$\begin{aligned}\underset{\sim}{RT}_i=&\left(\frac{\underset{\sim}{RT}_i(v_1)}{v_1}+\frac{\underset{\sim}{RT}_i(v_2)}{v_2}+\cdots+\frac{\underset{\sim}{RT}_i(v_j)}{v_j}+\right.\\&\left.\cdots+\frac{\underset{\sim}{RT}_i(v_m)}{v_m}\right)+\frac{\underset{\sim}{RT}_i(v_\infty)}{v_\infty}=\\&\{(v_1,\underset{\sim}{RT}_i(v_1)),(v_2,\underset{\sim}{RT}_i(v_2)),\cdots,\\&(v_j,\underset{\sim}{RT}_i(v_j)),\cdots,(v_m,\underset{\sim}{RT}_i(v_m)),\mid(v_\infty,\underset{\sim}{RT}_i(v_\infty))\}=\\&(\underset{\sim}{RT}_i(v_1),\underset{\sim}{RT}_i(v_2),\cdots,\underset{\sim}{RT}_i(v_j),\cdots,\underset{\sim}{RT}_i(v_m),\\&\mid\underset{\sim}{RT}_i(v_\infty))=(rt_{i1},rt_{i2},\cdots,rt_{ij},\cdots,rt_{im},\mid rt_{i\infty})\end{aligned} \tag{8}$$

称 $\underset{\sim}{RT}_i$ 为针对单因素 u_i 的实测评价，求取 $\underset{\sim}{RT}_i$ 的过程为实测评价单因素 u_i（对 u_i 实施实测评价）。

若 u_i 所发现的问题规模（数量）为 s_i，令：

$$s_i=\sum_{j=1}^{m}s_{ij} \tag{9}$$

求取式（8）中 V_s 各分量 $(v_{(j)}、v_\infty)$ 对于 $\underset{\sim}{RT}_i$ 的隶属度 rt_{ij}、$rt_{i\infty}$ 取决于 s_i：

$$\begin{cases}s_i\neq 0\text{ 时}\begin{cases}rt_{ij}=\dfrac{s_{ij}}{\sum\limits_{j=1}^{m}s_{ij}}\\rt_{i\infty}=0\end{cases}\\s_i=0\text{ 时}\begin{cases}rt_{ij}=0\\rt_{i\infty}=1\end{cases}\end{cases} \tag{10}$$

式（8）、（9）、（10）中，$m=deV(V)$，$1\leqslant j\leqslant m$，s_{ij} 为评价因素 u_i 在综合评价集 V 的 j 分量 v_j 上的实测问题规模（数量）。由式（10）可得 V_s 各分量（$v_{(j)}、v_\infty$）对于 $\underset{\sim}{RT}_i$ 的隶属度具有归一化属性，即：

$$rt_{i\infty}+\sum_{j=1}^{m}rt_{ij}\equiv 1 \tag{11}$$

单因素实测评价 $\underset{\sim}{RT}_i$ 具有以下特性：

1）“0 项增量”对应分量 $rt_{i\infty}$ 的取值范围为（0，1）；

2）$rt_{i(j)}$ 与 $rt_{i\infty}$ 的取值呈现非零互斥性，即：

$$\begin{cases}\text{若 } rt_{i\infty}=0,\text{则 } rt_{i(j)}\neq\overrightarrow{0}\\ \text{若 } rt_{i\infty}=1,\text{则 } rt_{i(j)}=\overrightarrow{0}\end{cases}$$

3）$\underset{\sim}{RT}_i$ 各分量之和满足归一化要求；

可采用最大隶属度原则获取单因素实测评价的评价结果，也可直接将 $\underset{\sim}{RT}_i$ 的模糊子集原始形式作为评价结果以保留更多的评价信息。

针对单因素 u_i 的实测评价的实质是构建论域为综合评价集 V_s 的模糊子集，以此实现基于实测问题规模的单因素模糊评价。由于单因素实测评价是对具体评价因素实施的独立评价，故无需考虑评价因素的权重因素的影响。

