

基于模糊层次分析法的软件易用性 过程综合评价方法^{*}

赵倩^a, 张璇^{a,b†}, 王黎霞^c, 刘聪^a, 杨帅^a

(云南大学 a. 国家示范性软件学院; b. 云南省软件工程重点实验室; c. 经济学院, 昆明 650091)

摘要: 对软件易用性过程进行综合评价是开发可信软件产品的必要因素。通过构建软件易用性生命周期模型和软件易用性度量模型搭建软件易用性过程综合评价模型,通过此模型对软件开发中的易用性问题进行有效的指导,规范易用性开发过程,从而保证能够产出高易用性的产品;利用模糊层次分析法对此模型进行度量评价,量化评估易用性过程优劣;通过一个实例分析说明该方法可以有效地应用于软件开发过程中,能够尽早地确保软件产品的易用性,为开发高易用度的软件产品指明了方向。

关键词: 软件过程; 易用性; 模糊层次分析法

中图分类号: TP302.7

文献标志码: A

文章编号: 1001-3695(2013)09-2730-06

doi:10.3969/j.issn.1001-3695.2013.09.043

Evaluation method of software usability process based on fuzzy analytic hierarchy process

ZHAO Qian^a, ZHANG Xuan^{a,b†}, WANG Li-xia^c, LIU Cong^a, YANG Shuai^a

(a. School of Software, b. Key Laboratory of Software Engineering of Yunnan Province, c. School of Economics, Yunnan University, Kunming 650091, China)

Abstract: The comprehensive evaluation of software usability process is necessary factors to develop credible software products. First, this paper constructed the software usability process comprehensive evaluation model by software life-cycle model and software usability measurement model. It used this model to guide and regulate the software usability development process. In that case it could be guaranteed to produce a high usability product. Secondly, it used the fuzzy analytic hierarchy process (FAHP) to measure and evaluate this model. Finally through an example shows that this method can effectively apply in the software development process, and early to ensure the usability of software in the development process. And this method indicates the direction of developing high usability software.

Key words: software process; usability; fuzzy analytic hierarchy process (FAHP)

0 引言

软件的易用性是指特定用户对于软件的易理解性、易学性、可操作性,以及此系统或者产品对于用户的吸引程度。ISO9241-210《人机交互设计指导国际标准》^[1]中定义易用性是指特定产品在特定的使用环境下被特定的用户用于特定用途时所具有的有效性、效率和用户主观满意度。它衡量的是一个产品是否符合用户的使用喜好,是一个产品能否立足市场的核心竞争力,也是软件可信中的一个关键属性。

20世纪80年代以来软件易用性的重要性已经被国外的一些企业所关注。IBM早在1970年就引入了易用性测试,Microsoft在1988年也开始了易用性测试。从90年代开始,易用性在美、欧等国家的软件及IT行业迅速普及^[2,3]。目前,世界

上主要的软件及IT企业都成立了专门的易用性部门,建立了自己的易用性规范,许多互联网公司也有专门的易用性人员。国际上也制定了一系列易用性及易用性相关的国际标准和工业标准,如ISO9241^[1]、ISO25010^[4]等,并且这些规范标准已经被业界广泛地接受和认可,易用性对于一个产品或系统的重要性可见一斑。

以人为中心的设计(human-centred design, HCD)是目前业界常用的判断软件易用性的方法。其强调以人为中心进行开发,以达到较高易用性——增加用户的接受度和满意度。ISO9241推荐用HCD的设计思想贯穿于整个项目的生命周期,从概念定义、分析、设计、执行、测试到维护。

然而,纵观目前业界对于易用性的研究及其应用不难发现:

收稿日期: 2012-12-07; **修回日期:** 2013-02-20 **基金项目:** 国家自然科学基金资助项目(61262025, 61262024); 云南省应用基础研究计划面上项目(2012FB118); 云南省教育厅科学研究基金资助项目(2012Y257); 云南省软件工程重点实验室开放基金资助项目(2011SE09); 2012年第五届云南大学研究生科研课题资助项目(yunuy65)

作者简介: 赵倩(1989-),女,山西晋中人,硕士研究生,主要研究方向为软件过程、可信软件;张璇(1978-),女,江苏南京人,副教授,博士研究生,主要研究方向为信息安全、软件工程(zhxuan@ynu.edu.cn);王黎霞(1962-),女,云南昆明人,副教授,硕士,主要研究方向为软件工程;刘聪(1989-),男,重庆人,硕士研究生,主要研究方向为Web应用、软件工程;杨帅(1987-),男,陕西西安人,硕士研究生,主要研究方向为软件过程、软件演化。

a) 易用性概念模糊。虽然各个国际标准中都对易用性进行了定义,但学术界和业界对易用性的定义也不统一,不同领域的专家和组织给出了不同维度和标准^[5]。易用性概念本身的模糊性和情境依存性依然存在。

