

武汉科技大学

硕士学位论文

中小企业软件质量度量模型的分析与研究

姓名：张刚

申请学位级别：硕士

专业：管理科学与工程

指导教师：汪勇

20081120

摘 要

随着信息化产业的发展，人们对软件质量的要求越来越高。但对软件质量评估目前还没有明确的准则和衡量标准，而且软件质量控制和评估在理论上和技术上都很不成熟，如何度量软件产品质量是几十年来一直困扰着人们的难题。

影响软件质量的某些质量属性是模糊的，造成了质量评估的困难。本文首先给出了软件质量的定义，介绍了软件质量管理的基本内容、影响软件质量管理的因素、软件质量管理存在的问题、两种软件质量保证体系及软件度量对软件质量的重要性。接着，介绍了软件质量度量方法，简要给出了五种软件质量度量模型，并进行了分析与比较，提出了新的软件质量度量模型。新的模型分为度量框架层和度量方法层。度量框架层主要是用于建立评价指标体系，该框架分为三个子层次，第一层为质量特性，它们是面向管理者和用户的质量属性；第二层为质量子特性，这些子特性体现了面向软件的质量属性，是对质量特性的进一步细化；第三层是度量，也叫指标量化层，是用户给予的具体评价，它的结果定量反映软件质量的优劣。度量方法层主要是在度量框架层建立的评价指标体系的基础上对软件质量进行综合定量评价。其中较为关键的是指标权重的确定，本文结合层次分析法和 Delphi 法给出了一种改进的层次分析法。

最后结合一个实例，对本文提出的软件质量度量模型进行了详细介绍，最后结果也反映了模型的合理性。

关键词：软件质量 软件质量度量 度量模型 层次分析法

Abstract

With the development of the information industry, the people's requirements to the software quality become more and more high. But because there are not clear norm and assessing standard at present, and software quality control and assessment are not well developed on the aspects of theory and technology, therefore, there are a lot of difficult problems to control software product quality systematically and objectively for decades.

Because the factors which affect the quality of the software are blurry, it is difficult to evaluate the quality of the software. At first this paper gives the definition of software quality, introduces the basic content of software quality management, the factors which affect software quality management, the problems of the software quality management, two kinds of software quality assuring system and the importance of software measurement to software quality. Then it introduces the methods of software quality measurement, gives five models of software quality measurement briefly, compares the models with each other, bring forward a new model of software quality measurement. The new model includes measurement structure and measurement method. According to measurement structure we can establish guideline system of measurement. This structure has three layers, the first is quality characteristic which faces governors and users, and the second is the seed of quality characteristic which faces the software, and is the detailed introduce of the quality characteristic, and the third is measurement also called indicator quantification layer which is the detailed value given by users and reflects the stand or fall of the software. On the base of the guideline system of measurement in the layer of measurement structure we can have a quantitative integrated evaluation of the software quality. in the layer of measurement method layer. In this layer, how to confirm the weight of factors is pivotal. This paper puts forward a new AHP through uniting the AHP and the Delphi.

At last, through an example it gives a detailed introduce of the new model, and the result of the example reflects the rationality of the new model.

Keyword: software quality/software quality measurement / model of measurement/AHP

武汉科技大学

研究生学位论文创新性声明

本人郑重声明：所呈交的学位论文是本人在导师指导下，独立进行研究所取得的成果。除了文中已经注明引用的内容或属合作研究共同完成的工作外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

论文作者签名： 张刚 日期： 2008.12.8

研究生学位论文授权使用授权书

本论文的研究成果归武汉科技大学所有，其研究内容不得以其它单位的名义发表。本人完全了解武汉科技大学有关保留、使用学位论文的规定，同意学校保留并向有关部门(按照《武汉科技大学关于研究生学位论文收录工作的规定》执行)送交论文的复印件和电子版本，允许论文被查阅和借阅，同意学校将本论文的全部或部分内容编入有关数据库进行检索。

论文作者签名： 张刚

指导教师签名： 12月

日 期： 2008.12.8

第一章 绪论

1.1 选题背景和研究意义

近年来,随着计算机应用领域的迅速扩大,计算机软、硬件新技术的不断涌现,人们对软件质量提出了更高的要求。目前,计算机已经广泛应用于航空、航天、工业控制、交通、银行、金融、医疗等领域。在这些应用领域中,软件质量往往关系到人民生命财产和生态环境的安危。一旦软件发生故障,就可能造成人的生命和财产的巨大损失或生态环境的破坏。

并且随着信息时代的发展,计算机软件的需求愈来愈复杂,规模愈来愈大。软件企业随着自身规模的不断扩大,所从事的软件工程项目的工作量和复杂度也不断升级。这些都导致软件企业在软件开发、服务提供和工程实施过程中所遇到的问题越来越多,软件质量要得到保障越来越具有挑战性^{[1][2]}。

软件质量是一模糊的、捉摸不定的概念。我们常常听说:某某软件好用;某某软件功能全、结构合理、层次分明、语言流畅。这些模模糊糊的语言实在不能算是软件质量评价,特别不能算是软件质量科学的定量的评价^[3]。

软件质量问题,已经成为西方工业国家政府、工业界、学术界以及社会各界关注的重要问题。1994 年夏,微软公司、IBM 公司、苹果公司等计算机厂商邀请了在英国的一些世界著名的计算机科学家探讨 21 世纪计算机软件的发展方向和战略,与会专家一致认为,21 世纪计算机软件发展的大方向将是质量的提高优先于性能和功能的改进,超高质量软件的开发技术将是打开 21 世纪高技术市场的钥匙。

在我国,软件产品质量低下,企业缺乏竞争力,更是一个长期困扰软件企业的问题。信息产业部在十五计划中,把提高软件产品质量,实现软件的产业化发展,加速软件产业和际的接轨,作为我国软件发展的重要目标。2000 年 6 月国务院颁发了《鼓励软件产业和集成电路产业发展的若干政策》。该文件的第五章第十七条明确提出“鼓励软件出口型企业通过 GB/T19000-ISO9000 系列质量保证体系认证和 CMM(能力成熟度模型)认证。其认证费用通过中央外贸发展基金适当予以支持^{[4][5]}。”

随着竞争的日益激烈,国内软件企业越来越意识到提高软件产品质量和客户满意度对于企业生存和发展的战略意义,因此研究和应用软件质量度量技术,建立企业的软件质量保障体系已成为一个迫切而重要的任务。

1.2 国内外研究现状分析

国际上从 20 世纪 70 年代就开始研究软件质量的控制问题,Rubey 和 Hartwick 在 1958 年首次提出从整体上度量软件的观点,但并没提出具体的度量方法和模型。

1970 年, Halstead 提出了软件科学(Software Science)的概念, 他认为任何一门学科要成为科学, 必须理论和实践结合, 而软件度量学正是反映了这种结合的学科。1976 年, Boehm 提出软件属性不仅有定性的研究, 还必须有定量的研究, 软件度量学正是顺应这种趋势产生的。用软件度量学的方法来科学地评估软件质量, 更有力地对软件开发过程进行控制和管理, 合理地组织和分配资源, 制定切实可行的软件开发计划, 以获得高质量的软件^[6]。

现在软件度量学已成为软件工程的一个重要研究方向。IEEE Std 1061—1992(Standard for a Software Quality Metric Method)中对软件度量学作了明确的定义, 对软件度量学的目的作了阐述。Boehm 等人于 1976 年提出了定量评估软件质量的概念, 并给出了 60 个质量度量公式, 还第一次提出了软件质量度量的层次模型; Boehm 等人将软件质量的概念分解为若干层次, 对于最底层的概念再引入数量化的指标, 从而得到软件质量的整体评价^[7]。

1978 年 McCall 和 Walters 提出了从软件质量要素、准则到度量的三个层次的软件质量度量模型。McCall 等人认为, 要素是软件质量的反映, 软件属性可用做评价的准则, 量化地度量软件属性可知软件质量的优劣。1985 年, 国际标准化组织 ISO 建议, 软件质量度量模型由三层组成, 分别对应 McCall 等人的质量要素、评价准则和度量。根据 ISO 近年来对软件质量的评价趋势, 反映软件质量要素选用逐渐向面向用户的观点靠拢, 使用户和软件人员方便地确定软件质量需求^[8]。

