文章编号:1671－4598(2022)08－0264－05 DOI:10.16526/j.cnki.11-4762/tp.2022.08.041 中图分类号:TP312 文献标识码:A

软件生命周期质量评价方法研究

李海霞，王 磊，李 智，王月波
（西南电子技术研究所 天奥软件测评中心，成都 610036）

摘要：针对软件质量现存评估标准一方面笼统抽象、主观性强，另外一方面缺乏对软件某个特定阶段质量评估模型的问题，通过对软件生命周期进行研究，对需求分析、软件设计、软件编码和软件测试4个阶段，从bug引入阶段、bug发现阶段、bug缺陷等级、bug数量、bug产生原因、bug修正代价6个方面进行分析，采用了改进加权模糊熵权法确定度量元权重系数，建立软件生命周期不同阶段质量评价模型及软件总体质量评价模型；通过工程实践结果证明，该生命周期质量评价模型能够有效地对软件不同阶段进行评价，为量化软件生命周期不同阶段质量水平提供了一种新方法，减少了软件质量评价中的主观性。

关键词：需求分析；软件设计；软件编码；软件测试；度量元；改进加权模糊熵权法

Research on Software Life Cycle Quality Evaluation Method

LI Haixia，WANG Lei，LI Zhi，WANG Yuebo
（China Electronic Technology Group Corporation Tenth Research Institute，Chengdu 610036，China）

Abstract：The existing evaluation criterias for software quality are more abstract and strong subjective on the one hand，and the quality evaluation model is lack in the specific stage of software on the other hand，through the research of software life cycle，the four stages of requirement analysis，software design，software coding and software testing are studies，including the bug stages of introduction，discovery，defect level，quantity，cause，detection，and so on，the improved weight fuzzy method for entropy weight is used to determine the weight coefficient of measurement element，and the quality evaluation model for different stages of software life cycle and the overall quality evaluation model of software are established. The engineering practice results show that the life cycle quality evaluation model can effectively evaluate the different stages of software，It provides a new method to quantify the quality level in the different stages of software life cycle，and reduces the subjectivity in software quality evaluation.

Keywords：requirement analysis；software design；software coding；software testing；imporved weighted fuzzy weight method

0 引言

随着人们对以软件为核心的载体依赖越来越大，软件产品质量也成为国际社会密切关注的问题，软件质量水平很大程度上决定着软件的市场推广程度及市场地位[1]。但目前软件漏洞众多等问题频频出现，软件质量理所当然地成为了大家关注的焦点，怎么确保被测软件的质量，并进行提高，从而使用户更加满意，已然成为各个国家需要重点解决的热点和难点问题。大部分企业均是依据现有的软件质量相关标准进行质量评估，但是现有的标准并未对软件生命周期某个特定的阶段进行质量评估，只是从概念上非常笼统的抽象出通用模型[2－5]，特定阶段软件质量评估和实施在我国尚未成熟，并且评估技术大多来自国外软件质量相关标准，几乎没有自主产权的有特色的评估体系[6－9]。另外，对于软件生命周期中和软件的质量问题最为密切相关的几个阶段，包括需求分析、软件设计、软件编码和软件测试，一方面缺乏对定性和定量指标的提取，另一方面，也未形成科学严谨的阶段质量评价模型。因此迫切需要制定一套专门针对软件生命周期中需求分析、软件设计、软件编码和软件测试的质量评价方法，该评价方法具有非常重大的现实意义。

1 软件生命周期概述

和世界万物类似，一个成熟的软件需要经历孕育、诞生、成长、成熟、衰亡等阶段，一般我们称之为软件生存周期（又叫软件生命周期）[10]。一般情况下，软件生命周期主要包括软件需求分析、软件设计、软件编码、软件测试和软件维护等阶段，软件测试[11]是一个系列过程，包括软件测试需求分析、软件测试计划设计、软件测试用例设计、软件测试执行等等，软件生命周期各个阶段均进行不同目的和内容的测试活动，因此，软件项目整个生命周期中均存在软件测试，从而确保软件生命周期各个阶段均正常使用。此外，软件生命周期各个过程均可能存在错误，软件测试只能确认软件存在的错误，并不能避免软件新错误的出现。

在整个软件生命周期中，其中软件需求分析、软件设计、软件编码和软件测试等4个阶段和软件质量息息相关。软件需求分析是对产品进行定义，软件设计是进行产品设

收稿日期:2022－03－03； 修回日期:2022－03－31。
作者简介:李海霞(1987－)，女，河南省周口市人，硕士，高级工程师，主要从事软件工程化、自动化测试技术方向的研究。
引用格式:李海霞，王 磊，李 智，等. 软件生命周期质量评价方法研究[J]. 计算机测量与控制，2022，30(8)：264－268，295.