b) 业内对于易用性的应用基本处于面向产品的易用性检测阶段。这种方式决定了软件产品只能在下一版本中对于相关的性能作调整,易用性评价滞后,无法在软件生产过程中约束产品的易用性问题。文献[6]提出基于观点挖掘的产品可用性建模与评价方法;文献[7]把可用性分为外观可用性、感知可用性和绩效可用性,提出基于模糊层次分析法的产品可用性评价方法;文献[8]研究和实现了面向 Web 业务系统的可用性评估系统。以上文献对易用性的度量均是针对已成型的软件产品度量,这种针对易用性的度量导致只能针对下一版本软件进行约束,对于已成型的软件产品则无能为力。ISO9241 中也强调了中长期评估设计所带来的效应的重要性:当评估交互产品时,有的效应只有在长时间使用产品服务之后才会显现出来。因此,越早对产品的易用性进行评估,就能越早地发现问题,并加以解决。并且每个项目是不同的,因此最终设计的产品也是不同的。然而获得好的结果的过程是基本一致的。并且过程是成熟的,适用于所有与易用性相关的产品。1987 年, Osterwil^[9]提出软件过程也是软件,并在文章中详细阐述了过程的本质以及过程也能够像软件一样执行。目前,这一思想也逐渐被业界广泛接受,人们越来越意识到:软件产品的质量在很大程度上依赖于产品开发时所使用的过程^[10,11]。因此,如果从软件开发的初始阶段就对软件产品的易用性问题进行控制、评测,那么可以使得软件产品在增加少量的成本和时间的情况下极大地增强产品的易用性,可以提高产品的满意度和竞争力。

c) 由于易用性概念的模糊性以及人们对其感知的不确定性,导致对易用性的度量存在较大的问题,业界和学术界均没有一个量化的合理标准,进而无法很好地指导软件的开发。模糊层次分析法作为一种定性与定量相结合的、系统化、层次化的分析方法,十分适合使用在软件易用性的度量中。在文献[7,12~14]中均使用此方法进行度量,具有相对的影响力。因此本文也沿用此方法对易用性进行度量。

同时,软件开发与演化是以软件过程为中心的,软件过程的规范与否将直接影响软件产品的质量,这也是 CMMI 所讨论的问题。目前,国内对于易用性过程的度量尚属空白,对易用性控制的滞后或缺失,导致生产厂家开发项目失败、成本增加、周期延长、产品竞争力以及企业信誉度受损;在用户方面造成严重的使用成本问题,导致培训和技术支持的开销增加,生产力水平降低。在过程中对易用性加以规范、度量,就能够很好地解决上述问题,保证软件产品具备好的易用性。从软件开发的初始阶段就对软件产品的易用性问题进行控制、评测,使得软件产品在增加少量成本和时间的情况下极大地增强产品的易用性,提高产品的满意度和竞争力。

针对以上问题和研究热点,笔者深入研究软件过程之于易用性的相关过程活动,在软件过程中提取出与易用性相关的过程以及活动,并将此过程/活动映射到 ISO9126^[14],使用现今广泛认可的国际标准定义易用性,并对每一项过程/活动通过进行易用性评测,然后固化于软件过程中,构建出软件易用性过程综合评价模型,并使用模糊层次分析法进行度量、评测,更直观、更形象。通过此方法,指导企业从产品开发初期就能够做

出高易用性的产品,解决了易用性评估滞后、评价不定量所带来的问题。

1 软件易用性过程度量模型

1.1 易用性开发生命周期模型

软件质量保证的目的在于实时地探测和反馈软件生命周期中对软件质量有影响的关键点,保证软件开发的顺利进行。易用性作为软件质量保证中最基本的一个属性^[4],对软件生命周期中的质量考量也具有十分重要的意义。传统的软件开发生命周期模型有瀑布模型、原型模型、演化模型、螺旋模型等^[15]。通过对这些模型的分析,发现软件开发生命周期中,从需求到最终交付用户使用存在以下基本过程阶段:需求分析、设计、实现、测试和验证、发布。软件生命周期如表 1 所示。在需求分析阶段的里程碑为需求说明书,设计阶段的里程碑为设计说明书,在实现阶段的里程碑为可测试的程序代码,在检验和验证阶段实现的里程碑为可交付用户使用的软件产品。发布之后需要对软件进行相应的维护和售后服务以及对用户进行评估。不论软件开发过程使用哪种过程模型,都是由以上基本过程块构成的,因此,以上过程块可以作为基础而通用的质量评价目标,对于其他的过程模型则可针对以上过程模块通过重复、循环等模式加以应用。

表 1 软件生命周期基本过程块

软件生命周期过程	质量评价里程碑
需求分析	需求说明书
设计	设计说明书
实现	程序代码、设计图
检验和验证	测试用例

通过研究软件过程中与易用性相关的过程,结合易用性工程中的易用性工程生命周期^[16],可以建立易用性开发生命周期模型,如图 1 所示。

需求分析——需求说明书	设计——设计说明书	实现——程序代码	检验/验证——测试用例	发布
●用户需求分析 ●易用性需求定义 ●易用性需求评估定义 ●易用性需求度量评价	●确立易用性结构 ●使用已验证的原则 ●并行设计 ●参与型设计 ●反复设计 ●设计中的易用性评估	●避免个人定义 ●捕捉设计道理	●易用性测试 ●对易用性测试评价	●用户培训 ●维护