3　单评估原子集失效评估

设有任一评估原子集 $ES=[et,U]$，$deU(U)=n$，给定 $m+1$ 度综合评价集 $V_s=V\cup\{v_\infty\}$，对评估原子集 ES 的失效评估是参考一级模糊综合评价[15]，采用加权综合 ES 所辖评价因素集 U 下所有评价因素的单因素模糊评价结果（参见 3.2）等方法，针对评估对象 et 求取论域为综合评价集 $V_s=V\cup\{v_\infty\}$ 的模糊子集 $\underset{\sim}{B}^{[et]}$，实现对 et 多因素模糊综合评估：

$$\underset{\sim}{B}^{[et]}=(b_1^{[et]},b_2^{[et]},\cdots,b_j^{[et]},\cdots,b_m^{[et]},|\,b_\infty^{[et]}) \tag{12}$$

式（12）所述的 $\underset{\sim}{B}^{[et]}$ 的分量函数形式为：

$$\underset{\sim}{B}^{[et]}=(b_{(j)}^{[et]},b_\infty^{[et]}) \tag{13}$$

式（13）中 $b_{(j)}^{[et]}$ 是针对评价集 V 实施模糊评价所得模糊向量 $(b_1^{[et]},b_2^{[et]},\cdots,b_j^{[et]},\cdots,b_m^{[et]})$ 的分量函数。

3.1　单评估原子集失效评估一般形式

设经测试，et 所发现的软件问题规模为 S_{et}，则：

$$S_{et}=S_U=\sum_{i=1}^{n}s_i \tag{14}$$

式（14）中 $n=deU(U)$，$1\leqslant i\leqslant n$，S_U 为 U 的实测问题规模。对给定的综合评价集 $V_s=V\cup\{v_\infty\}$，$\underset{\sim}{B}^{[et]}$ 定义为：

$$\underset{\sim}{B}^{[et]}=\begin{cases}\underset{\sim}{A}^{\circ}\underset{\sim}{R}=(b_{(j)}^{[et]},b_\infty^{[et]}\equiv 0) & S_{et}\neq 0\text{ 时；}\\ (\forall\, b_{(j)}^{[et]}=0,b_\infty^{[et]}\equiv 1) & S_{et}=0\text{ 时；}\end{cases} \tag{15}$$

式（15）中模糊子集 $\underset{\sim}{A}$ 为针对 U 中每一评价因素所构建的动态权重集（参见 2.3 节），运算符$^\circ$表示综合评判所采用的合成 6 算子（参见 2.4 节），模糊矩阵 $\underset{\sim}{\boldsymbol{R}}$ 为针对评估对象 et 所构建的模糊评价矩阵（参见 2.2 节）。

单评估原子集失效评估结果 $\underset{\sim}{B}^{[et]}$ 支持“0 项增量”拓展，能够实现对被测系统或软件全面的失效评估与质量状态评判。由式（15）可得 $\underset{\sim}{B}^{[et]}$ 在形式上具有与单因素实测评价结果一致的结构（参见“1 单因素实测评价”）。这种评估模式的统一性可确保后续针对更上层节点以及全系统的失效评估迭代解算的有效性。

3.2　构建模糊评价矩阵

3.2.1　单因素模糊评价

设对评价因素集 U 中任一评价因素 u_i 的单因素模糊评价为 $\underset{\sim}{R}_i$，其定义主要取决于 u_i 是否为叶子节点：

$$\begin{cases}u_i\text{ 为叶子节点},\underset{\sim}{R}_i=\text{对单因素 }u_i\text{ 的实测评价}\underset{\sim}{RT}_i\\ u_i\text{ 为非叶子节点},\underset{\sim}{R}_i=\begin{matrix}u_i\text{ 作为评估对象的单评估}\\ \text{原子集失效度评估}\underset{\sim}{B}^{[u_i]}\end{matrix}\end{cases} \tag{16}$$

3.2.2　模糊评价矩阵

评价因素集 U 中全部评价因素的单因素模糊评价结果为 n 个模糊子集：

$$\begin{aligned}&\text{评价因素 }u_1\quad \underset{\sim}{R}_1=(r_{11},r_{12},\cdots r_{1m},|\,r_{1\infty})\\ &\text{评价因素 }u_2\quad \underset{\sim}{R}_2=(r_{21},r_{22},\cdots r_{2m},|\,r_{2\infty})\\ &\qquad\vdots\qquad\qquad\vdots\\ &\text{评价因素 }u_i\quad \underset{\sim}{R}_i=(r_{i1},r_{i2},\cdots r_{im},|\,r_{i\infty})\\ &\qquad\vdots\qquad\qquad\vdots\\ &\text{评价因素 }u_n\quad \underset{\sim}{R}_n=(r_{n1},r_{n2},\cdots r_{nm},|\,r_{n\infty})\end{aligned} \tag{17}$$