计，软件编码是实现软件，软件测试则是验证软件是否满足需求。软件的质量控制有个很明显的特点，那就是发现软件问题越早，解决该软件问题的成本就会越低，反之则越高，这是因为软件问题可能出现在软件需求分析、软件设计和软件编码、软件测试这几个阶段的任意阶段，而在实际软件生命周期中，软件问题很多是等到了后面阶段才被发现，但此时很多工作已经基于这个错误的前提下进行了开展，而且引入问题的阶段越是靠前，其影响范围也就会越广，而且解决软件质量问题的成本是和开发阶段呈指数级别增长的，所以越是到后面，需要投入的人力成本和时间等成本也就会越多。因此，对软件生命周期每个阶段分别进行评价，建立合适的软件生命周期质量评价方法就尤其重要。

2　软件生命周期质量评价模型

2.1　软件生命周期质量评价流程

软件缺陷（bug）：软件在使用过程中存在的任何问题都称为软件的缺陷，简称 bug。

bug 引入阶段指的是在整个软件生命周期中，哪个阶段产生的 bug。而在该模型中，bug 引入阶段包括软件需求分析阶段、软件设计阶段、软件编码阶段和软件测试阶段。需求阶段可能因为需求描述不易理解，有歧义、错误等；设计阶段可能因为设计文档存在错误或者缺陷等；编码阶段可能开发人员代码出现错误等；测试阶段可能测试人员操作错误等引入了 bug，也就是说软件生命周期任何阶段都可能引入 bug。

bug 发现阶段指的是在整个软件生命周期中，哪个阶段发现的 bug。而在该模型中，bug 发现阶段包括软件需求分析阶段、软件设计阶段、软件编码阶段和软件测试阶段。bug 发现阶段一般晚于 bug 引入阶段，一般来说，bug 发现的越早，修复的成本越低，反之，bug 发现的越晚，修正该 bug 所付出的代价就越高，而且修复代价呈指数级增长。

bug 缺陷等级：按照 bug 对被测系统的影响程度，可分为关键缺陷、严重缺陷、一般缺陷、建议改进。

关键缺陷：非常严重的缺陷，比如软件或系统瘫痪、异常退出、频繁的死机，软件或系统重要部件无法正常运行，从而严重影响软件或系统功能的正常运行，均定义为关键缺陷。

严重缺陷：严重地影响软件或系统要求，或未实现基本功能，主要功能无法执行、或数据被破坏、或产生错误结果，而且是在用户常规操作中频繁出现，或者非常规操作中无法避免的主要问题，软件或系统无法满足重要的业务需求，或错误设计配置项或配置项后测试中发现配置有关的问题等。

一般缺陷：软件或系统基本满足用户业务要求，但次要功能丧失，或通过变通手段可以解决，或安装手册或部署文档错误而导致安装部署软件失败，或对应的业务流程功能未实现，但有替代方法来解决，或系统响应时间变慢、产生错误的中间结果但不影响实际使用等影响有限的问题。包括性能问题、安全问题、校验问题、乱码等。

建议改进：使操作者不方便或操作麻烦，但不影响执行工作功能或重要功能。包括界面问题、提示信息、易用性、统一性等，希望提出的建议但不强制进行的修改。

bug 数量：该评价模型中，指的是确认修改的 bug 数量，撤回的 bug 数量不计入该模型。

bug 产生原因：通过查阅某测评中心近 5 年的测试报告记录，提炼总结出常见的缺陷产生原因，并根据软件生命周期不同阶段分别划分，具体 bug 产生原因详见表 1 所示。