图1 易用性开发生命周期模型

1) 需求阶段

a) 用户需求分析。对用户需求进行分析,被作为一个最小的、最有效的、可靠的人机交互定义。它具有以下子特性:

(a) 定义用户习惯和学习能力,首先定义用户范围,用户不仅仅包括确实操作此系统的用户,还应该包括安装人员、维护人员、系统管理人员以及其他的支持人员;其次了解用户特征,通过了解用户的年龄组成范围、教育程度、计算机使用能力、工作经验以及工作环境、社会环境等了解用户特征,方便后续的界面定义以及文字定义;最后定义用户习惯和学习能力,并对用户演变作出预测。

(b) 收集当前相关产品的最新方法,可对此类相关产品做易用性测试,测试主要包括易理解性、易学习性、可操作性和吸引力;对不同的产品作出比较,分析相关产品的优势和弱点,并将分析结果运用到自己的产品研发中。

b) 易用性需求定义。此过程定义用户与系统之间的交

互,明确产品或系统的易用性目标。它具有以下子特性:

(a)根据上一阶段的用户需求分析,从易学习性、易理解性、可操作性等方面入手,定义产品的功能目标,以辅助产品设计、开发、测试。

(b)与开发人员协同工作,以明确产品的易用性需求、目标和可接受原则。

(c)对产品需求进行经济影响分析,它包括两个部分:关于对开发组织影响的评估(有助于确定易用性预算的量级)和关于对用户组织影响的评估(有助于对可用的易用性资源进行优先排序)^[16]。

c)易用性需求评估定义。定义产品易用性的度量和验收标准,确定产品易用性测试时间表。评估结果应该记录在文档中。

d)易用性需求度量评价。对产品的需求进行与易用性相关的度量评价,具体度量方法在下文中有阐述。

2)设计阶段

a)确立易用性结构。其目的是为了确立哪些系统的易用性需求应分配给系统的哪些模块。需要制定一个顶级的易用性系统体系架构,并确保所有的易用性需求分配到系统的每一项中。它具有以下子特性:

- (a)制定产品用户界面样式;
- (b)定义产品概念模型;
- (c)测试概念模型;
- (d)设计原型并测试。

b)使用已验证的原则。开发团队定义和发布一个标准的易用性设计手册(原则),手册中定义已规约的原则,如 28 原则、101 原则等,并且此设计原则应该被团队的产品经理认可。一般来说,开发团队应该尽量遵照手册上的原则执行。使用已经制定的规则可以降低产品风险,并为易用性测试提供便利。

c)并行设计、参与型设计、反复设计^[16]。

d)设计中的易用性评估。对产品的设计进行易用性相关的评估,重点是界面的评估,具体度量方法在下文中有阐述。

3)实现阶段

a)避免个人定义。在开发中遵照设计手册中的内容,不应任意地、随意地改变原有设计。如果有改变,必须经由产品经理及设计人员的同意,并应将改变写入文档,且编制入产品手册中。

b)捕捉设计道理^[16]。

4)测试阶段

a)易用性测试。易用性测试的目的是确认软件产品满足其产品需求。成功的易用性测试应满足:符合软件易用性需求;符合设计手册的规定;结果被记录。易用性测试应包括易理解性测试、易学性测试、易操作性测试、吸引力测试和易用的依从性测试。

b)对易用性测试评价。

5)发布阶段

a)用户培训。针对专有用户,使用专门的培训人员对系统的使用培训,包括系统操作培训、环境培训等等。

b)维护。发布后针对易用性问题对产品进行维护,记录易用性问题成为文档,为产品的再次开发提供帮助。

1.2 软件易用性度量模型

对易用性的度量中,本文沿用 ISO9126^[17]对易用性的定

义,将易用性划分为易理解性、易学习性、可操作性、吸引力以及易用性的依从性。在 ISO9126^[18]中,对这些易用性的子特性提供了评价因素以及评判标准,如图 2 所示。