式（17）中全体模糊集构建出模糊评价矩阵 $\underset{\sim}{\boldsymbol{R}}$：

$$\underset{\sim}{\boldsymbol{R}}=\begin{bmatrix}\underset{\sim}{R}_1\\ \underset{\sim}{R}_2\\ \cdots\\ \underset{\sim}{R}_i\\ \cdots\\ \underset{\sim}{R}_n\end{bmatrix}=\begin{bmatrix}r_{11}&r_{12}&\cdots&r_{1m}&r_{1\infty}\\ r_{21}&r_{22}&\cdots&r_{2m}&r_{2\infty}\\ \vdots&\vdots&\cdots&\vdots&\vdots\\ r_{i1}&r_{i2}&\cdots&r_{im}&r_{i\infty}\\ \vdots&\vdots&\cdots&\vdots&\vdots\\ r_{n1}&r_{n2}&\cdots&r_{nm}&r_{n\infty}\end{bmatrix} \tag{18}$$

由 $\underset{\sim}{RT}_i$、$\underset{\sim}{B}^{[et]}$ 定义可得 $\underset{\sim}{\boldsymbol{R}}$ 具有以下特性：

$$\begin{cases}S_{et}\neq 0\text{ 时}\begin{cases}\exists\, rt_{i\infty}=0\\ \text{若 } rt_{i\infty}=0,\text{则 } rt_{i(j)}\neq\overrightarrow{0}\\ \text{若 } rt_{i\infty}=1,\text{则 } rt_{i(j)}=\overrightarrow{0}\end{cases}\\ S_{et}=0\text{ 时}\begin{cases}\forall\, rt_{ij}=0\\ \forall\, rt_{i\infty}=1\end{cases}\end{cases} \tag{19}$$

3.3　动态权重集

当 $S_{et}\neq 0$ 时，综合各评价因素实测问题规模以及各类问题严重性（危害程度）等因素，通过量化各评价因素的失效程度，构建出基于实测的权重分布即为动态权重集。

3.3.1　拓展综合评价集向量

定义 5 阐述的综合评价集仅标定了测试过程中所发现的各问题的严重性等级，为实现对被测系统的失效度以及质量状态的定量评价，须将综合评价集中任意评价分量由单一语义表述向量化评估、评价等级等多个维度拓展，构建成相应的拓展综合评价集向量。

设拓展 V_s 后所得的拓展综合评价集向量为 $\vec{V}_s^*$：

$$\vec{V}_s^*=(\vec{v}_1^*,\vec{v}_2^*,\cdots,\vec{v}_j^*,\cdots,\vec{v}_m^*,|\,\vec{v}_\infty^*) \tag{20}$$

将 $\vec{V}_s^*$ 中任意分量 $\vec{v}_x^*(x\in\{[1,m]\cup\{\infty\}\})$ 定义为：

$$\vec{v}_x^*=(GoF,V,CoE,GoE) \tag{21}$$

采用 [] 运算符引用向量 $\vec{v}_x^*$ 各分量，$\vec{v}_x^*$ 向量各分量含义

定义如表 1。

表 1 拓展综合评价集分量描述

序号	分量	引用方式	含义
1	GoF	$\vec{v}_x^*[GoF]$	失效等级(grade－of－faultiness)：软件问题严重等级以及被测软件/系统失效度的语义标定
2	V	$\vec{v}_x^*[V]$	评估量值(value)：软件问题严重等级以及被测软件/系统失效度的量化标定，亦可反向标定软件或系统的质量状态
3	CoE	$\vec{v}_x^*[CoE]$	评价结语(comment－of－evaluating)：被测软件/系统质量状态等级化的语义标定
4	GoE	$\vec{v}_x^*[GoE]$	评价级别(grade－of－evaluating)：失效度或质量状态的统一符号化表述