早期的软件度量学和当时的面向功能分解的设计方法相联系, 大多以模块度量为主。针对面向过程的程序设计, Yourdon 和 Constantine 提出了影响软件质量的主要因素: 藕合性、内聚性、复杂性、模块化及规模大小。目前, 面向对象技术在软件产业中得到了广泛运用。面向对象软件开发方法在软件生存周期、系统结构、复用以及项目管理等方面同传统开发方法有着显著的区别, 这些区别很大程度上影响了软件度量。传统的度量方法, 如 McCabe 的环计数、HalStead 的软件科学等, 不再适用于面向对象的某些概念, 如类、继承、封装和消息传递等, 也不足以评估面向对象软件的质量。因而面向对象的软件度量的研究得到了普遍重视。

随着软件质量领域知识的增长, 国际各标准化组织在软件质量方面做了大量工作, 目前, 对其研究主要从两方面展开: 1)软件开发过程的质量保证, 以过程文档化和科学化管理为内容。2)软件过程和产品的质量评估, 包括中间和最终产品, 采用软件度量技术作为软件质量特性量化的主要技术。国际上流行的标准和模型充分体现了这一点, 其中具代表性就是 ISO 系列和 CMM^[9]。

在我国, 关于软件质量度量和评估的研究进入逐步成熟阶段, 参照国际标准也制定了一系列软件标准, 如 SSC 模型, GB/T14394 计算机软件可靠性与可维护性管理, GB/T1995 信息技术—软件生存期过程。

1.3 本论文研究内容及结构

1. 主要研究内容

根据以上研究背景、意义及研究现状，本论文主要针对中小软件企业的软件质量度量模型作了一些分析与研究。

本论文首先对软件质量和软件质量度量模型进行了概述，然后对各种度量模型进行比较，提出新的度量模型。根据新的度量模型，采用多层次模糊综合评价方法对中小企业的软件质量进行度量。最后运用实例介绍新的模型的使用。

2. 论文结构

第一章简单地介绍了本课题的研究背景、意义及现状。

第二章对软件质量进行概述。介绍了软件质量与软件质量管理，及现在软件质量管理中存在的问题和影响因素，然后介绍了ISO9000和CMM两种比较流行的软件质量管理体系。

第三章介绍了软件质量度量方法和几种度量模型，并对各种模型进行了比较，提出新的度量模型。

第四章针对前一章提出的新的度量模型进行详细的分析，通过实例介绍了新的度量模型的运用。

第五章对本文的工作进行了总结，并分析了存在的问题和需要进一步研究的内容。

第二章 软件质量的概述

为了更好地度量软件质量，明确度量过程中必须注意的事项，以及更全面的度量软件质量，我们必须首先了解软件质量的定义以及软件质量管理，熟悉当前两种流行的软件质量管理体系。从这些了解中我们能知道究竟是哪些因素影响软件的好坏，明确了这些因素，就能够准确地度量一个软件的质量。

2.1 软件质量与软件质量管理

2.1.1 软件质量的定义

McCall 等认为，特性是软件质量的反映，软件属性可用做评价准则，定量化地度量软件属性可知软件质量的优劣。1976 年 Boehm 提出了软件质量特性，形成了现在软件质量特性的雏形。1978 年 McCall 提出了体现软件质量的 11 个特性，包括使用性、可测试性、正确性、维护性、可靠性、移植性、效率、重用性、完整性、互操作性和适应性(灵活性)。美国国家标准 ANSI/IEEE Std729—1983 定义软件质量为“与软件产品满足规定的和隐含的需求的能力有关的特征或特性的全体”。它包含了四层含义：能满足给定需要的特性之全体；具有所希望的各种属性的组合的程度；顾客或用户认为能满足其综合期望的程度；软件的组合特性，它确定软件在使用中将满足顾客预期要求的程度^[10]。

随着市场经济的发展，卖方市场向买方市场的转变，本文认为影响软件质量的特性分为三组，反映用户在使用软件产品时的三种不同倾向，核心是满足客户需求。这三种倾向是：产品运行、产品修改和产品转移。信息系统作为一个产品，也可以参照这三种倾向来定义，如图 2.1 所示。

本文认为，软件质量应以客户满意为中心，以能否实现客户需求为成功的标准，软件质量就是软件产品和服务能够实现客户明确和潜在需求，能够完成客户既定目标及后期维护的能力。它包括四个维度^{[11][12]}：

(1) 以客户满意为中心。顾客就是上帝，顾客的满意就是软件产品的最终目的。如果不能使客户满意，功能再强大的软件也是低质量的产品。在软件产品的生命周期内，要时时体现出这一原则，才能体现软件的质量。

(2) 符合客户需求。实际的需求包括用户明确说明的和其隐含的需求。由于专业知识、工作时间等条件的限制，有些客户不能完全全面的表达自身的需求，这就要求软件项目的承担者能够利用丰富的经验和负责的工作态度，在开发过程中引导客户，不断完善软件的功能，更好地实现客户的明确需求和隐含需求。

(3) 软件产品不仅要满足用户当前的要求，还要考虑到维护工作。在它的生命周期内必须保证维护工作简单，要具备功能扩展能力，且修改容易。

(4) 软件开发必须遵循规定的标准。开发过程的标准化有利于提高软件质量水平。

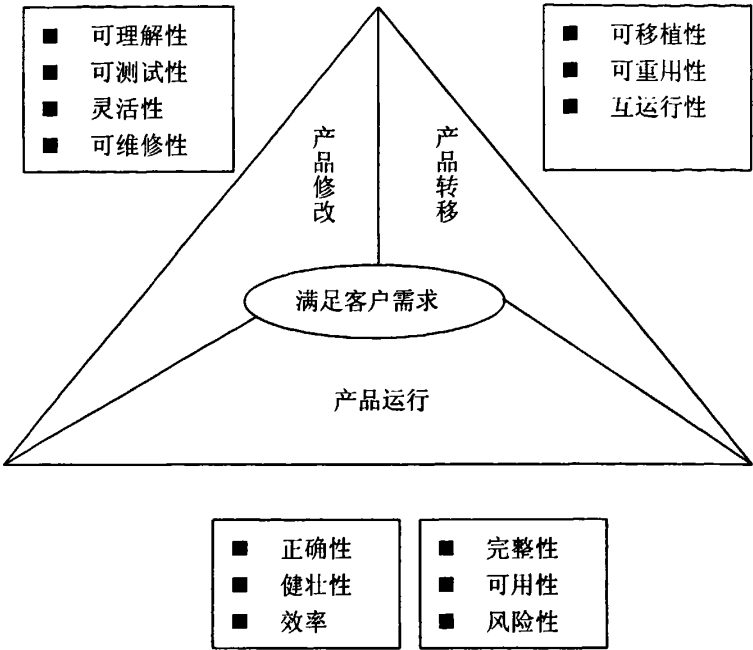


图 2.1 质量特性结构图

可见，从用户最感兴趣的角度来说，软件质量可以从三个方面来把握:如何使用软件、使用效果如何、软件性能如何；从软件开发的团队的角度来说，不仅要生产出满足质量要求的软件，也要能够运用最少的资源、以最快的进度生产出质量最优的产品；从软件维护者的角度看，要对软件维护方面的特性感兴趣；对企业的管理层来说，注重的是总体效益和长远利益，就是说质量好的软件一般可以帮助企业扩大市场；反之，质量差的软件一般会造成企业市场萎缩。

2.1.2 软件质量管理的基本内容

(1) 软件质量管理的概念

软件开发组织为了使自己提供给用户的软件产品达到并保持一定的质量水平，就必须进行严格的质量管理。软件质量管理就是在软件开发过程中开展以下质量活动：建立质量保证体系，确定各个岗位的职责和权限，并使其有效运行；制订质量保证计划，确定质量方针和质量目标；采取质量控制措施，发现和改正软件错误^[13]。

(2) 软件质量管理的内容

软件生产不同于硬件生产，其特点在于其产品无形、易变化、难于测量控制、维护复杂。如果在软件开发过程中仅仅依靠开发人员的个人创造力的发挥，而不建立规范的管理体系，没有严格的软件生产质量控制手段，软件质量将难于保证。软件质量管理的主要内容就是在软件开发机构中开展以下质量活动：建立质量保证体系、制订质量保证计划、采

