

· 软件技术与实现 ·

一个软件质量评价系统的设计与实现

施小英 王 辉 钟权雄 周克俭 左孝凌

(上海交通大学计算机系 上海 200030)

摘要 文中介绍了一个基于国家标准的软件质量度量模型和评价过程模型(简称 JT-SQE 模型)的设计思想和主要技术。度量模型为四层次树形结构,评价过程分需求定义、度量和计分三个步骤。该模型为评价软件质量提出了一套可行的评价体系,并简单介绍了基于该模型的软件质量评价工具的结构和功能。

关键词 质量特性 子特性 质量度量 度量问题

Design and Implementation of a Software Quality Evaluation System

Shi Xiaoying Wang Hui Zhong Quanxiong Zhou Kejian Zuo Xiaoling

(Department of Computer Science, Shanghai Jiaotong University Shanghai 200030)

【Abstract】 Based on the national standard, the JT-SQE system consists of two parts. One is the model for software quality measurement, which is of hierarchy structure. The other is the processes of requirement definition, measurement and grading. According to the model and the processes, a tool is implemented. The structure and function are also introduced in this article.

【Key words】 Quality characteristic; Sub characteristic; Quality metric; Metric question

为了科学地评价软件的质量,必须建立一套详细可行的软件质量评价体系,把对软件质量的度量量化。国家标准从用户观点出发,考虑软件的六个特性:功能性、可靠性、易用性、效率、可维护性和可移植性。这六个质量特性以最小的重叠描述了软件的质量,但还只是概括的、定性的。应该对这六个质量特性逐步细化,直至可以定量度量,从而建立一套详尽可行的质量度量模型。在度量模型基础上对软件质量进行度量和评价,就是评价过程。为了使软件质量的评价具有实用价值,我们对此进行了有益的探索和初步的尝试。我们的研究工作包括两方面的内容:一是设计一个基于国际标准 ISO9126 的软件质量度量模型和评价过程模型;二是开发了一个软件质量评价工具(简称 JT-SQE 系统)。^{*}

1 软件质量度量模型

为了度量软件的质量,必须了解什么是软件质量。它定义为⁽³⁾:对用户在功能和性能方面的需求的满足,对规定的标准、规范的遵循,以及正规软件理所当然应该具有的品质。

这个定义强调了以下三点:1)用户的需求是度

量软件质量的基础;不满足用户需求,软件质量就不过关;2)规定的标准、规范定义了软件开发的准则;不遵循所规定的标准、规范,必然导致软件质量欠缺;3)对软件的某些要求并不明确的提出来,但作为公认的准则,也应该得到满足。

度量软件质量而设计的模型必须能充分反映软件质量的上述要求。根据我们的研究,同时参照其它一些成果(如 Boehm, McCall 等人的模型),我们提出一个软件质量度量模型,称为 JT-SQE 度量模型。JT-SQE 度量模型自上而下,从抽象到具体分为四个层次,分别为质量特性、质量子特性、质量度量和度量问题。JT-SQE 模型自上而下分别是一对多的关系,因而构成树形结构。

1.1 质量特性

在 JT-SQE 度量模型中采用国家标准推荐的六个软件质量特性⁽²⁾:

(1) 功能性:满足明确或隐含的需求的程度。

^{*} 施小英 女,51岁,副教授,主要研究软件工程,面向对象技术和质量评价

收稿日期:1997-03-18

(2) 可靠性: 在规定的条件和条件下, 软件维持其性能水平的能力。可靠性除了反映软件满足用户需求正常运行的程度, 也反映在故障发生时能继续运行的能力。

(3) 易用性: 用户为使用软件所需作的努力。

(4) 效率: 在规定的条件下, 软件的性能水平与所使用资源量这间的关系。资源可包括其他软件产品、硬件设施和操作, 效率反映了在完成功能要求时, 有没有浪费资源。

(5) 可维护性: 进行指定的修改所需的努力。一个易于维护的软件, 便于纠正错误, 或根据用户需求的改变进行相应的改变, 从而延长了软件的寿命。

(6) 可移植性: 从某一环境转移到另一环境的能力。

在软件质量特性之间存在着有利影响或不利影响。例如由于效率的要求, 应尽量使用汇编语言, 然而用汇编语言编写的软件系统的可维护性却很差。所以说软件质量特性之间有时是矛盾的。