拓展综合评价集向量 $\vec{V}_s^*$ 各分量 $\vec{v}_x^*$ 按照失效程度由高至低排列（对应质量状态由低至高），分量序号 x 越大表示问题严重性等级越低、软件或系统失效程度越低、质量状态越好，例如：$\vec{v}_1^*$ 表示问题最为严重、失效度最高、质量状态最差，$\vec{v}_\infty^*$ 表示无问题、失效度为 0、质量状态最好。

应在实施测评前首先确定 $\vec{V}_s^*$ 以及其分量 $\vec{v}_x^*$（可参照式(21) 方式定义 GoF）所构成的综合评价指标体系，其中对 $\vec{v}_x^*[V]$ 分量的构建应遵循以下原则：

$$\begin{cases}\vec{v}_\infty^*[V]\equiv 0\\ \vec{v}_1^*[V]>\vec{v}_2^*[V]>\cdots\vec{v}_j^*[V]\cdots>\vec{v}_m^*[V]>\vec{v}_\infty^*[V]\end{cases} \tag{22}$$

3.3.2 评价因素及评价因素集失效度

对 $\forall u_i\in U$，定义评价因素 u_i 的失效度 FL_{u_i} 为：

$$FL_{u_i}=\sum_{j=1}^{m}(s_{ij}\times\vec{v}_j^*[V]) \tag{23}$$

评价因素集 U 的失效度 FL_U 为：

$$FL_U=\sum_{i=1}^{n}FL_{u_i}=\sum_{i=1}^{n}\left(\sum_{j=1}^{m}(s_{ij}\times\vec{v}_j^*[V])\right) \tag{24}$$

3.3.3 构造动态权重集

当 $S_{et}\neq 0$ 时，设针对评价因素集 U 构建对应的动态权重集为 $\underset{\sim}{A}$，则模糊集 $\underset{\sim}{A}$ 定义为：

$$\underset{\sim}{A}=(\underset{\sim}{A}_{(u_1)},\underset{\sim}{A}_{(u_2)},\cdots,\underset{\sim}{A}_{(u_i)},\cdots,\underset{\sim}{A}_{(u_n)})=(a_1,a_2,\cdots,a_i,\cdots,a_n) \tag{25}$$

式 (25) 中 a_i 为评价因素 u_i 对应的动态权重 ($1\leqslant i\leqslant n$)，且 a_i 定义为：

$$a_i=\underset{\sim}{A}_{(u_i)}=\frac{FL_{u_i}}{FL_U} \tag{26}$$

显然，在 $S_{et}\neq 0$ 的情况下，动态权重 a_i 满足非负性和归一化要求：

$$\sum_{i=1}^{n}a_i=1,a_i\geqslant 0 \tag{27}$$

3.4 合成算子选取

根据模糊综合评价过程中所选取的合成算子。通常将评价模糊子集的求取分为“模式Ⅰ：M（∧，∨）”、“模式Ⅱ：M（·，∨）”、“模式Ⅲ：M（∧，⊕）”、“模式Ⅳ：M（·，⊕）”、“模式Ⅴ：M（·，+）”五种模式[16]。∧、∨、⊕运算在评价因素较多等情况下可能导致有用信息丢失，故模式Ⅰ～Ⅳ适用于关注评价对象极限值或突出其主要因素等场合。相比较而言模式Ⅴ：M（·，+）能够保留单因素评价的全部信息，适用于综合考虑各方面因素影响的场合，在实际工程应用中效果较好，但该模式要求参加评估的权重应具有归一化的属性[17]。综合上述分析，本文选取模式Ⅴ作为合成算子，且在动态权重集的设计上进行了归一化处理。

3.5 单评估原子集失效评估

单评估原子集失效评估是针对单一评估对象 et 的多因素一级模糊综合评价。按照式 (15) 定义，根据 S_{et} 是否为 0，采用不同方式求解 $\underset{\sim}{B}^{[et]}$：

1) $S_{et}=0$ 时，$b_{(j)}^{[et]}=\vec{0}$，“0 项增量”对应分量 $b_\infty^{[et]}$ 直接置 1，即：

$$\underset{\sim}{B}^{[et]}=(b_1^{[et]},b_2^{[et]},\cdots,b_j^{[et]},\cdots,b_m^{[et]},\mid b_\infty^{[et]})=$$
$$\vec{0}\mid 1)=(0,0,\cdots,0\cdots,0\mid 1)$$