用质量控制措施，统称为进行软件质量保证活动。

2.1.3 影响软件质量管理的因素

软件本身的特点和目前软件开发模式使隐蔽在软件内部的质量缺陷不可能完全避免，影响软件质量的因素是多种多样的^{[14][15]}。

(1) 软件需求模糊以及软件需求的变更从根本上影响着软件产品的质量。

软件是可视性很差的复杂的逻辑实体，不同于任何其他制造业的产品。使得软件质量难以把握的一个因素是软件需求。确定需求，在软件开发的初期阶段让它真正反映用户的意图是一件非常重要的工作，它既是后继阶段开发的基础，又是软件开发完成后验收的依据，而且还是工期和开发成本估计的出发点，供方和用户自然都十分关心它。但实际上，软件需求既不可见，也往往说不清。软件用户常常自己弄不清楚，或者说不明白自己对所要软件的需求，或者会出现用户提出的需求一变再变的情况。这种现象必定会给软件的开发工作带来许多困难，也就必定会埋下软件质量缺陷的隐患。

(2) 手工开发工作难以避免出现差错

目前软件开发工作大多仍是手工劳动，但又需要开发人员集中精力，全神贯注投入的智力密集性工作。对于这种复杂、细致而又可见性差的工作，出错的可能不容易完全排除。

(3) 软件质量管理的实际困难

软件质量指标许多尚未量化。软件开发的管理人员往往更关心项目开发的成本和进度，因为成本和进度是显而易见的，并且易于度量，而软件质量则完全不同。

目前许多软件机构的产品质量责任尚未落实到人。如果软件开发的管理人员对于交付的软件含有多少错误并不负什么责任，那么他们必定没有多高的热情去控制开发的质量，更不必说保证质量并不容易，且保证质量是昂贵的。

不规范的开发习惯难以纠正。取得高质量的软件产品，主要取决于参与开发的人员，然而软件开发人员的习惯一旦形成便难以改变，他们的行为也难以控制。

人员之间的沟通容易出现問題。许多软件工程项目需要若干甚至许多技术人员和管理人员参与。工作中他们之间信息和思想方面的交流和沟通是十分必要和频繁的，但问题也常常发生在互相交流中。对问题的不同认识和误解如不能及时消除，必定埋下影响产品质量的祸根。

软件项目组中人员流动会影响产品质量。软件项目组中人员的流动难以完全避免。事实表明，对于尚未建立成熟的软件过程的机构来说，从软件项目组离去的人员，特别是骨干力量，他们的离去会带走思想、技术和经验，这不能不说是一种损失，项目开发的质量也会受到影响。

2.1.4 软件质量管理存在的问题

(1) 软件质量管理尚未受到应有的重视

我国软件外包企业的管理普遍比较薄弱,其中质量管理尤其薄弱,这与长期以来不重视管理工作有关。分析质量管理薄弱的原因,其中包括认为重要的是技术,管理并不重要。长期以来,认识不到现代管理是一门重要的行为科学,因此需要我们投入力量在实践中学习和掌握它。特别是在市场竞争的环境下,企业如果不能在管理上付出努力,就很难在竞争中有所作为。我们软件企业的管理大大落后于欧美国家,致使为国际公认的我国的人才优势和雄厚的智力资源无法充分发挥。具有聪明才智的技术人员在非规范化的管理环境下不能为工程项目的大厦添砖加瓦,管理薄弱反而会使有序的活动成为无效的活动或者导致力最互相抵消。我国软件开发人员在外国企业中能够得到出色的发挥,这事实值得我们深思^[16]。

(2) 软件开发没有达到工程化

不少软件开发人员凭自己的经验,习惯了自己熟悉的一套开发方法和步骤,或者习惯了非规范、任意性很强的作法,不愿受规范化管理的约束。一些开发活动事先不作计划;活动过程中不作记录;项目临近结束补写资料;赶制文档。在开发进度由于各种原因延误的情况下,降低测试工作等要求追求进度等等。

(3) 开发规范中没有体现质量管理的基本要求

目前不少软件外包企业并未建立自己的开发规范,使得项目或产品的开发工作无章可循、无法可依。有的项目或产品开发获得成功完全是因为参与的技术人员或管理人员优秀,但他们的成功作法并未得到推广。特别是当他们由于各种原因一旦离开岗位以后,工作情况会发生很大滑坡^[17]。

(4) 软件企业的组织结构有待合理化

一些软件开发企业的人员组织结构不够合理,责任不够明确,制度不够严密。无疑这一情况必定会表现在产品质量上。

2.2 软件质量保证体系

软件质量管理和质量保证工作应该不断创新,适应形势发展需要,主动将全面质量管理和质量改进思想纳入质量管理和质量保证计划,使软件质量提高到新的水平。就整体而言,我国的软件质量管理和质量保证工作仍处于创建阶段,国外的软件质量管理和质量保证工作也不尽完善,因此在此只介绍一些业已成熟的软件质量管理与保证理论。

2.2.1 基于 CMM 的软件质量管理与保证体系

1987 年,美国 Canegie-Mellon(卡内基-梅隆)大学的软件工程研究所(SEI, Software Engineering institute)以 W S.Huphrey 为首的专家组提出了“描述软件过程的特征:一种成熟度框架”的专题报告。将软件企业产品开发过程的成熟度分为五级,从 1 级到最高的 5 级。提出了由 101 个问题构成的提问表,根据这些问题的打分综合评估软件企业达到的级别,并可分析指出需改进的主要差距。1991 年在此报告及实践的基础上形成了 SEI 的软件

能力成熟度模型(CMM)(Capability Maturity Model)简称为 CMMI1.0 版,到 1993 年改进为 CMMI.1 版,以后又采用了 ISO 的软件评估标准 SPCE 的一些方法和内容改进为 CMM2.0,但其基本框架没有改变^[18]。

1. CMM 现状

CMM 是目前国际上最流行也是最实用的一种软件生产过程标准,现已经得到了众多国家软件产业界的认可,成为当今从事规模型软件生产不可缺少的一项内容。众所周知,如今正处在一个全球性竞争空前激烈的时代,其中信息系统对于各个企业以至企业集团的生存与发展都具有举足轻重的意义。这样一来,就对企业信息系统的开发提出了更加严格和苛刻的要求。例如要求所开发的信息系统更能满足企业战略决策的需要,其中包括缩短开发周期,提高开发效率,扩大实现发展战略的要素等。随之而来的是信息系统的多样化、复杂化,其中包括必须面对需求的多样化和复杂化,以及信息技术的迅速进步。所有这些都对信息系统的开发与管理提出了更高的要求。如何解决这些日益复杂的课题,结论只有一个,就是要实行软件开发生产组织的变革,实现软件开发的标准化,提高软件的生产效率,真正实现软件的工业化。所以,研究、讨论和实践 CMM 模型及其实现方法既十分必要,又合乎当今信息产业的发展潮流。

自 70 年代出现软件危机以来,学术界和企业界对软件工程环境、工具和技术的研究都倾注了大量的人力、财力和物力,多年来也取得了许多成果。但是不可否认的一个事实是软件开发与生产的效果却始终未如人意。究其原因,一个不同于以往的概念逐渐被接受:一个企业的软件能力更取决于该企业的过程能力,特别是在软件开发和生产中的成熟度。一个企业过程能力越是成熟,该企业的软件生产能力就越有保证。提高软件开发与生产的成熟度,可以使新技术的应用收到更好与更快的效果。这主要是因为基于 CMM 模型的软件成熟度实践要求尽量采用更加规范的开发标准和方法,使用更加科学和精确的度量方法,选择便于管理和使用的开发工具。所有这些,都造成了整个工程的可重构性、可分解性和最优化,从而进一步明确了整个项目中必要和不必要的工作,明确了整个项目的风险,以及各个阶段进行评估的指标与应急措施。此外 CMM 也不仅仅应用于软件开发组织内,它也可作为认证机构的认证工具 and 用户评测一个企业是否达到所要求的能力的依据。

2. CMM 的原理

CMM 是一个软件工程实践的纲要,以逐步演进的架构形式不断地完善软件开发和维护过程。CMM 具备变革的内在原动力,与静态的质量管理系统标准 ISO9001(International Organization for Standardization, 1987)形成鲜明对比。ISO9001 在提供一个良好的体系结构与实施基础方面是很有效的;而 CMM 是一个演进的、有动态尺度的标准以驱使一个组织在当前的软件实践中不断地改进完善。基于 CMM 模型的软件成熟度实践要求:尽量采用更加规范的开发标准和方法,使用更加科学和精确的度量方法,选择便于管理和使用的开发工具。所有这些,形成了整个工程的可重构性、可分解性和最优化,使整个项目进一步明确了必要和不必要的工作,明确项目的风险,明确各个阶段进行评估的指标与应急措施。