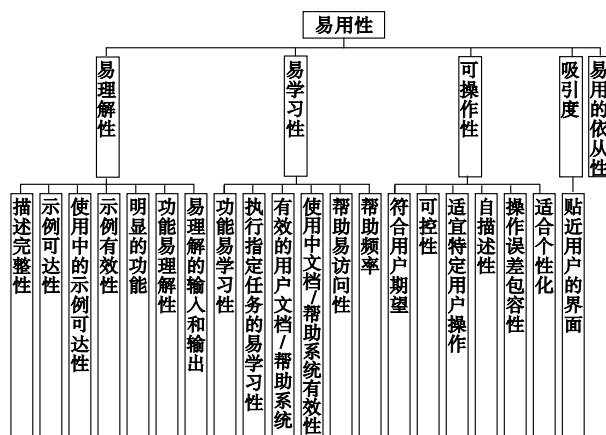


图2 软件易用性度量模型

1.3 软件易用性过程综合评价模型

ISO9241 中强调了中长期评估设计所带来的效应的重要性:当评估交互产品时,有的效应只有在长时间使用产品服务之后才会显现出来。因此,在过程中尽早地评估和操作易用性是可行的,也是必要的。同时笔者认为,软件易用性过程度量不应单纯地理解为 ISO9126^[17]中所概括的子属性的集合,也不能简单地理解为需求、设计、实现、测试、维护等阶段中易用性的线性组合,它是一个综合的、复杂的、相互制约的问题。软件易用性过程可以分解为一组影响它们的活动。以需求阶段为例,可分解为用户评估、相关产品评估、易用性需求定义评估。为了更加详实、准确地描述软件过程中的易用性,又可以将活动分解为子特征。而对于分解出的每组特征,为了能够进行定量或定性的分析与计算,需要构建出软件过程易用性的层次结构模型——易用性综合评价模型。模型的构建主要通过对软件生命周期中的各个过程块进行易用性分析。根据文献[18]所构建的软件演化过程框架,从软件生命周期过程块出发,以每个过程块下的活动为对象,选取可以度量的活动,从易用性度量模型中选取相关的特征作为度量指标进行度量,并对每个活动列出度量指标。具体模型如图 3 所示。

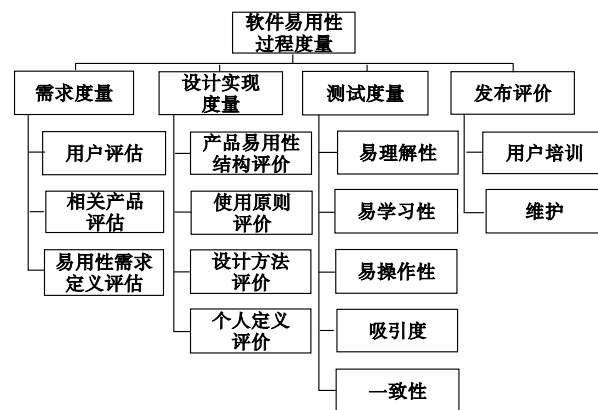


图3 软件易用性过程综合评价模型

2 易用性评估方法

模糊综合评价^[19]是基于模糊集和理论的一种具体应用方法,是对具有多种属性的事物或总体优劣受多种因素影响的事物作出的一个综合这些属性或因素的合理的总体评价,且结合

易用性过程的特点,即受个人因素影响较大,具有一定的非客观性,故使用模糊层次度量法对其进行综合评价是合适的。模糊层次分析法的基本步骤为:确立评价指标体系;设定评价集并确立评价因素的隶属度函数;确立评价因素的权重;进行模糊综合评价。

2.1 层次模型的因素指标集确认

易用性过程综合评价模型中的各个子特性即是层次模型中的因素指标集。对指标集的评价方法如表2所示。

2.2 确立评估等级及隶属度函数

根据模糊分析法,确定评语等级 $V = \{v_1, v_2, v_3, \dots, v_n\}$ 以及评价因素在 V 上的隶属度函数。将评语等级定义为四个等级 {perfect, good, average, poor}, 分别代表用户对于软件易用性的质量体验等级:优,良,中,差;然后由易用性评估专家给出在每个等级下的阈值 $(v_1, v_2, v_3, v_4, v_5)$, 即每个指标的取值范围;最后根据构造的隶属度函数将度量值和阈值进行比对,得到各因素的评价等级。

本文采用梯形分布与半梯形分布^[20]构建软件易用性评价等级的隶属度函数。梯形分布和半梯形分布是构造隶属度函数时常用到的分布函数,并且符合质量渐变连续的特点,因此,使用模糊方法评价事物的质量时一般使用这两种分布^[21]。

设 A 为评价等级上的模糊子集,隶属度函数为

$$\mu A_1(v) = \begin{cases} 1 & v_1 \leq v \leq v_2 \\ \frac{v - c_2}{v_2 - c_2} & v_2 < v < c_2 \\ 0 & \text{其他} \end{cases} \quad (1)$$

$$\mu A_2(v) = \begin{cases} 1 & v_2 \leq v \leq v_3 \\ \frac{v - c_1}{v_2 - c_1} & c_1 \leq v \leq v_2 \\ \frac{v - c_3}{v_3 - c_3} & v_3 \leq v \leq c_3 \\ 0 & \text{其他} \end{cases} \quad (2)$$

$$\mu A_3(v) = \begin{cases} 1 & v_3 \leq v \leq v_4 \\ \frac{v - c_2}{v_3 - c_2} & c_2 \leq v \leq v_3 \\ \frac{v - c_4}{v_4 - c_4} & v_4 \leq v \leq c_4 \\ 0 & \text{其他} \end{cases} \quad (3)$$

$$\mu A_4(v) = \begin{cases} 1 & v_4 \leq v \leq v_5 \\ \frac{v - c_3}{v_4 - c_3} & c_3 \leq v \leq v_4 \\ 0 & \text{其他} \end{cases} \quad (4)$$

其中: v_1, v_2, v_3, v_4, v_5 分别是不同等级下的阈值, c_1, c_2, c_3, c_4 分别为区间 $(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_5)$ 的中点。通过此种方法,对于特定的指标值 v 可以通过式(1)~(4)得出在四个级别上的值,如图4所示。