2) $S_{et}\neq 0$ 时，加权综合评估对象 et 所辖的全部评价因素在综合评价集 V_s 上的隶属度（需将 V_s 拓展为 $\vec{V}_s^*$），并进行归一化处理，即：

$$\underset{\sim}{B}^{[et]}=\underset{\sim}{A}^{\circ}\underset{\sim}{R}$$

取模式Ⅴ作为合成算子，则有：

$$b_x^{[et]}=\sum_{i=1}^{n}(a_i\cdot r_{ix})=\sum_{i=1}^{n}\left(\left(\frac{FL_{u_i}}{FL_U}\right)\cdot r_{ix}\right)=$$
$$\frac{1}{FL_U}\sum_{i=1}^{n}(FL_{u_i}\cdot r_{ix}) \tag{28}$$

其中：$x=1,2,\cdots,m,\infty$。

按式 (28) 求解出全部 $b_x^{[et]}$ 后，再进行归一化处理则得到最终的 $\underset{\sim}{B}^{[et]}$。

4 系统失效综合评估

单评估原子集失效评估本质上属于一级模糊综合评价。实际情况下，针对某一被测系统不仅需要考虑诸多具有模糊性的评价因素，各评价因素之间往往呈现出不同的层次结构（参见定义 3 所述）。因此对被测系统整体实施系统失效综合评估时，需采用多级模糊综合评价方法。

在图 1 所示的层次化系统结构中，设根节点为 $ES_{root}=[et_{root},U_1]$，其对应的评价因素集 U_1 为：

$$U_1=\{u_1,u_2,\cdots,u_i,\cdots,u_n\}$$

对 U_1 任意分量 $u_i(i=1,2,\cdots,n)$ 可进一步细分：

$$U_i=\{u_{i1},u_{i2},\cdots,u_{ij},\cdots,u_{im}\}$$

对 u_i 任意分量 $u_{ij}(i=1,2,\cdots,n;j=1,2,\cdots,m)$ 继续进行细分：

$$U_{ij}=\{u_{ij1},u_{ij2},\cdots,u_{ijk},\cdots,u_{ijp}\}$$

如此迭代划分，直至各叶子节点。

采用 SFSE 模型对上述层次化多因素系统实施系统失效

综合评估的流程是，按照自下向上的顺序逐层对各层中所有节点依次进行失效评估，最终求解出全系统（顶层根节点）在综合评价集 V_s（需将 V_s 拓展为 $\vec{V}_s^*$）上的隶属度，如图 2 所示。

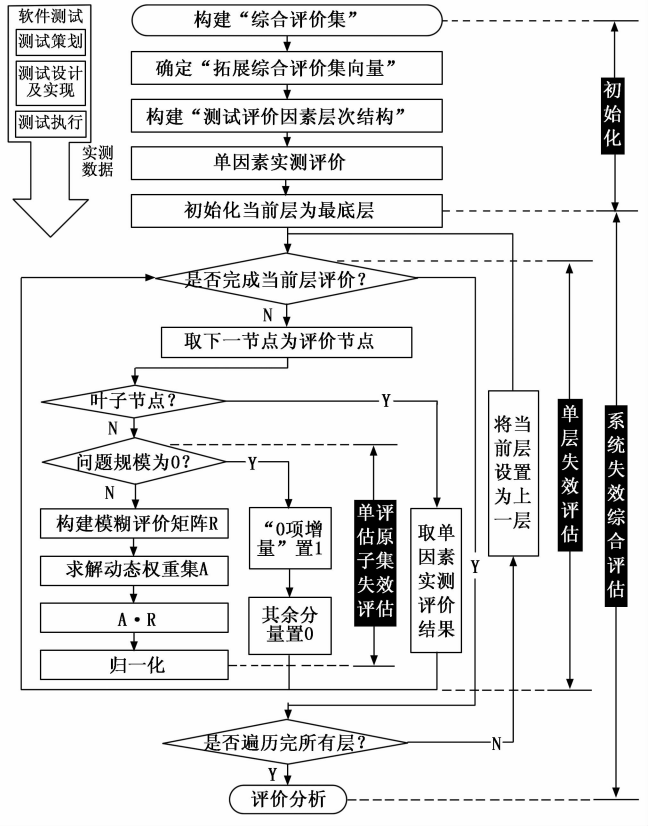

图 2　SFSE 模型（系统失效评估模型）总体流程

5　实验及分析

在某型系统软件测评过程中，以软件测试实测数据作为输入，采用 SFSE 模型实现对被测系统软件质量状态的量化评价。

首先，基于 GJB2786 构建综合评价集 V_s：

$$V_s = V \bigcup \{v_\infty\} = \{v_1, v_2, v_3, v_4, | v_\infty\} =$$
$$\{\text{致命}, \text{严重}, \text{一般}, \text{轻微}, | \text{无问题}\}$$