在 CMM 模型及其实践中，企业的软件过程控制管理能力被作为一项关键因素。所谓软件过程控制管理能力，是指把企业从事软件开发和生产的过程本身透明化、规范化和运行的强制化。所设定的过程可能有问题，但问题会在执行的过程中反映出来;企业在过程执行一段时间后，根据反映的问题可以来改善这个过程。周而复始，使过程逐渐完善、成熟。因此，项目的执行不再是黑箱，企业清楚地知道项目是否按照规定的过程进行。软件开发及生产过程中成功或失败的经验教训可为后续的项目所用，从而大大加快软件生产的成熟程度提高。

软件过程成熟度是指针对具体的软件过程进行明确定义、管理、测量、控制以及有效的程度。成熟度级别分别是严格定义的。每个成熟度级别由一组过程目标集合组成，满足这些目标，就能确定软件过程的重要组成部分。达到了每个成熟度级别，就建立了软件过程的不同组成部分，从而使企业的过程控制管理能力提高。

每个成熟度级别都包含几个关键过程域(KPA)。关键过程域确定了实现一个成熟度级别所必须解决的问题。

如表 2.1 是 CMM 中五个软件过程成熟度的结构，及与其相关的 18 个关键过程域(Key Process Areas),每一个 KPA 都与一些目标相关，代表某种对过程的要求。除了第一级，CMM 的每一级都按相同的结构组成，KPA 不仅标明了某一级成熟度所要求的目标和评估标准，也说明了要达到此级成熟标准所需解决的具体要点^[19]。

表 2.1 CMM 模型

级别	关键过程域
5 级—优化级	过程更改管理、技术改革管理、缺陷预防
4 级—已管理级	软件质量管理、定量过程管理
3 级—已定义级	同行评审、组间协调、软件产品工程、集成软件管理、培训活动、组织过程定义、组织过程焦点
2 级—可重复级	软件配置管理、软件质量保证、软件子合同管理、软件项目跟踪和监控、软件项目计划、需求管理
1 级—初始级	

2.2.2 基于 ISO9000 的软件质量保证体系

1. ISO9000 系列标准简介

国际标准化组织制定的 ISO9000 系列标准是国际公认的质量管理和质量保证标准，被世界各国和地区广泛采用。按照该系列标准的定义，产品是“活动或过程的结果”，“产品包括服务、硬件、流程型材料、软件或它们的组合。”所以毫无疑义，软件企业、软件产品的质量管理和质量保证，也应该遵循这套标准。这套标准源于国际标准化组织(英文缩写为 ISO)1986 年发布的 ISO8402 “质量术语”和 1987 年发布 ISO9000 “质量管理和质量保

证标准—选择和使用指南”、ISO9001 “质量体系—设计开发、生产、安装和服务的质量保证模式”、ISO9002 “质量体系—生产和安装的质量保证模式”、ISO9003 “质量体系—最终检验和试验的质量保证模式”、ISO9004 “质量管理和质量体系要素—指南”等 6 项国际标准，统称为系列标准，或称为 1987 版 ISO9000 系列国际标准。

1990 年负责制定 ISO9000 系列标准的 ISO/TC176 质量管理和质量保证技术委员会决定对 1987 年版的 ISO9000 系列的 6 项标准进行修订，并采纳 1987 年最初提出的 ISO9000 系列标准的修订战略，将这次修订分为两个阶段进行。第一阶段称之为“有限修改”，即在标准结构上做不大的变动，仅对标准的内容进行小范围修改，但这种修改要趋向于将来的修订本，以便更好地满足标准使用者的需要。

1994 年 ISO/TC176 完成了对标准的第一阶段的修订工作，ISO 发布 T1994 版 ISO8402，ISO9001-1，ISO9001，ISO9002，ISO9001 和 ISO9004 等 6 项国际标准，统称为 1994 版 ISO 标准。这些标准分别取代 1987 版的 6 项标准。

1994 年发布 ISO9000 族修订本时，ISO/TC176 提出了“ISO9000 族”的概念，“ISO9000 族”是指由 ISO/TC176 制定的所有国际标准。ISO 在发布上述 6 项质量标准时，已陆续制定发布了 10 项指南性国际标准。这样，ISO9000 族国际标准就从 1987 年仅有的 6 项发展到 1994 年的 16 项。其中包括 ISO9001-3；1991 “质量管理和质量保证标准—第 3 部分：ISO9001 在软件开发、供应和维护中的使用指南。”这个指南是专门针对软件的质量管理和质量保证而制定的，对软件企业和软件产品的质量管理和质量保证具有重要的意义。ISO/TC176 在 1994 年完成对标准的第一阶段的“有限修改”工作后，随即启动修订战略第二阶段的工作，称之为“彻底修改”。1996 年，在广泛征求标准使用者意见、了解顾客对标准修订的要求、比较各种修改方案后，相继提出了“2000 版 ISO9001 的标准结构和内容的设计规范”和“ISO9001 修订草案”，作为 1994 版修订标准的依据。在 2000 版 ISO9001 里面引入了全面质量管理的概念，全面质量管理的基本工作方法是 PDCA 循环，即由计划 (PLAN)、实施 (DO)、检查 (CHECK) 和处理 (ACTION) 这四个密切相关的阶段所构成的工作方式。国际质量管理学界普遍认为，PDCA 循环是一个非常科学的工作方式，引进 PDCA 循环，即过程方法模式，对质量保证有很大的益处^{[20][21]}。

2. ISO9001-3

ISO9001-3 是国际标准化组织关于软件质量管理和保证而制定的国际标准因此，它适合合同环境下的软件开发质量的保证。在这里，提出了较完整的质量体系要素。质量体系是由质量体系要素构成的，在 ISO9001 中一共有 20 个质量体系要素，这些体系要素主要是针对硬件而设计的，对软件不完全适用。在 ISO9001-3 中针对软件的特点将软件的质量体系要素区分为三种类型，设计了 22 个体系要素，其中结构类型要素 4 个，生命周期活动类型要素 9 个，支持活动类型要素 9 个。

(1) 结构类质量体系要素共 4 个，它们是：①领导的责任；②质量体系的建立和运行；③内部质量体系审核；④纠正措施。

(2) 寿命周期类质量体系要素共 9 个, 它们是: ①合同评审; ②需方要求规范; ③开发策略; ④质量策划; ⑤设计和实施; ⑥试验和确认; ⑦验收; ⑧复制交付和安装; ⑨维护。

(3) 支持活动类型质量体系要素共 9 个, 它们是: ①技术状态管理; ②文件控制; ③质量记录; ④测量; ⑤规则和惯例; ⑥工具和方法; ⑦采购; ⑧配套的软件产品; ⑨培训。

很明显, 不管是 CMM 还是 ISO 都强调对产生应用软件之过程的管理, 提高软件产品的生产效率和软件的质量, 同时, 软件工程理论的广泛运用也推动了软件产业由小规模生产到集成自动化生产迈进。这也充分说明, 软件产品的质量不仅表现在最终产品的质量, 还应该包含软件产生过程的质量, 只有这样, 才能使软件组织连续不断地生产出高质量的软件产品^[22]。

另外, 从管理层次上看, ISO 要比 CMM 所处的级别高, ISO 只是提出了一个质量管理框架, 是属于指导性的框架, 而 CMM 提出的框架是一个操作性很强的框架, 其 KPA 过程非常明确地提出了过程目标和过程注意事项。因此, 在软件组织中, 可以将 ISO 和 CMM 结合起来应用, 即: 把 ISO 作为软件质量管理的指导性框架, 把 CMM 作为具体实施层的应用, 这样就可以充分利用二者的优势来共同完成对软件开发过程的质量控制, 从而达到既提高软件开发效率又保证所开发的软件具有较高的质量。目前, 虽然 ISO 和 CMM 在国外的应用已经有了十几年的历史, 是一套比较实用的质量控制和保证方法, 但其在国内应用也只有 5, 6 年的光景, 并且应用水平很低, 原因是多方面的, 其中一个主要原因就是没有形成一套适合我国软件组织的软件质量控制方法, 这也是阻碍我国软件企业发展的瓶颈之一^[23]。