表 1　软件生命周期不同阶段 bug 产生原因度量元

软件生命周期阶段	bug 产生原因
需求分析	需求二义性
	需求错误
	需求不一致
	纯粹性错误
	需求遗漏
设计	设计错误
	设计不一致
	纯粹性错误
	设计遗漏
编码	开发工具本身问题
	编码遗漏
	编码错误
测试	测试环境配置错误
	软件版本部署错误
	测试标准不恰当
	测试遗漏

度量元：通常用来对软件或系统进行量化管理时，需要重点关注信息对象的基本属性的描述。依据度量数据获取方式，可以把度量元划分为基本度量元（又称为直接度量元）和派生度量元（又称为间接度量元）两种。基本度量元，顾名思义，其数据来源可直接度量获得，派生度量元，其数据来源于其他的数据，一般由两个或多个基本度量元组合获得。在该评价模型中，为了量化软件生命周期不同阶段质量，采取基本度量元方式，以下简称度量元。

熵权法：是统计学领域一种客观赋权方法，在统计学领域中，但数据越分散，熵值越小，可以认为该数据包含信息越多，因此权重越大。在具体使用过程中，根据各指标的数据的分散程度，利用信息熵计算出各指标的熵权，再根据各指标对熵权进行一定的修正，从而得到较为客观的指标权重。

为了能够客观地评价软件生命周期不同阶段质量，针对需求分析、软件设计、软件编码和软件测试这 4 个阶段，分别从 bug 引入阶段、bug 发现阶段、bug 缺陷等级[12]、bug 数量、bug 产生原因、bug 修正代价 6 个方面，筛选体

现每个阶段质量的度量元，对每个度量元进行定义，并采用改进的加权模糊熵权法[13-17]确定每个度量元的权重系数，减少主观因素干扰，提高度量元权重系数客观性，从而建立起一套行之有效的计算方法和评价机制。

该评价模型主要分以下几步。第 1 步：提取度量元，以 bug 引入阶段、bug 发现阶段、bug 缺陷等级、bug 数量、生命周期不同阶段 bug 产生原因、bug 修正系数 6 个方面作为度量元；第 2 步：bug 产生原因度量元权重系数计算，为减少主观因素干扰，采用改进的加权模糊熵权法确定权重系数；第 3 步：需求阶段质量评价；第 4 步，设计阶段质量评价；第 5 步：编码阶段质量评价；第 6 步：测试阶段质量评价；第 7 步，总体质量评价。整个评价流程如图 1 所示。

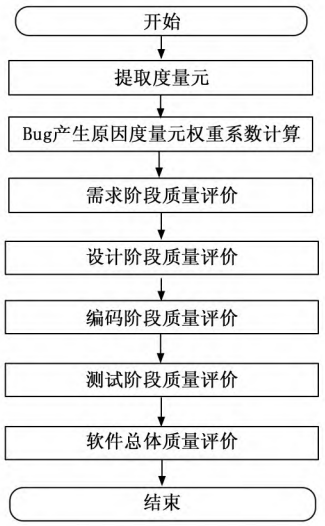

图 1 软件生命周期质量评价流程

2.2 提取度量元

软件生命周期质量评价方法中，以 bug 引入阶段、bug 发现阶段、bug 缺陷等级、bug 数量、生命周期不同阶段 bug 产生原因、bug 修正系数 6 个方面作为度量元，并建立需求阶段 bug 产生原因度量元集合 $R=\{e_1, e_2, \ldots, e_i\}$，设计阶段 bug 产生原因度量元集合 $D=\{d_1, d_2, \ldots, d_j\}$，编码阶段 bug 产生原因度量元集合 $C=\{c_1, c_2, \ldots, c_j\}$，测试阶段 bug 产生原因度量元集合 $T=\{t_1, t_2, \ldots, t_j\}$。

2.3 Bug 产生原因度量元权重系数计算

为了减少主观因素干扰，提高度量元权重系数客观性，采用改进的加权模糊数熵权法，首先应用模糊数来表示多名专家对 bug 产生原因度量元做出的评价结果，建立直觉模糊集合。