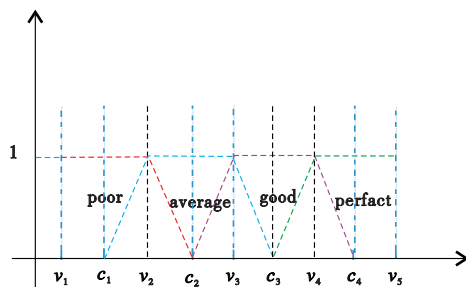


图4 易用性隶属度分布函数

2.3 评估因素权值的确定

根据 AHP 方法首先建立权重比较矩阵。设存在 n 个评价因素的确定权重,则权重比较矩阵如下:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

表2 易用性过程评价方法

软件过程块	易用性活动	子特征	易用性映射	度量方法	评价结果
需求	用户评估	年龄	易学习性、易理解性	经验分布函数	数值
		使用计算机年份	易学习性、易理解性	经验分布函数	数值
		受教育程度	易学习性、易理解性	经验分布函数	数值
	相关产品评估			是否进行相关产品评估,评估程度	百分比
	易用性需求定义评估	用户进行评估	易学习性、易理解性、易操作性、吸引力	用户调查,走访,问卷	数值
		开发人员进行评估	易学习性、易理解性、易操作性、吸引力	讨论	数值
		测试人员进行评估	易学习性、易理解性、易操作性、吸引力	讨论	数值
设计实现		经济影响分析		是否进行经济影响分析,分析深度	百分比
	产品易用性结构评价	界面评价	易学习性、易理解性、易操作性、吸引力、一致性	ISO9126 ^[18] 贴近用户界面的百分比	数值或百分比
		概念模型评价	易学习性、易理解性、易操作性、吸引力、一致性	ISO9126 ^[18] 中对易用性的度量方法	数值或百分比
	原型评价	原型评价	易学习性、易理解性、易操作性、吸引力、一致性	ISO9126 ^[18] 中对易用性的度量方法	数值或百分比
		使用已有原则	一致性	使用已有原则的百分比	百分比
		并行设计、参与型设计、重复设计		是否使用此三种设计方式,使用的程度	数值和百分比
	避免个人定义			使用个人定义的百分比	百分比
测试	易用性测试	易理解性		ISO9126 ^[18] 中易理解性度量	数值或百分比
		易学习性		ISO9126 ^[18] 中易学习性度量	数值或百分比
		易操作性		ISO9126 ^[18] 中易操作性度量	数值或百分比
		一致性		ISO9126 ^[18] 中一致性度量	数值或百分比
		吸引力		ISO9126 ^[18] 中吸引力度量	数值或百分比
评价	用户培训维护			是否进行用户培训	布尔值
				是否进行易用性维护	布尔值

其中: a_{ij} 表示因素*i*和*j*的重要程度比。根据表3来构建权重矩阵。其中, $a_{ij}=1/a_{ji}$ 。

表3 指标权重标度

权重意义	模糊度量级
因素 <i>i</i> 与 <i>j</i> 同等重要	1
因素 <i>i</i> 比 <i>j</i> 略重要	3
因素 <i>i</i> 比 <i>j</i> 明显重要	5
因素 <i>i</i> 比 <i>j</i> 绝对重要	7
中间值	2,4,6

对权重比较矩阵进行归一化,求得权重值。此处使用方根法判断矩阵*A*的权重向量。具体步骤如下:首先计算比较矩阵的每一行元素乘积;计算乘积的*n*次方根,并进行归一化处理,即得其权重向量 W_i 。公式如下:

$$W_i = \frac{\sqrt[n]{\prod_{j=1}^n a_{ij}}}{\sum_{i=1}^n \sqrt[n]{\prod_{j=1}^n a_{ij}}} \quad i=1,2,\dots,n \quad (5)$$

其中:

$$w_i = \sqrt[n]{\prod_{j=1}^n a_{ij}} \quad i=1,2,\dots,n \quad (6)$$

然后对权重矩阵进行一致性判断。判断方法如下:

一致性指标 CI 定义为

$$CI = \frac{\lambda_{\max} - n}{n - 1} \quad (7)$$

其中 $\lambda_{\max}$ 是判断矩阵的最大特征:

$$\lambda_{\max} = \sum_{i=1}^n \frac{(Aw_i)_i}{nw_i} \quad (8)$$

当 $CR \leq 1$ 时,判断矩阵具有满意的一致性,否则就要对判断矩阵进行重新调整。

2.4 进行模糊评价

建立因素集和评语集后,即可对软件产品的易用性进行评价,建立两者间的模糊关系矩阵:

$$R = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nn} \end{pmatrix}$$

其中: R_{ij} 表示指标 u_i 被评为 v_j 的隶属度。通过式(1)~(4)求得。然后可得该指标的综合评判: $B=A \cdot R$ 。其中合成算子采用加权平均法计算($M(\cdot, \oplus)$)。

$$B_i = \sum_{j=1}^n a_j r_{ij} = \min \left\{ 1, \sum_{j=1}^n a_j r_{ij} \right\} \quad (9)$$