2.3 软件度量对软件质量的控制

1. 度量软件过程和项目控制软件质量

软件度量可以帮助项目经理更好地工作。它帮助定义和执行更加实际的计划, 正确地分配宝贵的资源以使那些计划能够到位, 并按计划准确地监控进展和性能。软件度量为做出关键项目决策和采取适当行动提供了所需要的信息。度量帮助管理者在组织的各个层次上标识、设定优先级、跟踪和交流目标以及相关的问题。度量可准确地描述软件项目过程和产品的状态。它对于在项目行使周期中客观地描绘项目活动的进展以及相关软件产品的质量非常关键。度量促进了主动性的管理战略。度量促进在早期发现并纠正技术和管理上的问题, 这些问题如果留待以后将会更加困难并且代价昂贵。费用、进度、能力、技术质量和性能都是影响软件项目质量的因素, 通过度量项目经理可以在系统设计之前了解系统的规模、可能需要的人力和时间、系统的结构等信息, 在系统设计好之后了解系统的数据复杂度、任务功能划分情况等, 从而较好的分配人力和物力, 优化软件项目和产品性能。通过度量, 软件工程师可以知道软件的哪个部分更容易出错并对其进行强度较大的测试。

2. 分析度量数据进行软件质量评估

对于一个开发好的软件, 我们事后可以对软件的质量进行度量, 以对该软件各方面的

质量要素进行测定，从而定量的评估这个软件的整体质量。定量评估就需运用软件度量学的理论，因此，软件度量学对软件质量的保证和控制起着至关重要的作用。

2.4 本章小结

本章首先介绍了软件质量的定义，软件质量管理的内容，影响软件质量管理的因素及软件质量管理存在的问题，然后简要介绍两种软件质量保证体系：基于 CMM 的软件质量管理与保证体系和基于 ISO9000 的软件质量保证体系。

第三章 软件质量度量模型

在整个软件生存周期中, 为了实现软件质量控制, 提高软件质量, 必须评估软件质量需求是否得到满足。软件度量提供了一个定量的方法来评价产品内部属性的质量, 能够在软件产品完成之后进行质量评估, 因而减少了软件质量评估中的主观性。软件质量度量就是从整体上度量软件质量, 用于软件开发过程中对软件进行质量控制, 并最终对软件产品进行评价和验收^[24]。

3.1 软件质量度量方法

IEEE Std1061 软件质量度量方法学提供了系统地进行软件质量度量的途径, 包括建立某个软件系统的质量需求、标识、实现、分析并确认该软件的质量度量过程。该方法学跨越整个软件生存周期, 并包括下列 4 个步骤^[25]:

1. 建立软件质量需求

质量需求表达了在具体应用的特定环境下对软件产品质量的定量要求, 应该在软件开发前或初期进行定义, 它是有效构造软件质量和客观评价质量的前提。质量需求规格说明可定量定义在所需质量特性的直接度量及其直接度量目标值。直接度量用来验证最终产品是否达到了质量需求。

2. 准备度量

由软件质量特性和子特性描述的软件质量需求常常无法直接测量, 需要进一步确定相关的度量元。在度量的准备阶段, 应根据应用环境, 为软件开发的各个阶段和其最终产品分别确定适当的度量元、建立度量元、质量子特性、质量特性的映射模型, 确定合理的评估准则。

3. 实现软件质量度量

数据收集过程规定从数据收集点到度量评价的数据流程, 确定有关数据的收集条件, 给出工具的使用说明及数据存放规程。在全面实施度量前, 最好首先在小范围内试验数据收集和度量计算规程, 分析其数据是否一致、度量要求是否确切, 尤其要检查主观判断的数据说明和要求是否清晰; 检查样板度量过程的费用, 修改或完善费用分析; 检查所收集到的数据的准确性、度量单位的合适性、所收集到的数据之间的一致性, 确认数据样本的随机性、最小样本数、相似性等。

4. 分析质量度量结果。

分析并报告度量结果不仅要做出度量和评估的结论, 还要进行度量元的确认, 从而确定哪些度量元的确适用于当前软件质量度量活动并可以用来预测软件质量特性值, 根据这些度量值和由此而计算得到的直接度量的预测值决定被度量对象是否需要做进一步的度

量和分析。

5. 确认软件质量度量

把预测的度量结果与直接度量结果进行比较，以确定预测的度量是否准确地测定了它们的相关质量要素。

3.2 软件质量度量模型

软件质量模型是软件质量评价的基础，软件质量模型代表了人们对软件质量特性的认识程度和理解程度，也代表了软件质量评价研究的进展状况^[26]。

当前的软件质量模型有很多种，比较常见的有五种：McCall 模型，Boehm 模型，FURPS 模型，ISO/IEC9126，Dromey 模型。

3.2.1 McCall 模型

1979，McCall 提出软件质量模型，把软件质量进行基于 11 个特性之上，而这 11 个特性分别面向软件产品的运行、修正和转移，如图 3.1 所示^[27]。

- (1) 正确性：一个程序满足她的需求规约和实现用户任务目标的程度。
- (2) 可靠性：一个程序满足一所需的精确度完成它的预期功能的程度
- (3) 效率：一个程序完成其功能所需的计算资源和代码的度量。
- (4) 完整性：对未授权人员访问软件或数据的可控制程度。
- (5) 可用性：学习、操作、准备输入和解释程序输出所需的工作量。
- (6) 可维护性：定位和修复程序中一个错误所需的工作量。
- (7) 灵活性：修改一个运行的程序所需的工作量。
- (8) 可测试性：测试一个程序以确保她完成所期望的功能所需的工作量。
- (9) 可移植性：把一个程序从一个硬件和/或软件系统环境移植到另一个环境所需的工作量。
- (10) 可复用性：一个程序可以在另外一个应用程序中复用的程度。
- (11) 互连性：连接一个系统和另一个系统所需的工作量。

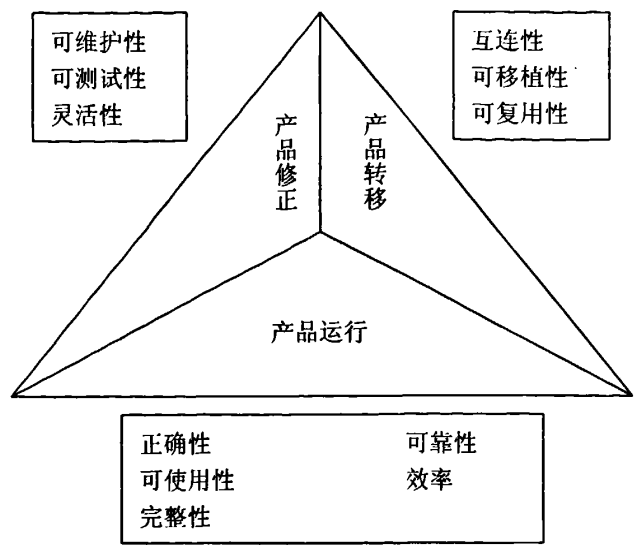


图 3.1 McCall 模型

McCall 等又给出了一个三层次模型的质量度量框架。如图 3.2 所示。McCall 等人认为，要素是软件质量的反映，软件属性可用做评价的准则，定量化地度量软件属性可知软件质量的优劣^[28]。

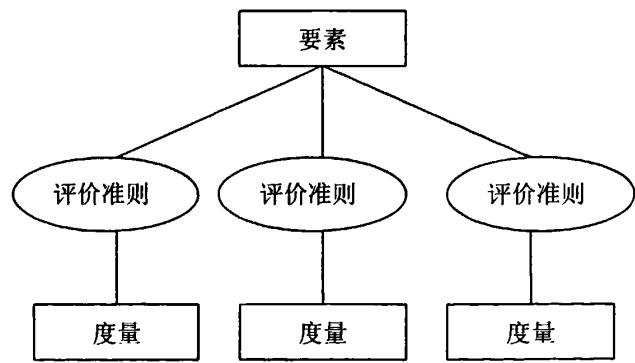


图 3.2 McCall 质量度量模型框架

- McCall 定义的软件质量要素评价准则共 21 种，它们是^[29]：
- (1) 可审查性(audit ability)。检查软件需求、规格说明、标准、过程、指令、代码及合同是否一致的难易程序。
 - (2) 准确性(accuracy)。计算和控制的精度，是对无误差程序的一种定量估计。最好表示成相对误差的函数。值越大表示精度越高。
 - (3) 通信通用性(communication commonality)。使用标准接口、协议和频带的程序。
 - (4) 完全性(completeness)。所需功能完全实现的程度。在软件开发项目中一致的设计和文档技术的使用。
 - (5) 简明性(conciseness)。程序源代码的紧凑性。