直觉模糊集的一些基本概念：设 X 为一给定的论域，则称 $A=\{<\chi i, \mu A(xi), \nu A(\chi i)>|\chi i\in X\}$ 为 X 上的一个直觉模糊集。其中 $\mu A: X\rightarrow[0,1]$ 和 $\nu A: X\rightarrow[0,1]$ 分别为 A 的隶属度和非隶属度函数，且对于任意的 $\chi i\in X$，有 $0\leqslant\mu A(\chi i)+\nu A(\chi i)\leqslant 1$ 成立。进一步，称 $\pi A(\chi i)=1-\mu A(xi)-\nu A(\chi i)$ 为模糊集 A 中元素 χi 的犹豫度，$a=(\mu a, \nu a)$ 是直觉模糊数，且满足 $0\leqslant\mu a\leqslant 1$，$0\leqslant\nu a\leqslant 1$[9-10]；专家分别对不同阶段 bug 产生原因度量元进行评价，采集到需求阶段 bug 产生原因度量元直觉模糊数，设计阶段 bug 产生原因度量元模糊数，编码阶段 bug 产生原因度量元模糊数，测试阶段 bug 产生原因度量元模糊数，然后采用加权模糊数熵权法确定不同阶段 bug 产生原因度量元的权重，加权模糊熵权法计算公式为：

假设 $X=\{x1, x2, xi, xn\}$，$A=\{<\chi i, \mu A(xi), \nu A(\chi i)>|\chi i\in X\}$ 为专家给出的模糊集，则模糊集的熵计算公式为：

$$E(A)=\frac{1}{n}*\sum_{j=1}^{n}\frac{1-|\mu_A(x_j)-\nu_A(\chi_j)|+\pi_A(\chi_j)}{1+|\mu_A(x_j)-\nu_A(\chi_j)|+\pi_A(\chi_j)}\quad(1)$$

考虑到 bug 产生原因权重信息的客观性，减少专家判断的主观性，以 bug 产生原因问题个数占比作为加权系数，提出 bug 产生原因度量元权重公式为：

$$W_i=\frac{n_m}{N_m}*\frac{1-E(A_i)}{\sum_{j=1}^{n}(1-E(A_j))}$$

式中，n_m 为软件生命周期不同 bug 产生原因度量元对应的问题数；N_m 为软件生命周期不同 bug 产生原因度量元对应的问题总数。

进行归一化处理，最终 bug 产生原因度量元权重计算公式为：

$$\lambda_i=\frac{W_i}{\sum_{i=1}^{n}W_i}\quad(3)$$

2.4 需求阶段质量评价

需求阶段：即软件需求分析阶段，是开发人员经过深入细致的调研和分析，准确理解用户和项目的功能、性能、可靠性等具体要求，将用户非形式的需求转述为完整的需求定义，从而确定系统必须做什么的过程。

该模型中，从两个维度对需求阶段质量减分评价

维度 1：bug 引入阶段＋bug 等级＋bug 数量，在 bug 引入阶段是需求阶段时，提出需求阶段质量评价第一个减分项公式

$$R_1=\sum_{i=1}^{n}V_i*RK_i\quad(4)$$

式中，V_i 为 bug 缺陷等级权重系数，且满足 $\sum_{i=1}^{n}V_i=1$；n 为 bug 缺陷等级总分类数；RK_i 为需求阶段引入的不同缺陷等级对应的 bug 缺陷数量

维度 2：bug 引入阶段＋bug 数量＋bug 产生原因，在 bug 引入阶段是需求阶段时，提出需求阶段质量评价第二个减分项

$$R_2=\sum_{k=1}^{rn}RS_k*RM_k\quad(5)$$

式中，RS_k 为需求阶段 bug 产生原因权重系数；rn 为需求阶段 bug 产生原因总分类数；RM_k 为需求阶段 bug 产生原因对应的 bug 缺陷数量。

从而定义需求阶段质量评价减分项 $R=R_1+R_2$，R 越大，表明需求阶段质量越不好，在后续项目中越需要加强需求人员的能力。

2.5　设计阶段质量评价

设计阶段：即软件设计，是从软件需求规格说明书出发，根据需求分析阶段确定的功能设计软件系统的整体结构、划分功能模块、确定每个模块的实现算法以及编写具体的代码，形成软件的具体设计方案。

该模型中，从两个维度对设计阶段质量进行减分评价

维度 1：bug 引入阶段＋bug 等级＋bug 数量，bug 引入阶段是设计阶段时，设计阶段质量评价第一个减分项：

$$D_1=\sum_{i=1}^{n}V_i*DK_i \tag{6}$$

式中，V_i 为 bug 缺陷等级权重系数，且满足 $\sum_{i=1}^{n}V_i$；n 为 bug 缺陷等级总分类数；DK_i 为设计阶段引入的不同缺陷等级对应的 bug 缺陷数量。