3 案例评估

本文将此方法应用到一个实际项目中。此项目作为《工程数学》的教辅工具,共历时四个月,开发过程严格遵守软件过程相关内容,从项目可行性分析、需求分析、设计、测试到产品安装,完成了一个包含基本加减乘除运算、面积体积运算、矩阵运算、二进制运算等的多功能数学计算器。在开发阶段,笔者收集了与易用性相关的评估数据,使用上述方法对此软件的易用性过程进行评价,并在项目完成后对此数学计算器的成型软件进行了易用性评估,以求通过对比,验证此方法的有效性。

首先在软件过程中对易用性进行评估。从需求阶段开始

就对软件易用性过程进行评估。使用定性评估的方法,因为定性评估方法可以挖掘出一些蕴藏很深的思想,使评估的结论更全面、更深刻^[22]。评估过程为首先建立软件易用性过程综合评价模型,如表4所示。

表4 需求过程易用性综合评价模型

过程	活动	子特征	评价方法	取值	专家阈值
用户评估		是否进行年龄评估,评估人数	评估人数为20人,年龄段为:19~45岁	0.42	(1,0.8,0.6,0.3,0)
		是否进行计算机使用年限评估,评估人数	评估人数为20人,使用年份为5~23年	0.47	(1,0.8,0.6,0.3,0)
		是否进行受教育程度评估,评估人数	评估人数为20人,受教育程度:大学及以上,最高受教育程度为博士	0.51	(1,0.8,0.6,0.3,0)
需求	相关产品评估	对市面上已有的具备代表性的计算机进行评估:Windows 7系统计算器、Matrix 10、Kalkules(高等数学计算器)		0.95	(1,0.8,0.6,0.3,0)
易用性需求定义评估	用户进行评估	用户调查,走访,问卷		0.11	(1,0.8,0.6,0.3,0)
	开发人员进行评估	讨论		0.74	(1,0.8,0.6,0.3,0)
	测试人员进行评估	讨论		0.51	(1,0.8,0.6,0.3,0)
	经济影响分析	是否进行经济影响分析,分析深度		0.46	(1,0.8,0.6,0.3,0)

根据上述模型,通过公式可以求得需求阶段、设计阶段、测试阶段、评价阶段的易用性评估向量如下所示:

需求阶段 $B_1 = [0.1092 \ 0.5044 \ 0.3864]$

设计阶段 $B_2 = [0.206 \ 0.231 \ 0.075 \ 0.488]$

测试阶段 $B_3 = [0.0 \ 0.315 \ 0.685]$

评价阶段 $B_4 = [0.0 \ 0.412 \ 0.588 \ 0]$

经过最终评估,可得此软件的易用性过程评估结果为 $B = [0.121 \ 0.3640 \ 0.168 \ 0.347]$ 。

假设对此评语集赋值(优:90,良:80,中:60,差:30),则对*B*转换赋值为72.9,属于中等范围。由此可见,在此软件开发过程中,易用性过程评价为中等。

在此软件项目开发完毕后,对此产品进行了易用性度量。该度量使用了业界流行的可用性评估方法——软件启发式评估。该方法是使用一套相对简单、通用、有启发性的可用性原则进行可用性评估。在此案例的可用性评估中,聘请四位可用性专家确定权重。

选取九位用户对该数学计算器进行操作。分为三组,每组进行两个计算类别的操作,其中两人指定不同的操作内容,一人不指定具体操作(即自由操作)。由工作人员对其操作的时间、操作位置、点选按钮进行记录。在操作结束后,请用户填写启发式评估调查问卷。经过统计之后,得出下述结论,如表5所示。

表5 启发式评估结果

二级指标	三级指标	组合权重	统计结果
有效性	界面能否正确完成特定功能或行为转换	0.109 8	1
	界面能够提供帮助用户完成任务转换的提示功能	0.063 8	1
	完成特定功能或行为转换的正确率	0.062 4	1
	界面是否以平时的语言精确呈现错误信息,提供详尽而可靠的帮助文档系统	0.066 3	0.753
	用户能否方便地退出异常状况	0.040 8	0.862
	是否支持“撤销”和“恢复”功能	0.023 1	0.302
	相似操作的执行方式是否相同	0.040 8	0.944
	任务完成率(完成任务数、总任务数)	0.066 8	0.855
	界面在没有颜色的情况下是否仍然可用	0.049 7	1
	界面的系统术语和操作逻辑是否保持一致	0.046 0	1
可用性	系统在出错时是否提供通俗易懂的语言精确指出问题的所在,并给用户提出修正错误的方法建议	0.047 3	0.672
	完成任务所耗时间	0.044 6	70 s
	是否提供了适当的反馈	0.015 7	0.834
	界面是否提供了必要的信息,避免让用户记忆过多的信息	0.026 4	0.841
	界面是否有良好的菜单设计、界面是否简洁明了、除去了不必要的信息	0.016 7	0.940
	界面是否使用用户熟悉的词汇、概念	0.015 3	0.940
	效率		
	界面的命令按钮与使用频繁的按钮是否放在界面上的醒目位置	0.034 4	0.651
	界面是否提供快捷方式,以使用户操作具有灵活性和高效性	0.023 5	0.012
	界面中相似功能的界面或按钮是否具有统一性	0.021 6	0.966
用户满意度	界面的操作难度	0.042 1	0.834
	在错误上消耗的时间	0.025 8	28 s
	帮助和文档使用的次数	0.025 2	0
	用户使用后的直觉感受	0.018 1	0.977
	完成任务转换后显示的界面是否具有较好的符合可用性准则	0.022 1	0.841
	能正确完成的任务与所有目标任务的比率	0.031 7	98%
	完成任务的效率让用户满意的程度	0.033 7	98%