拓展 V_s，构建成相应的拓展综合评价集向量 $\vec{V}_s^*$：

$$\vec{V}_s^* = \begin{pmatrix} \vec{v}_1^* \\ \vec{v}_2^* \\ \vec{v}_3^* \\ \vec{v}_4^* \\ \vec{v}_\infty^* \end{pmatrix}^T =$$

	[GoF]	[V]	[CoE]	[GoE]	备注
$\vec{v}_1^*$	致命	20	极差	Ⅰ	(10，20]
$\vec{v}_2^*$	严重	10	差	Ⅱ	(5,10]
$\vec{v}_3^*$	一般	5	一般	Ⅲ	(2,5]
$\vec{v}_4^*$	轻微	2	较好	Ⅳ	(0,2]
$\vec{v}_\infty^*$	无问题	0	良好	∞	0

之后，综合分析被测系统功能、技术性能指标以及被测软件各需求项，梳理出被测系统对应的测试评价因素层次结构，如图 3 所示。

软件测试执行完毕后，汇总、分析测试问题，得到所有叶子节点相关实测数据（如表 2 所示），图 3 所述评价因素层次结构中其它非叶子节点的实测数据可由相关叶子节点实测数据汇总得到。

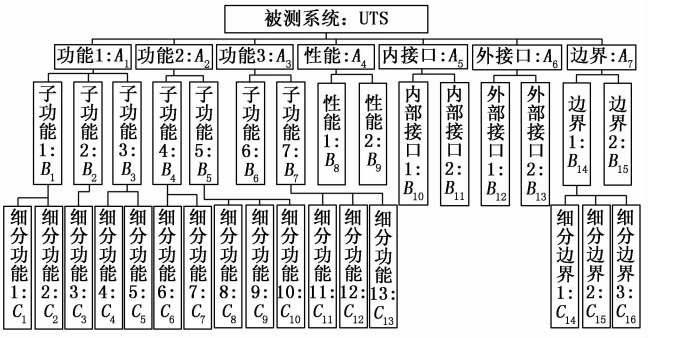

图 3　某型系统软件测试评价因素层次结构示意图

表 2　某型系统软件测试问题分布分析

序号	1	2	3	4	5	6	7	8	1	2	3	4
节点	B_6	B_8	B_9	B_{10}	B_{11}	B_{12}	B_{13}	B_{15}	C_1	C_2	C_3	C_4
致命	0	0	0	0	0	0	0	0	0	0	0	0
严重	2	0	3	1	2	2	3	0	0	0	2	3
一般	1	6	11	7	6	4	7	4	0	0	5	1
轻微	2	12	9	13	14	15	11	18	0	0	1	2
序号	5	6	7	8	9	10	11	12	13	14	15	16
节点	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}
致命	0	0	0	0	0	0	0	0	0	0	0	0
严重	2	0	1	2	0	3	2	0	3	0	0	1
一般	7	8	4	10	7	3	12	1	9	3	0	3
轻微	18	9	10	3	12	13	4	3	2	1	0	19

单因素实测评价所有叶子节点，结果如图 4 所示。

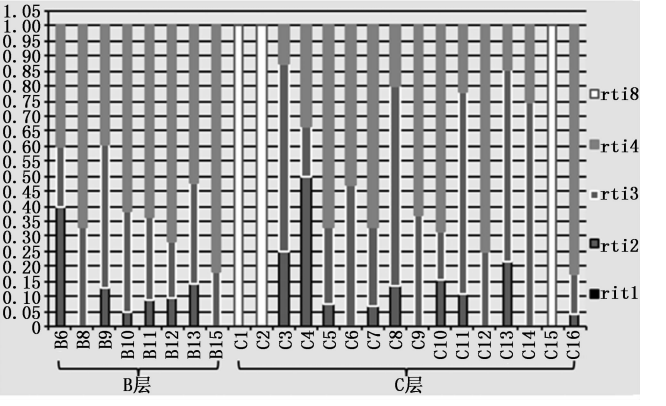

图 4　单因素实测评价结果

自底向上逐层实施失效评估。最底层 C 层所有节点均为叶子节点，B 层中 B_6、B_8、B_9、B_{10}、B_{11}、B_{12}、B_{13}、B_{15} 均为叶子节点，直接置单因素实测评价为相应节点评价结果即可（如图 4、图 5 所示）。