维度 2：bug 引入阶段＋bug 数量＋bug 产生原因，设计阶段质量评价第二个减分项：

$$D_2=\sum_{k=1}^{dn}DS_k*DM_k$$

式中，DS_k 为设计阶段 bug 产生原因权重系数；dn 为设计阶段 bug 产生原因总分类数；DM_k 为设计阶段 bug 产生原因对应的 bug 缺陷数量。

从而定义阶段质量评价减分项 $D=D_1+D_2$，D 越大，表明设计阶段质量越不好，在后续项目中越需要加强设计人员的能力；

2.6　编码阶段质量评价

编码阶段：即软件编码，是将设计阶段得到详细设计转换为基于某种计算机语言的程序，即源程序代码。

该模型中，从两个维度对编码阶段质量进行减分评价

维度 1：bug 引入阶段＋bug 等级＋bug 数量，bug 引入阶段是编码阶段时，编码阶段质量评价第一个减分项公式：

$$C_1=\sum_{i=1}^{n}V_i*CK_i \tag{8}$$

式中，V_i 为 bug 缺陷等级权重系数，且满足 $\sum_{i=1}^{n}V_i=1$；n 为 bug 缺陷等级总分类数；CK_i 为设计阶段引入的不同缺陷等级对应的 bug 缺陷数量。

维度 2：bug 引入阶段＋bug 数量＋bug 产生原因，bug 引入阶段是设计阶段时，编码阶段质量评价第二个减分项公式：

$$C_2=\sum_{k=1}^{cn}CS_k*CM_k \tag{9}$$

式中，CS_k 为编码阶段 bug 产生原因权重系数，且满足 $\sum_{k=1}^{cn}CS_k=1$；cn 为编码阶段 bug 产生原因总分类数；CM_k 为编码阶段 bug 产生原因对应的 bug 缺陷数量。

从而定义编码阶段质量评价减分项 $C=C_1+C_2$，C 越大，表明编码阶段质量越不好，在后续项目中越需要加强编码人员的能力；

2.7　测试阶段质量评价

测试阶段：即软件测试，描述一种用来促进鉴定软件的正确性、完整性、安全性和质量的过程。换句话说，软件测试是一种实际输出与预期输出之间的审核或者比较过程。软件测试的经典定义是：在规定的条件下对程序进行操作，以发现程序错误，衡量软件质量，并对其是否满足设计要求进行评估的过程。

测试阶段质量评价不同于其他阶段，本发明中，对于 bug 发现阶段晚于 bug 引入阶段的情况，测试阶段质量评价中也需要扣分。统计资料显示在需求阶段修正一个错误的代价假如是 1，则在设计阶段就是它的 2～3 倍，在编码阶段就是它的 5～8 倍，在测试阶段就是它的 20～30 倍[18-19]，修正错误的代价不是随时间线性增长，而几乎是呈指数增长的，软件的质量问题越到后面解决成本越高。根据修正错误的代价大小，确定需求阶段错误修正代价系数是 1，设计阶段错误修正代价系数是 e，编码阶段错误修正代价系数是 e^2，测试阶段错误修正系数是 e^3，归一化处理后各个阶段 bug 修正代价权重系数如表 2 所示。

表 2　软件生命周期不同阶段 bug 修正代价系数

阶段名称	需求分析	设计	编码	测试
Bug 修正代价系数	0.49	0.27	0.15	0.09

该模型中，从 3 个维度进行测试阶段质量减分评价。

维度 1：bug 引入阶段＋bug 等级＋bug 数量，bug 引入阶段是测试阶段时，测试阶段质量评价第一个减分项公式：

$$T_1=\sum_{i=1}^{n}V_i*TK_i \tag{10}$$

式中，V_i 为 bug 缺陷等级权重系数，且满足 $\sum_{i=1}^{n}V_i$；n 为 bug 缺陷等级总分类数；TK_i 为测试阶段引入的不同缺陷等级对应的 bug 缺陷数量。