将统计结果归一化,可得此数学计算器的易用性评估结果为0.7456,即按100分来计算,此数学计算器的易用性评估结果为74.6分。

通过对案例使用两种易用性评估方法可以看出,使用笔者的易用性过程评估方法对软件过程进行评估的结果与面向产品的易用性评估结果相近。由此可以推断,使用软件易用性过程评估方法可以有效地在软件过程中评测易用性的好坏,并可以预测最终产品的易用性。同时,易用性过程评估方法可以及时调控软件易用性,为开发出高易用性的软件产品提供一种有效的方法。对于上述方法,笔者仍在继续进行实验性评估验证,希望能够找到一种最有效的评估标准。

4 结束语

目前,越来越多的企业开始关注易用性在软件生产过程中的影响,但是目前国内尚未有成型的过程评价方法。本文根据已有国际标准进行提炼,同时运用模糊层次评价法对易用性进行了度量和量化。可以根据软件开发过程较好地评判软件易用性,给出了易用性能力成熟度的一种评价方法。

参考文献:

- [1] ISO 9241-11, Ergonomic requirements for office work with visual display terminals[S].
- [2] HALSKOV K. The diversity of usability practices[J]. *Communication of the ACM*, 1999, 42(5): 60-97.
- [3] ROSENBAUM S, JOHN J, HUMBURG J. A toolkit for strategic usa-

bility: results from workshops, panels, and surveys[C]//Proc of SIGCHI Conference on Human Factors in Computing Systems. New York: ACM Press, 2000: 337-344.

- [4] ISO25010, Systems and software engineering-systems and software quality requirements and evaluation (SQuARE)-system and software quality models[S].
- [5] 闵庆飞,王菊,刘振华. 移动系统可用性综合研究框架[J]. *计算机应用研究*, 2012, 29(2): 421-426.
- [6] 易力,王丽亚. 基于观点挖掘的产品可用性建模与评价[J]. *计算机工程*, 2012, 38(16): 270-274.
- [7] 李永锋,朱丽萍. 基于模糊层次分析法的产品可用性评价方法[J]. *机械工程学报*, 2012, 48(14): 183-191.
- [8] 吴鹏越,佟秋利,杜韶. Web业务系统可用性评估系统的研究与实现[J]. *计算机工程与设计*, 2009, 30(2): 338-341.
- [9] OSTERWEIL L J. Software processes are software too[C]//Proc of the 9th International Conference on Software Engineering. 1987: 2-13.
- [10] MONTANGERO C, DERNIAME J C, KABA B A, et al. The software process: modeling and technology[C]//Lecture Notes in Computer Science, vol 1500. Berlin: Springer-Verlag, 1999: 1-13.
- [11] CMMI Product Team. CMMI for development, version 1.2-improving process for better products[R]. 2006.
- [12] 周荣刚. IT用户体验质量的模糊综合评价研究[J]. *计算机工程与应用*, 2007, 43(31): 102-105.
- [13] 熊鹏程,范玉顺. 基于模糊层次分析法的集成软件质量评估模型[J]. *计算机应用*, 2006, 26(7): 1497-1499, 1506.

仍不低于 SNM 方法,主要在于字段权重的动态调整,提高了相似度的计算精度。数据集为 1.5 和 2 万条记录时,两种方法的效果相当,主要由于窗口大小为 50 时已足够用于相似重复记录的检测,基本不会出现漏配的现象,但在记录为 2.5 万条时改进方法仍优于 SNM 方法,因为改进方法的记录相似度计算精度要优于 SNM 方法。

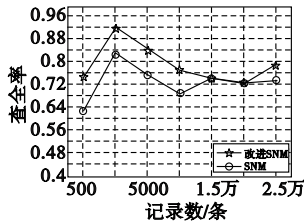


图5 SNM与改进SNM方法查全率对比

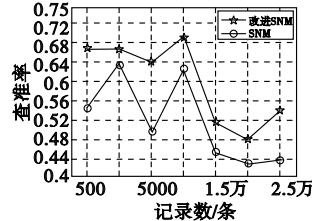


图6 SNM与改进SNM方法查准率对比

从图6中可以看出,窗口设为50后,改进SNM方法的查准率仍然高于SNM方法。由实际数据可知,窗口为50时,很多记录比较是没有必要的,此时改进方法能够缩小比较窗口,减少将非相似重复记录误认为是相似重复记录的情况,同时采用动态调整等级法,提高了检测的准确性。另外,从图中可以看出,针对不同数据集SNM方法查准率的波动较大,检测结果不够稳定。改进SNM方法除了1~1.5万条记录时查准率与SNM方法一样存在波动,其查准率整体比SNM方法稳定。