- (6) 一致性(consistency)。在软件开发项目中一致的设计和文档技术的使用。
- (7) 数据通用性(data commonality)。在程序中使用标准的数据结构和类型。
- (8) 容错性(error-tolerance)。系统在各种异常条件下提供继续操作的能力。
- (9) 执行效率(execution Efficiency)。程序运行效率。
- (10) 可扩充性(expandability)。能够对结构设计、软件质量度量数据设计和过程设计进行扩充的程度。
- (11) 通用性(generality)。程序部件潜在的应用范围的广泛性。
- (12) 硬件独立性(hardware independence)。软件同支持运行的硬件系统不相关的程序。
- (13) 检测性(instrumentation)。监视程序的运行，一旦发生错误时，标识错误的程序。
- (14) 模块化(modularity)。程序部件的功能独立性。
- (15) 可操作性(operability)。操作一个软件的难易程度。
- (16) 安全性(security)。控制或保护程序和数据不受破坏的机制，以防止程序和数据受到意外的或蓄意的存取、使用、修改、毁坏或泄密。
- (17) 自文档化(self-documentation)。源代码提供有意义文档的程度。
- (18) 简单性(simplicity)。理解程序的难易程度。
- (19) 软件系统独立性(software system independence)设计语言特征、操作系统特征以及其他环境约束无关的程度。
- (20) 可追踪性(traceability)。从一个设计表示或实际程序构件中学到需求的能力。
- (21) 易培训性(training)。软件支持新用户使用该系统的的功能。

软件质量要素与评价准则之间的关系如下表所示。于是，软件质量要素 F_j 的值可用下式计算：

$$F_j = \sum_{k=1}^L C_k M_k \quad j=1, 2, 3, \dots, 11$$

其中： M_k 是软件质量要素 F_j 对第 k 种评价准则的测量值， C_k 是相应的加权系数。由于 McCall 定义的评分准则多数都没有客观的测量方法，所以只能凭主观印象为其评价定值。McCall 将评分准则都分为 0-10 级。0 级最低，10 级最高。因此， M_k 的取值可以是 0, 0.1, 0.2, 0.3, ..., 1.0。

加权系数 C_k 满足： $\sum_{k=1}^L C_k = 1$ 。其中， $C_k \geq 0$ ，当质量要素 F_j 与 k 项评价准则无关时，

$C_k = 0$ ， L 表示评价准则的项数。对 McCall 的评价准则， $L=21$ 。

3.2.2 Boehm 模型

Boehm 模型着手于软件总体的功效，也就是说，对于一个软件系统而言，除了有用性以外，它的开发过程必定是一个时间，金钱和能量的消耗过程。考虑到系统交付时使用它的用户类型，Boehm 模型从几个维来考虑软件的效用。总功效可以被分解成可移植性，有效性，可维护性。其中，有效性可以细分为可靠性，效率，运行工程；可维护性可以细分为测试性，可理解性，可修改性。例如，从 Boehm 的模型可知，可维护性能从可测试性，可理解性及可修改性来度量，即高可维护性意味这高可测试性，高理解性和高可修改性。具体模型如图 3.3 所示^[30]。

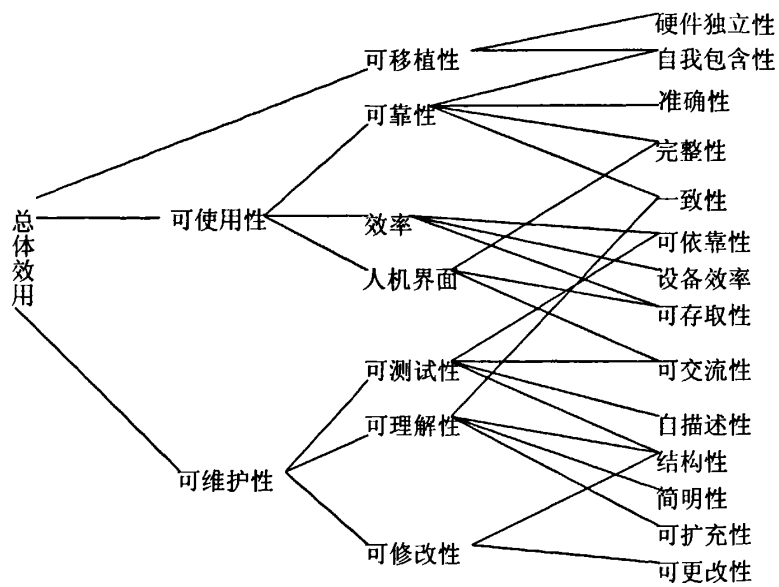


图 3.3 Boehm 模型

3.2.3 FURPS 模型

McCall 和他的同事提出的质量要素代表了被提出的众多软件质量“检查表”之一，Hewlett-Packard 提出了一套考虑软件质量的因素，简称为 FURPS—功能性 functionality)，可用性(usability)，可靠性(reliability)，性能(performance)和支持度(supportability)。质量因素是从早期工作中的得出的，五个主要因素每一个都定义了如下评估方式^{[31][32][33]}：

- (1) 功能性：通过评价特征集和程序的能力、交付的函数的通用性和整体系统的安全性来评估。
- (2) 可用性：通过考虑人的因素、整体美学、一致性和文档来评估。
- (3) 性能：通过度量错误的频率和严重程度、输出结果的准确度、平均失效间隔时间、

从失效恢复的能力、程序的可预测性等来评估。

(4) 性能：通过处理速度、响应时间、资源消耗、吞吐量和效率来评估。

(5) 支持度：包括扩展程序的能力(可扩展性)、可适应性和服务性(这三个属性 代表了一个更一般的概念—可维护性)、以及可测试性、兼容度、可配置性(组织和控制软件配置的元素的能力)、一个系统可以被安装的容易程度、问题可以被局部化的容易程度。FURPS 质量因素和上述描述的属性可以用来为软件过程中的每个活动建立质量度量。

3.2.4 ISO/IEC9126 软件质量模型

1985 年，国际标准化组织(ISO)建议，软件质量度量模型由三层组成。在 ISO1985 中，高层称软件质量需求评价准则(SQRC)，中层称软件质量设计评价准则(SQDC)，低层称软件质量度量评价准则(SQMC)。分别对应 McCall 等人的质量要素、评价准则和度量。ISO 认为应对高层和中层建立国际标准，在国际范围内推广应用软件质量管理(SQM)技术，而低层可由各使用单位自行制定。ISO 高层由 8 个质量要素组成、中层由 23 个评价准则组成。

ISO/IEC9126 中，将高层要素减少到 6 个，更名为软件质量特性，中层 23 个评价准则减少到 21 个，更名为软件质量量子特性。这些质量特性与子特性的关系如表 3.1 所示^[34]。

表 3.1 质量特性与子特性的关系

质量特性	质量量子特性
功能性（Functionality）	实用性，准确性，互操作性，一致性，安全性
可靠性（Reliability）	健壮性，容错性，可恢复性
可使用性（Usability）	可理解性，可学习性，可操作性
效率（Efficiency）	时间性能，资源性能
可维护性（Maintainability）	可分析性，可修改性，稳定性，可测试性
可移植性（Portability）	适应性，可安装性，可替换性

ISO/IEC9126 认为：软件质量特性可以精确到多层子特性。可以为每一质量特性定义一组子特性，这些子特性是软件产品或软件和过程的独立特性；然后对每一质量量子特性定义一组度量，利用这些度量对质量量子特性进行定量测量，进而达到一个新的水平定量度量软件质量的目的。因此，软件质量度量模型是一个树形结构，将软件质量划分为质量特性、质量量子特性、度量三层次。如图 3.4 所示^[35]。

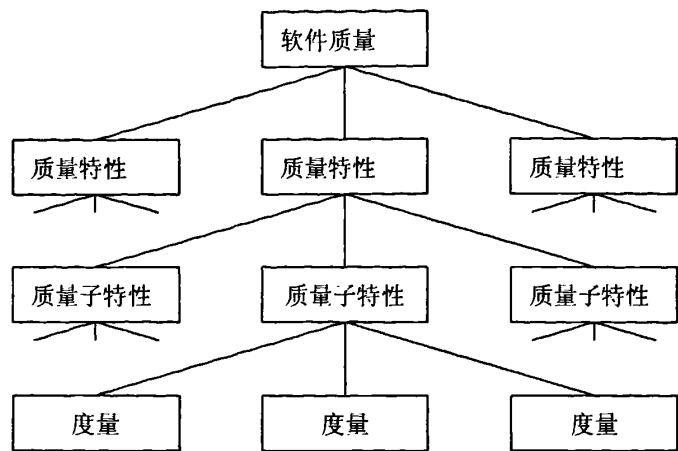


图 3.4 ISO9126 软件质量度量模型

3.2.5 Dromey 模型

Dromey 提出了一个工作框架，建立和使用一个应用质量模型来评价软件需集，设计阶段和开发阶段。Dromey 指出，一个高层次的质量属性，比如可移植性和可维护性不能直接建成于软件中，这些属性是通过一系列的子特性来实现的。Dromey 提出了三个基于软件产品开发过程的模型：需求模型，设计模型和实施质量模型^[36]。