对 B_1 节点有，$S_{B1} = S_{C1} + S_{C2} = 0$，故 $\underset{\sim}{B}^{[B_1]} = 0$。

对 B_2 节点，根据 2.2（2）节介绍的方法求取模糊评价矩阵为：

$$\underset{\sim}{\boldsymbol{R}}_{B_2} = [0 \quad 0.25 \quad 0.625 \quad 0.125 \quad | \quad 0]$$

根据 3.3（3）节介绍的方法，求取 B_2 节点动态权重集：

$$\underset{\sim}{A}_{B_2} = [1]$$

节点 B_2 的单评估原子集失效评估结果为：

$$\underset{\sim}{B}^{[B_2]} = \underset{\sim}{A}_{B_2} \circ \underset{\sim}{R}_{B_2} = [0 \quad 0.25 \quad 0.625 \quad 0.125 \quad | \quad 0]$$

同理，B_5 节点单评估原子集失效评估结果为：

$$\underset{\sim}{B}^{[B_5]} = \underset{\sim}{A}_{B_5} \circ \underset{\sim}{R}_{B_5} = [0.36893 \quad 0.28641 \quad 0.34466] \cdot \left[\begin{array}{cccc|c} 0 & 0.13333 & 0.66667 & 0.2 & 0 \\ 0 & 0 & 0.36842 & 0.63158 & 0 \\ 0 & 0.15789 & 0.15789 & 0.68421 & 0 \end{array}\right] =$$