从图7中来看,比价窗口为50时,改进SNM算法的整体性能也是优于SNM算法的。

实验2中的程序运行时间对比如图8所示。从图8中可以看出,由于改进SNM算法采用了变步长伸缩窗口,减少了不必要的记录比较,节省了相似重复记录的检测时间,提高了比较效率。

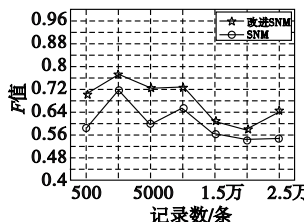


图7 SNM与改进SNM方法F值对比

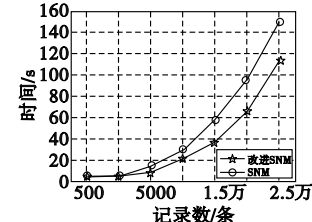


图8 SNM与改进SNM方法运行时间对比

4 结束语

本文详细阐述了SNM算法,并指出其存在的不足,提出了相似重复记录检测的改进方法。采用变步长伸缩窗口,在避免漏配的同时,减少了不必要的比较,根据记录相似度确定合理

的窗口改变步长的大小。采用动态调整等级方法,解决了人为设定固定权重的不足,提高了检测精度。并且还实现了噪声字段删除、等级、编辑距离、传递闭包等方法。通过对实际系统中数据的测试,验证了改进方法的有效性和优越性。同时可以看到,本文实现的SNM和改进SNM方法的各检测指标偏低,这主要是由于排序方法不科学、字符匹配函数不准确造成的,也是下一步有待改进和完善的地方。

参考文献:

- [1] 韩京宇,徐立臻,董逸生. 数据质量研究综述[J]. 计算机科学, 2008,35(2):1-5.
- [2] BILENKO M, MOONEY R J. Adaptive duplicate detection using learnable string similarity measures[C]//Proc of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington DC:ACM Press,2003:39-48.
- [3] CHANDEL A, HASSANZADEH O, KOUZAS N, et al. Benchmarking declarative approximate selection predicates[C]//Proc of ACM SIGMOD International Conference on Management of Data. [S. l.]: ACM Press,2007:353-364.
- [4] SARAWAGI S, COHEN W W. Semi-Markov conditional random fields for information extraction[C]//Advances in Neural Information Processing Systems. 2004.
- [5] VIOLA P, NARASIMHAN M. Learning to extract information from semi-structured text using a discriminative context free grammar[C]//Proc of the 28th Annual International ACM SIGIR Conference on Research and Development in information Retrieval. [S. l.]:ACM Press,2005:330-337.
- [6] 庞雄文,姚占林,李拥军. 大数据量的高效重复记录检测方法[J]. 华中科技大学学报,2010,38(2):8-11.
- [7] 王宏志,樊文飞. 复杂数据上的实体识别技术研究[J]. 计算机学报,2011,34(10):1843-1852.
- [8] 鲁均匀,李星毅,施化吉,等. 基于内码序值聚类的相似重复记录检测方法[J]. 计算机应用研究,2010,27(3):874-878.
- [9] HERNANDEZ M A, STOLFO S J. Real-world data is dirty: data cleansing and the merge/purge problem[J]. Data Mining and Knowledge Discovery,1998,2(1):9-37.
- [10] 陈伟,王昊,朱文明. 一种提高相似重复记录检测精度的方法[J]. 计算机应用与软件,2006,23(10):29-30.
- [11] 肖满生,周浩慧,王宏. 基于模糊综合评判的相似重复记录识别方法[J]. 计算机工程, 2010,36(13):51-53.
- [12] HERNANDEZ M A, STOLFO J S. The merge/purge problem for large databases[C]//Proc of ACM SIGMOD International Conference on Management of Data. [S. l.]:ACM Press,1995:127-138.

(上接第2735页)

- [14] 周津慧,王宗奎,等. 基于模糊评价方法的软件质量评价研究[J]. 系统工程与电子技术,2004,26(7):988-991.
- [15] SOMMERVILLE I. Software Engineering[M]. 9th ed. 程成,等译. 北京:机械工业出版社,2011.
- [16] NIELSEN J. 可用性工程[M]. 刘正捷,译. 北京:机械工业出版社,2004.
- [17] ISO/IEC 9126, Software engineering: product quality is an international standard for the evaluation of software quality[S].
- [18] LI Tong. An approach to modeling software evolution process [M].

北京:清华大学出版社,2008.

- [19] 肖刚,古辉,程振波,等. 实用软件文档写作[M]. 北京:清华大学出版社,2005.
- [20] 李安贵,张志宏,孟艳,等. 模糊数学及其应用[M]. 北京:冶金工业出版社,2005.
- [21] 石柱,何新贵. 软件质量的模糊属性及其表示[J]. 计算机工程设计,2001,22(4):1-4,20.
- [22] SWANSON M, BARTOL N, SABATO J, et al. Security metrics guide for information technology systems [R]. [S. l.]:National Institute of Standards and Technology Special Publication,2003.