维度 2：bug 引入阶段＋bug 数量＋bug 产生原因，bug 引入阶段是测试阶段时，测试阶段质量评价第二个减分项公式：

$$T_2=\sum_{k=1}^{tn}TS_k*TM_k \tag{11}$$

式中，TS_k 为测试阶段 bug 产生原因权重系数，且满足 $\sum_{k=1}^{tn}TS_k=1$；tn 为编码阶段 bug 产生原因总分类数；TM_k 为测试阶段 bug 产生原因对应的 bug 缺陷数量。

维度 3：bug 引入阶段＋bug 发现阶段＋bug 修正系数，bug 引入阶段是测试阶段时，测试阶段质量评价第三个减分项：

$$T_3=\sum_{k=1}^{n}(Y_k+1-F_k)*P \tag{12}$$

式中，Y_k 为 bug 引入阶段，本发明中把需求阶段定义为 1，设计阶段定义为 2，编码阶段定义为 3，测试阶段定义为 4；F_k 为 bug 发现阶段，本发明中均定为测试阶段，$n=4$；P 为 bug 引入阶段对应的修正系数，参见表 2。

从而定义测试阶段质量评价减分项 $T=T_1+T_2+T_3$，

T 越大，表明测试阶段质量评价越不好，在后续项目中越需要加强测试人员的能力。

2.8 总体质量评价

总体质量减分评价公式

$$S = R + D + C + T \tag{13}$$

式中，R 为需求阶段质量减分评价；D 为设计阶段质量减分评价；C 为编码阶段质量减分评价；T 为测试阶段质量减分评价。S 越大，表明该软件总体质量越差。

3 测试结果与分析

从某具有丰富经验的 XX 测评中心机构中，查找近 5 年测试报告记录，从中筛选中各个阶段 bug 产生原因，定义软件生命周期不同阶段度量元详见表 2。

首先邀请 5 位具有专业知识的软件测评专家，分别对需求分析、设计、编码和测试阶段 bug 产生原因进行评价，为了对不同阶段 bug 产生原因之间的重要程度进行定量的描述，定义如表 3 所示的标度[20]，括号 3 个数字代表（μ_{ij}，ν_{ij}，π_{ij}）。

表 3 bug 产生原因重要程度定义的标度表

bug 产生原因相对重要性说明	bug 产生原因相对重要性程度标度
指标 i 与指标 j 相比极其重要	(0.9,0.1,0.0)
指标 i 与指标 j 相比特别重要	(0.8,0.1,0.1)
指标 i 与指标 j 相比重要很多	(0.7,0.2,0.1)
指标 i 与指标 j 相比明显重要	(0.6,0.3,0.1)
指标 i 与指标 j 相比稍微重要	(0.5,0.4,0.1)
指标 i 与指标 j 相比同等重要	(0.5,0.5,0.0)
指标 j 与指标 i 相比极其重要	(0.1,0.9,0.0)
指标 j 与指标 i 相比特别重要	(0.1,0.8,0.1)
指标 j 与指标 i 相比重要很多	(0.2,0.7,0.1)
指标 j 与指标 i 相比明显重要	(0.3,0.6,0.1)
指标 j 与指标 i 相比稍微重要	(0.4,0.5,0.1)
指标 j 与指标 i 相比同等重要	(0.5,0.5,0.0)

根据表 3，各位专家两两比较软件生命周期不同阶段 bug 产生原因重要程度，并以某 XX 军用模拟训练系统中系统数据管理软件为例，其不同阶段对应问题数如表 4 所示。

以专家给出的直觉模糊数为基础，利用公式（1）～（3）计算软件不同阶段 bug 产生原因权重系数，如表 5 所示。

利用公式（4）～（13），分别计算软件生命周期不同阶段质量减分评价，如表 6 所示。

根据公式（14），计算该软件总体质量评价

$$S = 2.43 + 3.57 + 11.72 + 6.658 = 24.378$$

从表 6 中可以看出，被评软件中需求分析阶段评分结果为 2.43，设计阶段评分结果为 3.57，编码阶段评分结果为 11.72，测试阶段评分结果为 6.658，软件总体质量评分结果是 24.378。该软件生命周期不同阶段质量优劣顺序为需求分析阶段＞设计阶段＞测试阶段＞编码阶段，故对于项目组管理人员来说，需要重点关注编码阶段质量，提高编码人员能力。经对数据做人工分析，可以看出评价的结果和新设计的计算准则是基本吻合。