1.2 质量子特性

设计了如下 16 个质量子特性:

(1) 功能性

准确性: 能否得到正确或相符的结果或效果的能力; 依从性: 遵循有关的标准、约定、法规及类似规定的程度; 安全性: 防止对程序及数据的非授权的故意或意外访问的能力。

(2) 可靠性

成熟性: 由软件故障引起失效的频度; 容错性: 在软件故障或违反指定接口的情况下, 维护规定的性能水平的能力; 易恢复性: 在失效发生后, 重建某性能水平并恢复直接受影响数据的能力, 以及为达此目的所需的时间和努力。

(3) 易用性

易理解性: 用户为认识逻辑概念及其应用范围所花的努力; 易学性: 用户为学习软件应用(例如运行控制, 输入输出)所花的努力; 易操作性: 用户为操作和运行控制所花努力。

(4) 效率

时间特性: 软件执行其功能时的响应时间和处理时间; 资源特性: 软件执行其功能时所使用的资源数。

(5) 可维护性

易分析性: 为诊断缺陷或失效原因及为判定待修改的部分所需努力; 易改变性: 进行修改、排错或适应环境变化所需努力; 易测试性: 与确认已修改软件

所需的努力有关的软件属性。

(6) 可移植性

适应性: 软件无需采用有别于为该软件准备的的活动或手段就可能适应不同的规定环境的能力; 易安装性: 在指定环境下安装软件所需努力。

1.3 度量和度量问题

JT-SQE 度量模型的第三层为度量, 每一度量由若干度量问题组成, 这些度量问题构成了度量模型的第四层。根据度量问题的回答与计分可反映量的得分, 从而可反映质量子特性及特性的得分。我们设计了 30 多个度量和 100 多个度量问题, 细致地深入到影响软件质量的多个方面, 为度量软件的质量提供了较为可靠的依据。

2 软件质量评价过程

在度量模型的基础上, 就可以进行软件质量的评价。软件质量的评价过程分为 3 个步骤, 即质量需求定义、度量和计分与评级。

2.1 质量需求定义

此阶段主要完成如下 3 个工作:

(1) 规定对质量特性的需求

JT-SQE 度量模型的最上层采用了 6 个质量特性, 分别反映了软件质量 6 个方面的情况。但对于具体软件来说, 各个特性的重要性是不同的, 用户对其要求也有所不同。而软件质量是相对于用户需求的。同样的质量特性, 会由于用户需求的不同而得到不同的评价结果。比如, 1% 的故障率也许对于个人理财软件来说无伤大雅, 但对期货交易系统来说就是糟糕透顶的。因此, 在对软件进行评价之前, 必须由用户规定对软件质量的要求。软件质量评价的最终结果就是相对于这个质量需求的。

在综合考虑了软件环境及应用的特性后, 可以为软件质量特性规定一个原始的需求指标, 质量需求指标反映了该质量特性对软件的相对重要性。

(2) 质量需求指标的调整

然而质量特性之间是存在矛盾的。对用户所规定的质量需求, 有时不能得到满足。质量需求定义的第二步工作就是综合考虑特性间的关系后, 对用户规定的原始需求进行合理的调整, 使其更切实可行。

可根据质量特性之间的相互影响的关系⁽¹⁾, 对用户所规定的原始质量需求指标进行调整。比如某次调整如下, A、B、C 分别表示高、中、低的要求。为了定量化地表示特性与软件质量的关系, 指标应该用一个数值范围来表示。如果 x 代表质量指标的值,

则 A 可以表示: $0.9 \leq x \leq 1$, B 表示: $0.8 \leq x \leq 0.9$, C 表示: $0.6 \leq x \leq 0.8$ 。

表1 质量需求指标调整

特性 需求指标	效率	可靠性	易用性	功能性	可维护性	可移植性
原始	A	A	B	A	A	C
调整后	B	A	B	A	A	C

该软件中,效率、可靠性、功能性和可维护性的需求指标都是 A。由于效率与可靠性、效率与可维护性是相互矛盾的,要它们同时达到 A 级不大现实,因而适当降低对效率的要求,由 A 改为 B。