$$[0 \quad 0.10361 \quad 0.40589 \quad 0.4905 \quad | \quad 0]$$

按照上述方式单评估原子集失效评估 B 层其它所有非叶子节点。最终得 B 层失效评估结果如图 5 所示。

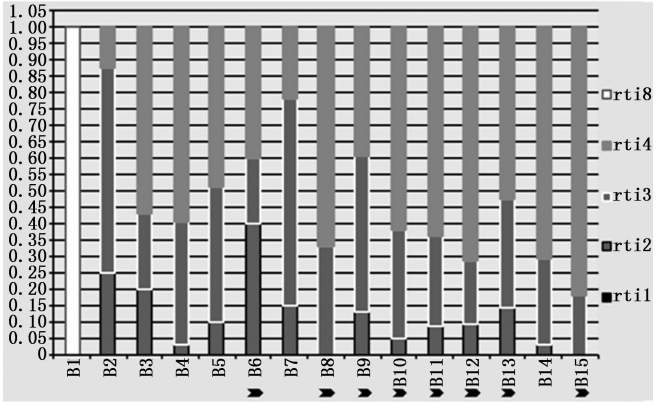

图 5 第 2 层（B 层）失效评估结果

同理可得 A 层失效评估结果如图 6 所示。

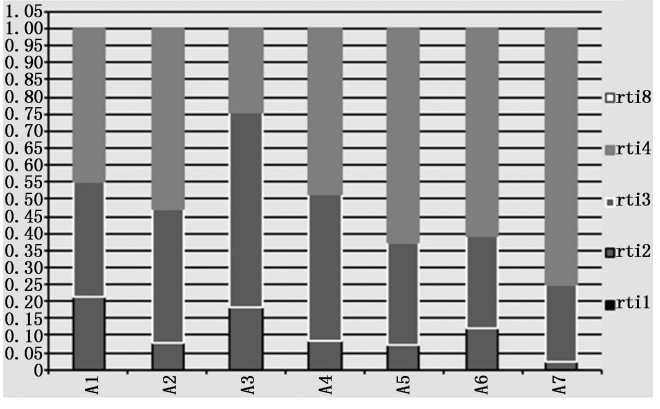

图 6 第 3 层（A 层）失效评估结果

完成根节点 UTS 下所有层节点的失效评估后，对被测系统实施失效评估。求取根节点模糊评价矩阵为：

$$\underset{\sim}{\boldsymbol{R}}_{UTS} = \left[\begin{array}{cccc|c} 0 & 0.21464 & 0.33598 & 0.449390 & 0 \\ 0 & 0.18505 & 0.57006 & 0.24489 & 0 \\ 0 & 0.085572 & 0.42841 & 0.48601 & 0 \\ 0 & 0.070281 & 0.30161 & 0.62811 & 0 \\ 0 & 0.12163 & 0.26964 & 0.60874 & 0 \\ 0 & 0.020141 & 0.22904 & 0.75082 & 0 \end{array}\right]$$

根节点 *UTS* 的动态权重集为：

$$\underset{\sim}{A}_{UTS} = [0.13647 \quad 0.2421 \quad 0.1596 \quad 0.12105 \quad 0.11488 \quad 0.12105 \quad 0.10486]$$

被测系统 *UTS* 的系统失效评估为：

$$\underset{\sim}{B}^{[UTS]} = \underset{\sim}{A}_{UTS} \circ \underset{\sim}{R}_{UTS} =$$

$$[0 \quad 0.11312 \quad 0.37579 \quad 0.5111 \quad | \quad 0]$$

基于拓展综合评价集向量 $\vec{V}_s^*$，采用加权平均的方式求取被测系统 *UTS* 的整体量化评价结果 Val_{UTS}：

$$Val_{UTS} = \underset{\sim}{B}^{[UTS]} \cdot \vec{v}^*[V] = \underset{\sim}{B}^{[UTS]} \cdot \begin{pmatrix} \vec{v}_1^*[V] \\ \vec{v}_2^*[V] \\ \vec{v}_3^*[V] \\ \vec{v}_4^*[V] \\ \vec{v}_{\infty}^*[V] \end{pmatrix} =$$

$$[0 \quad 0.113\,12 \quad 0.375\,79 \quad 0.511\,1 \quad | \quad 0] \cdot \begin{pmatrix} 20 \\ 10 \\ 5 \\ 2 \\ 0 \end{pmatrix} =$$

$$4.032\,3$$

最后根据量化评价获取被测系统 UTS 的评价等级及评价结语：由于 $Val_{UTS} = 4.032\,3 \in (2,5]$，故对 UTS 的评价级别为Ⅲ级，对应的质量状态评价结语为“一般”。

除上述加权平均方法以外，在工程实践中通常亦可采用最大隶属度原则获取最终评价结果，但这种方式丢弃的信息较多，易导致得到不合理结果的情况。相较而言，模糊子集 $\underset{\sim}{B}^{[UTS]}$ 各分量反映了被测系统对综合评价集 V_s 相应分量的隶属度，保留了较多信息，因此在无需单一量化评估的场合，$\underset{\sim}{B}^{[UTS]}$ 可直接作为对被测系统 UTS 系统失效性的定量评价以及对 UTS 质量状态的间接量化评价。

6 结语

SFSE 模型通过引入动态权重集、拓展综合评价因素集等方式，以软件测试所发现的问题规模、问题严重等级作为输入，以拓展综合评价因素集评估量值等分量作为调节手段，充分利用软件测试过程中软件问题确认、软件问题严重等级确定等既有成果，消除了传统 AHP－Fuzzy 模型中人为主观因素的影响。验证实验采用 MATLAB 仿真实现了 SFSE 模型，并在无人工干预的情况下以自动化的方式对被测系统及软件的质量状态实施了量化评估。经实验验证，SFSE 模型适用于软件编程的自动化实现，可行性好，可直接应用于软件测试活动工程化实践，具有较强的实用价值。

后续可基于工程应用实践，不断验证模型评价效果，并对模型进行必要的调整和改进；同时可尝试采用集合论等数学工具描述测试评价因素层次结构、评估对象等要素，对单评估原子集失效度评估与单因素实测评价形式具有一致性（重点是当 $S_{\alpha} \neq 0$ 时，$\underset{\sim}{B}^{[\alpha]}$ 与 $\underset{\sim}{RT}_i$ 具有形式一致性等关键推论进行形式化论证，进一步增强模型在理论上的严谨性。

（下转第 271 页）