表 4 软件不同阶段对应问题数

阶段	bug 原因	问题数	bug 等级			
			关键	严重	一般	建议
需求分析	需求二义性	2	0	0	1	1
	需求错误	1	0	1	0	0
	需求不一致	1	0	0	1	0
	纯粹性错误	1	0	0	1	0
	需求遗漏	1	0	1	0	0
设计	设计错误	2	0	0	2	0
	设计不一致	3	0	1	1	1
	纯粹性错误	1	0	0	0	1
	设计遗漏	2	0	1	1	0
编码	开发工具问题	3	0	1	2	0
	编码遗漏	2	0	0	2	0
	编码错误	11	0	3	7	1
测试	环境配置错误	1	0	1	0	0
	版本部署错误	5	0	1	3	2
	标准不恰当	2	0	0	1	0
	测试遗漏	2	0	0	2	0

表 5 软件不同阶段 bug 产生原因权重系数

阶段	bug 原因	权重系数
需求分析	需求二义性	0.33
	需求错误	0.18
	需求不一致	0.18
	纯粹性错误	0.14
	需求遗漏	0.17
设计	设计错误	0.22
	设计不一致	0.38
	纯粹性错误	0.11
	设计遗漏	0.29
编码	开发工具问题	0.16
	编码遗漏	0.10
	编码错误	0.74
测试	环境配置错误	0.07
	版本部署错误	0.54
	标准不恰当	0.21
	测试遗漏	0.18

表 6 软件不同阶段质量评价结果

阶段	需求分析	设计	编码	测试
评分	2.43	3.57	11.72	6.658

4 结束语

目前的软件质量评估标准一方面只是从概念上定义了一个笼统抽象的通用模型来评估整个软件的质量，另外一方

（下转第 295 页）

效应敏感外设及其损伤剂量的概率模型分析 [J]. 核技术，2021，44 (3)：63－68.
[11] 王恩美，邬树楠，吴志刚. 在轨组装空间结构面向主动控制的动力学建模 [J]. 力学学报，2020，52 (3)：805－816.
[12] 肖　帆，李　光，游雨龙. 空间 3R 机械手逆向运动学的多模块神经网络求解 [J]. 中国机械工程，2019，30 (10)：1233－1238.
[13] 刘东亮，王军光，张　洁，等. 基于知识单元挖掘的网络文库信息存储模型研究 [J]. 情报学报，2020，39 (2)：171－177.
[14] 黄正峰，李雪健，鲁迎春，等. 65 nm CMOS 工艺的低功耗加固 12T 存储单元设计 [J]. 计算机辅助设计与图形学学报，2019，31 (3)：504－512.
[15] 张　旭，陈爱军，沈小燕，等. 基于线激光传感器的工件尺寸测量系统的误差补偿方法 [J]. 计量学报，2020，41 (12)：1449－1455.
[16] 尤晶晶，符周舟，李成刚，等. 并联式六维加速度传感器的解耦参数辨识及其扰动分析 [J]. 振动与冲击，2019，38 (1)：134－141.
[17] 贾　康，洪　军，张银行. 一种拉刀螺旋容屑槽前刀面磨削砂轮安装位姿计算方法 [J]. 机械工程学报，2019，55 (11)：205－214.
[16] 曾祥君，陈　磊，喻　锟，等. 基于配电网双端信息融合的单相断线故障实时监测方法 [J]. 电力科学与技术学报，2020，35 (3)：12－18.
[17] 王吉岱，徐东晓，孙爱芹，等. 基于多传感器信息融合的输电线路巡检机器人自主越障方法研究 [J]. 机床与液压，2020，48 (9)：24－28.
[18] 杨克克，罗　阳，赵忆文，等. 基于主方向傅里叶变换算子的 2D/3D 分级配准 [J]. 机器人，2021，43 (3)：296－307.
[19] 李　东，张成祥，赵　迪，等. 基于伪逆极坐标傅里叶变换的快速 ISAR 方位定标 [J]. 电子与信息学报，2019，41 (2)：262－269.
[20] 郑钰馨，奚　鹰，李梦如，等. 旋转向量减速器纯扭转模型固有频率和灵敏度分析 [J]. 吉林大学学报（工学版），2019，49 (2)：499－512.
[21] 肖延辉，田华伟，张永胜. 结合深度迭代缩放卷积神经网络的 PRNU 提取算法 [J]. 信号处理，2020，36 (9)：1582－1589.
[22] 刘　未，朱宏辉. 隧道安全预警机器人自主导航方法研究 [J]. 计算机测量与控制，2021，29 (8)：172－177.
[23] 黄剑雄，刘小雄，章卫国，等. 基于视觉/惯导的无人机组合导航算法研究 [J]. 计算机测量与控制，2021，29 (2)：137－143，149.