(3) 子特性加权

质量需求定义的最后一步工作是设定属于每个质量特性的子特性的子特性及其权值。

对每个定义了质量需求指标的特性,找出其所有子特性,并为这些子特性加权。其权值表明子特性在特性中所占的比重。一个特性的加权矢量为 $W = (W_1, W_2, \dots, W_m)$ 。其中, W_j 是第 j 个子特性的权值, $\sum_{j=1}^m W_j = 1$ 。

2.2 度量

度量是把度量和度量问题应用到软件产品上的活动。根据对度量问题的回答和测量,可以定量地得到软件在度量上的得分。度量问题分 3 种类型:

- Yes/No 型: 对度量问题作出肯定或否定的回答。

- ABCDE 型: 对度量问题不是明确地肯定或否定,而是用五个等级作出较模糊的回答。

- 统计型: 统计具有某属性的子模块数目,其占总模块数的比例等。比如统计某模块调用其它模块的次数。

2.3 计分与评级

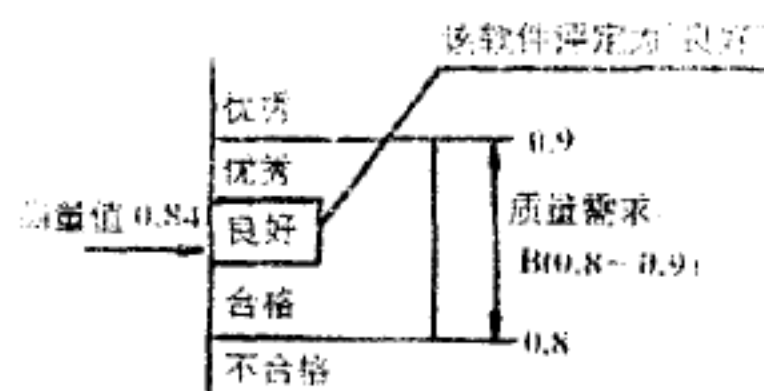


图1 特性得分到等级的映射

首先自上而下建立了软件质量度量模型(特性→子特性→度量→度量问题),然后在最下层把度量问题实施到软件上,得出度量的得分。最后自下而上,

由度量的得分,通过算术平均,得出各子特性的得分;再由子特性的得分,利用前面得到的加权矢量 W_j ,得到 6 个特性的得分。

特性的得分并不绝对表明软件质量的优劣,而是相对于用户需求的。对照用户的需求定义,我们把特性得分映射到软件质量上去,如图 1 所示。

3 JT-SQE 系统的实现

根据以上的质量度量模型和评价过程模型,我们在 Windows95 环境下开发了相应的软件质量评价工具,辅助专家进行软件质量的评价和评估。

3.1 系统结构

本系统由 7 个模块组成,而 7 个模块分别由若干个子模块组成,它们的功能如下:

- 1) 系统登录: 输入普通用户或系统管理员的口令,启动本系统。

- 2) 项目管理: 由新建项目、打开项目、关闭项目和保存项目 4 个子模块组成。对软件进行一次评价所输入和产生的所有数据都存放在该软件的项目文件里。项目文件以.mdb 为后缀。

- 3) 评价准备: 包括软件登记、专家登记、质量需求定义、子特性加权、等级定义等 5 个功能。在软件登记模块中输入该软件的有关信息,包括: 软件名称、开发单位、评价日期和软件简介。在专家登记模块中输入评价小组每位专家的有关信息,包括: 每位评价专家的姓名和简介。在质量需求定义模块中,用户规定对软件质量的要求,以及对该需求的调整。软件质量评价的最终结果就是相对于这个质量需求的。在子特性加权模块中,为每个特性的子特性加权。权值表明了子特性在特性中的比重。在等级定义模块中,定义软件得分(百分制)到软件等级(优、良、合格、差 4 级)的对应关系。