（上接第 268 页）

面缺乏针对软件生命周期不同阶段质量评估模型。针对这些问题，对软件生命周期中和软件质量问题最为密切相关的几个阶段，包括需求分析、软件设计、软件编码和软件测试，制定了新的软件生命周期质量评价方法，分析度量元，并提出改进的加权模糊熵权法确定加权系数，给出评价方法和评价流程，最后通过实例对方法进行了有效验证，为软件生命周期不同阶段质量建立了一套有效的评价标准。当然，评价标准需要经过实践不断验证和不断的改良，评价度量元和加权系数也是需要在实践中不断的验证而修改和补充。

参考文献：

[1] 崔潇丹. 领域软件的质量评估方法研究与应用 [D]. 上海：东华大学，2018.
[2] PRESSMAN R S. Software engineering a practitioner’s approach [M]. London：HMCGraw-Hill International Computer Science，1992：20－30.
[3] COMFORD J，STONE J. Engineering safety [M]. London：HMCGraw-Hill Book Company，1992.
[4] HON B B W. Idetifying quality requiems conflicts [J]. IEEE Software，1996，13 (2)：23－35.
[5] BOEHM. Software risk management：principles and practices [J]. IEEE Software，1991，8 (1)：32－41.
[6] 刘宇柯. 基于 CMMI 的软件开发项目管理研究 [D]. 广州：广东工业大学，2015.
[7] 齐　轶，王　勇. 基于层级属性关系软件评估模型 [J]. 计算机测量与控制，2017，25 (6)：20－20.
[8] 吴　坚，吴　刚. 软件质量模型的研究 [J]. 计算机工程与科学，2006，28 (8)：125－127.
[9] 钱鸿生. 基于风险管理的软件生命周期模型研究 [D]. 上海：同济大学，2006.
[10] 刘玉军，冯　飞，曹　乐. 一种航空机载嵌入式软件安全性评价方法研究 [J]. 计算机测量与控制，2020，28 (3)：2－3.
[11] 刘　涛，李　娜. 航空机载软件测试质量评价方法研究 [J]. 计算机测量与控制，2018，26 (11)：285－287.
[12] 邢薇薇，王新刚. 航空机载软件缺陷分类方法研究与应用 [J]. 测控技术，2016，35 (9)：102－106.
[13] 刘　宏，李好威. 基于熵权法与灰色关联分析的 WSNs 路由算法 [J]. 传感器与微系统，2017，36 (8)：1－2.
[14] 刘培德，王娅姿. 一种属性权重未知的区间概率风险型混合多属性角色方法 [J]. 控制与决策，2012，27 (2)：276－280.
[15] 上官延华，冯荣耀，柳宏川. 一种基于熵和均方差法综合赋权的 K-means 算法 [J]. 计算机与现代化，2010 (4)：34－36.
[16] 赵　萌，任峥嵘，邱菀华. 基于直觉模糊熵的专家权重确定方法及其验证 [J]. 控制与决策，2015，7 (30)：2－3.
[17] 刘满凤，任海平. 基于一类新的直觉模糊熵的多属性决策方法研究 [J]. 系统工程理论与实践，2015，35 (11)：2－3.
[18] 付剑平，陆民燕. 基于模糊综合评价的软件测试性度量 [J]. 计算机工程与应用，2009，45 (27)：1－2.
[19] 权　岩. 基于 CMMI 模型的软件工程优化研究 [D]. 南京：南京邮电大学，2014.
[20] 张　黎. 基于直觉模糊层次分析法的民航机场服务质量评价 [P]. 郑州：郑州航空工业管理学院，2020，1：7－8.