- 4) 评价过程: 采用 4 层次质量度量模型: 质量特性→质量子特性→度量→度量问题,根据对度量问题的回答和测量,可定量得到软件在度量上的得分。

- 5) 报表生成: 可以生成 12 种报表,以供查看和打印。

- 6) 系统管理: 包括修改口令、度量问题维护以及打开或关闭背景音乐 3 个功能。在度量问题维护模块中,可以选择某一度量,然后增加、修改或删除该度量所属的度量问题。

- 7) 联机帮助: 介绍本系统所使用的评价模型以及操作说明。

3.2 主要技术特点

1) 采用软件工程的方法和原理以及面向对象的技术进行设计和实现, 使各模块具有较高的内聚力和较低的耦合度的良好特性;

2) 在文件的设计中采用了双向链表的数据结构, 便于数据的访问;

3) 采用了树形结构作为度量问题打分的操作界面,使得外部界面与内在模型保持一致,有利于理解和使用;

4) 提供超文本方式联机帮助, 便于用户使用;

5) 使用了下拉菜单、滚动条、背景音乐、封面动画等技术, 为用户提供了方便、舒适的操作环境。

4 结束语

在 Windows95 环境下开发的软件质量评价工具: JT-SQE 系统, 是主要用来辅助专家进行软件质

量的评价和评估的。该工具具有界面友好、操作简单、易于修改和维护等特点,尤其设计了度量问题维护功能,允许专家自行增加度量问题,具有一定的可扩充性。运用该工具,对实际软件进行了质量评价,取得了较为可信的结果。但如何采用动态跟踪的方法,进一步提高质量评价的自动化程度,提高评价的准确性和可信度,亟待进一步去研究、探索。

参考文献

- 1 中国软件行业协会上海分会. 软件质量及其评价技术. 北京: 清华大学出版社, 1990. 47~52
- 2 中华人民共和国标准. 信息技术-软件产品评价-质量特性及其使用指南. 1991
- 3 Pressman R S. Software Engineering. 3Edition. McGraw-Hill Inc. 1992. 550

(continued)

(上接第5页)

理机 CPU0), 其它 CPU 在由 CPU0 启动后, 将等待 CPU0 进入操作系统后对它们进行引导。CPU0 在启动其它处理器后, 建立该处理器对应的 Restart Block 表格, Restart Block 主要用于操作系统对多处理器的引导进行控制并传递引导参数。每个处理器的 Restart Block 以链表的形式相连。CPU0 在进入操作系统后利用该表将引导标志及参数传递给等待引导的其它处理器, 其它处理器根据该表的引导标志以及参数转向指定的地址执行相应的 OS 程序。

2.7 开放式 Firmware

使用开放式 Firmware 的主要目的是使得系统的引导过程与系统配置无关。为实现这种引导过程的独立, 开放式 Firmware 要求引导设备具有一种驻留于适配器 ROM 内与处理器类型无关的引导 Driver, 它还必须提供一种执行这些 Driver 的方法。

开放式 Firmware 使用一种与硬件无关的解释语言即 FCode, FCode 具有一个为引导而扩充的 Forth 字典。开放式 Firmware 利用 FCode 建立描述系统硬件的配置树, 并进行内存的配置。Forth 编程语言、Forth 字典、硬件配置树以及内存配置是开放式 Firmware 用于引导所必需的四个基本要素。

对于没有 FCode ROM 的内置设备, 要求系统提供一种方法用于提供与开放式 Firmware 兼容的 Driver。一种实现方法是将这些用 FCode 编写的 Driver 从磁盘加载到 Flash ROM 或其它非易失性存储器中。

为实现平台的兼容性，开放式 Firmware 必须提供与其兼容的用户接口和设备 Driver 接口，并提供两种 Endian 方式(Big-Endian 和 Little-Endian)引导系统的能力。

3 结束语

Firmware 是一种新型计算机体系结构所必需的系统软件。开放式 Firmware 是其发展的主要方向。Firmware 的实现与移植是操作系统移植的前提和基础。Firmware 尤其是开放式 Firmware 的开发将是计算机研究开发的重点之一。

参考文献

- 1 PowerPC Reference Platform Specification (Version 1.0).IBM, 1994
- 2 IEEE Std. 1275-1994, IEEE Standard for Boot (Initialization Configuration) Firmware: Core Requirements and Practices, 1994