3.3 新的软件质量度量模型

3.3.1 模型分析与比较

通过分析以上五种模型，可以看出，他们各有优缺点。McCall 模型的最大贡献在于，它建立了软件质量特征和软件度量项之间的关系，但是它有些度量项不是客观指标，而是主观判断。另外它没有从软件生存周期不同阶段的存在形态来考虑，而仅仅考虑一种产品形态，不利于软件产品早期的缺陷发现和维护成本的降低。Boehm 模型与 McCall 模型相似，也是一种由纵向软件特征构成的层次模型，唯一的差别在于特征的种类，另外，Boehm 模型包括 McCall 模型没有的硬件领域的质量要素。FURPS 模型的缺点在于没有考虑软件产品的重用性或是可移植性，而对于现代软件产业来说，产品的易移植性非常重要，它直接影响到软件产业的经济规模问题。ISO9126 模型的贡献在于将软件质量特征分为外部特征和内部特征，考虑到软件产品不同生命周期阶段的不同形态问题，但是该模型没有清楚给出软件质量特征如何去度量。Dromey 模型意在建立软件质量特征和子特征之间的关系，同时意图明确影响软件质量特征的产品属性，但是，并未清楚的给出^{[37][38][39][40]}。

表 3.2 五种软件质量度量模型的比较

度量标准/目标	McCall	Boehm	FURPS	ISO9126	Dromey
正确性	*	*		*	
可靠性	*	*	*	*	*
完整性	*			*	
可用性	*	*		*	
效率性	*	*	*	*	*
可维护性	*	*	*	*	*
可测试性	*			*	
互操作性	*				
适应性	*	*	*		*
可重用性	*	*			
可移植性	*	*		*	
明确性		*			
可变更性		*		*	
文档化		*		*	*
过程成熟			*		
恢复力		*			
易懂性		*			
有效性		*			
功能性			*		
普遍性		*			
经济性		*			

注：其中“*”表示该模型提出的影响软件产品质量的因素

在这些模型中，软件质量特征被定义为：功能性，可靠性，可维护性，应用性，效率和可移植性。这些特征被认为是软件质量基本特征并值得去研究。但是，软件质量模型应建立在实用的基础上，在实际的开发中，应该根据不同系统的软件的不同要求做不同的选择，比如，航空航天用的软件系统，它的可靠性是一个至关重要的问题，对于商业领域中的软件，可维护性则是关键的问题，而在实时系统中，效率尤为重要。因此，在建立软件质量模型时，应该考虑以下几个方面：

- (1) 软件的类型不同，所考虑的质量要素侧重点不同，同时，商业目标决定质量目标，能满足客户需求的软件就可以称为好的软件。
- (2) 在考虑软件成品的同时，要考虑软件过程的质量保证。
- (3) 结合全面质量管理的观点，应该考虑人的因素，包括开发商和用户。

3.3.2 基于动态权重的软件质量度量模型的提出

为了科学的度量软件的质量，不仅需要详细的软件质量度量框架图，而且必须有一种合适的度量方法去对软件质量进行定量化度量，在此提出一个二层结构的基于动态权重

的软件质量度量模型。这个模型分为二层，第一层是度量框架层，可以根据它建立评价指标体系，这个框架分为三层，第一层为质量特性，它们是面向管理者和用户的质量属性；第二层为质量子特性，这些子特性体现了面向软件的质量属性；第三层是度量，它的结果定量反映软件质量的好坏。模型的第三层为度量方法层，在第一层建立的评价指标体系的基础上，对软件质量进行综合定量评价。其中较为关键的是指标权重的确定。可以看出，第一层是第二层的输入，也是第二层的基础。新模型如图 3.5 所示。

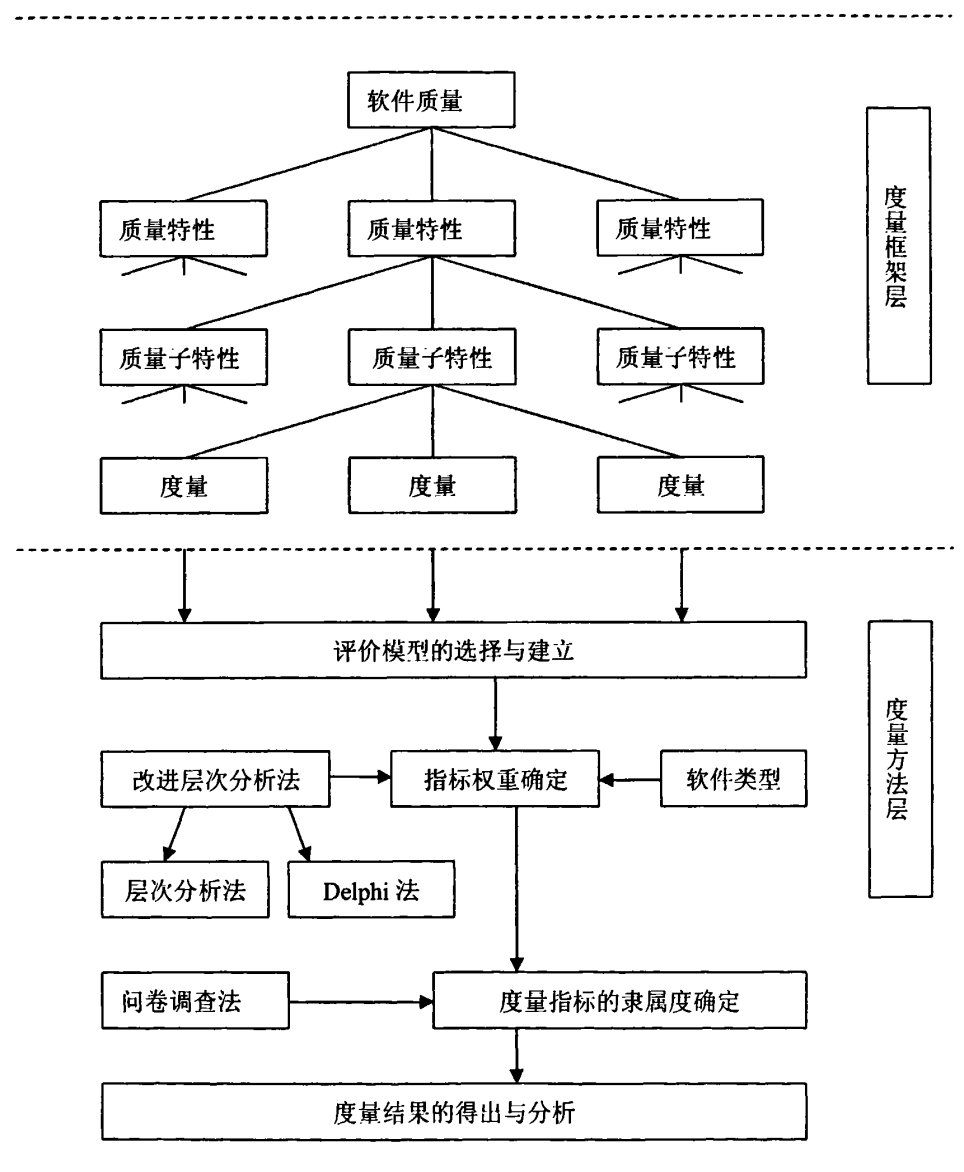


图 3.5 基于动态权重的软件质量度量模型

3.4 本章小结

本章介绍了软件质量度量的方法，并详细介绍了 5 种常用的软件质量度量模型，对这几种模型进行了分析和比较，最后结合这几种模型，提出了一个全新的软件质量度量模型。

第四章 中小企业软件质量度量模型的分析

4.1 实施软件度量的策略

Grady 和 Caswell 在惠普公司实施软件度量程序的经验提供了有价值的信息,可以帮助我们在组织内部实施软件度量。Grady 和 Caswell 给出了实施软件度量程序的 10 个步骤^[41]:

- (1) 定义软件程序的目标
- (2) 明确责任
- (3) 开展调查
- (4) 定义要收集的初始度量
- (5) 推荐这些度量的初始样本
- (6) 获取用于自动数据收集和分析的工具
- (7) 进行软件度量培训
- (8) 宣传成功实例
- (9) 建立度量数据库
- (10) 以有序的方式建立一个改进此过程的方法

质量的改善既需要开销又能节省开支。开销用于确定和收集度量,费用节省来自于产品发布后较低的保障费用,较少的维护次数,较高的用户满意度,销售量的增加。

4.2 建立软件质量评估体系

4.2.1 建立软件质量度量框架

(1) 框架定义

IEEE 标准 1061-1998 定义了建立软件质量度量框架的方法学。如表 4.1 所示^[42]。

表 4.1 来自 IEEE 标准 1061-1998 的定义

度量框架	一种用来组织、选择、沟通、评价软件系统要求的质量属性的辅助决策法。它逐层分解为特性、子特性和度量
质量特性	一个与质量有关的面向管理的软件属性
质量子特性	质量特性分解出来的技术组件
直接度量	一种不依赖于任何其他属性测量的度量
软件质量度量	一个函数,他的输入是软件数据,输出是一个单一数值,它可解释为给定的软件属性对其质量的影响程度
产品度量	一种用来测量软件开发过程中任何中间或最终产品特性的度量

(2) 度量框架

本文使用的软件质量度量框架如图 3.5 所示。该框架是一个包括 3 层的树形结构。第一层是软件产品必须满足的质量要求。这些要求通常是用用户的术语来表示的，例如^[43]：

- ① 产品在用户当前使用的平台和操作系统上运行
- ② 产品是可靠的并能提供防止数据丢失的机制
- ③ 产品提供完成某种任务所需的功能
- ④ 产品易于使用
- ⑤ 产品有较高的利用率
- ⑥ 产品易于维护

质量特性代表了不同用户的质量要求，如表 4.2 所示。

表 4.2 与质量需求相关联的质量特性

质量需求	质量特性	描述
产品在用户当前使用的平台和操作系统上运行	可移植性	与软件可从某一种环境转移到另一种环境的能力有关的一组属性
产品是可靠的并能提供防止数据丢失的机制	可靠性	与在规定的一段时间和条件下，软件维护其性能水平的能力有关的一组属性
产品提供完成某种任务所需的功能	功能性	与一组满足明确或隐含的用户需求功能及其指定的性质有关的一组属性
产品易于使用	可使用性	与使用（包括使用前的准备和结果评估）所需的努力和用户对所使用所作的个人评价有关的一组属性
产品有较高的利用率	效率	在规定的条件下，软件的性能水平与使用资源之间有关的一组属性
产品易于维护	可维护性	与进行指定的修改所需努力有关的一组属性

软件的质量特性之间还存在着相互有利和不利的影晌，如表 4.3 所示。正是由于这些特性之间存在相互关系，使得没有一种软件的设计能够使软件所有的质量特性都是最佳的。所以，在实际开发过程中，应该根据软件的不同类型和不同的要求，对不同的软件质量特性给出不同的权值，做出不同的选择^{[44][45]}。

表 4.3 各质量特性间的有利影响和不利影响

质量特性	可使用性	可靠性	可维护性	可移植性
可使用性		*		
可靠性				
可维护性	*	*		
可移植性			*	

注：*表示有利，空表示无直接影响

框架的第二层表示质量子特性。通过将每一个质量特性分解为可以测量的属性，就可

以得到这些子特性。与表 4.2 所列的质量特性有关的质量子特性列于表 4.4 中。

表 4.4 与质量特性相关联的质量子特性

质量特性	质量子特性	描述
功能性	可重用性	程序或程序的一部分在其他程序中应用的程度
	依从性	遵循有关规定的程度
	准确性	软件在计算和控制时能否得到正确结果的能力
	互操作性	连接一个系统与另一个系统的难易程度
	安全性	保护和控制软件和数据的能力
可靠性	正确性	软件满足需求及用户目标的程度
	完整性	对未授权人员访问软件或数据行为的控制程度
	健壮性	软件中的错误能随着时间自行消除的能力
	容错性	软件遇到错误时仍能维持性能的能力
	可恢复性	软件在失效发生后能否恢复到失效前的数据的能力
效率	时间性能	软件的处理时间和响应时间
	资源性能	软件利用资源的效率
可使用性	可理解性	用户理解怎样使用软件的难易程度
	可学习性	用户学习使用该软件的难易程度
	可操作性	用户操作软件的难易程度
	可吸引性	界面的友善程度
可维护性	可分析性	诊断缺陷或失效原因的难易程度
	可修改性	对软件中的错误进行修改的难易程度
	稳定性	软件在环境改变的情况下继续维持其性能的难易程度
	可测试性	对软件测试使其无错误的难易程度
可移植性	适应性	软件在改变环境后说具有的适应能力
	可共存性	软件安装后与其他软件的共存能力
	可安装性	安装该软件的难易程度
	可替换性	软件被其他软件替换的难易程度

这些质量子特性之间并不是相互独立的，它们之间有的存在正向的关系，有的存在着反向的关系。例如可理解性和可分析性和可修改性存在正向的关系，高效率与可靠性和适应性存在着反向的关系。它们之间相互影响，相互牵制，所以要考虑到软件具体的应用特性^[46]。

框架的第三层是度量特性。一个度量特性至少与一个质量子特性有关，一个质量子特性至少包括一个度量特性。与表 4.4 所列的质量子特性有关的度量特性列于表 4.5 中，度量特性的描述如表 4.6 所示。

表 4.5 与质量子特性相关联的度量特性

质量子特性	度量特性
可重用性	通用性、自描述模块独立性、机器独立性、软件系统独立性
依从性	可审计性、文档完备性、自描述性
准确性	准确性

互操作性	模块独立性、通信共享性、数据共享性
安全性	存取控制、存取审查
正确性	可跟踪性、一致性、完备性
完整性	工具性、可审计性
健壮性	健壮性
容错性	容错性
可恢复性	可恢复性
时间性能	存储效率、执行效率
资源性能	资源性能
可理解性	一致性、自描述性、结构性、简明性、易读性
可学习性	可学习性
可操作性	可训练性、可通信性
可吸引力	可吸引力
可分析性	自描述性、一致性、简明性、易读性
可修改性	结构性、可扩充性、简明性、易读性
稳定性	可扩充性、通用性、自描述性、模块独立性、软件系统独立性、机器独立性
可测试性	简单性、工具性、自描述性、模块独立性、结构性、可通信性、可审计性
适应性	可扩充性、通用性、自描述性、模块独立性、软件系统独立性、机器独立性
可共存性	数据共享性、通信共享性、模块独立性
可安装性	可安装性
可替换性	可替换性

表 4.6 度量特性描述

度量特性	描述
通用性	软件部件应用的宽度
自描述性	对软件本身的功能说明的程度
模块独立性	程序中功能部件的相对独立程度，反映为内聚和耦合的程度
机器独立性	软件独立于某个特定设备的程度
软件系统独立性	软件独立于操作系统或其他环境限制的程度
可审计性	检查与标准符合性的难易程度
文档完备性	系统需求、设计、使用等说明输的完善程度
通信共享性	软件使用标准接口、协议、带宽的程度
数据共享性	软件对标准数据结构、类型的使用程度
存取控制	软件对存取信息的控制程度
存取审查	软件对存储信息的审查程度
可跟踪性	从一个设计或程序部件映射到相应需求的能力
一致性	软件开发过程中设计与文档的统一程度
完备性	软件对需求功能的实现程度
工具性	开发相应工具的容易程度
结构性	程序中代码组织的标准控制结构程度
简明性	实现软件功能所使用的最少代码的程度
易读性	软件容易看懂的程度

可训练性	软件使用户能够使用的能力
可通信性	软件可方便的为输入输出提供数据的能力
存储效率	软件的存取性能
执行效率	软件的执行性能
简单性	软件程序代码简洁的程度
可扩充性	软件结构、设计、功能可扩充的程度

我们将所有的质量特性、质量子特性和度量特性放在一起，便于进行比较，如表 4.7 所示。

表 4.7 特性对比表

质量特性	质量子特性	度量特性
功能性	可重用性	通用性、自描述模块独立性、机器独立性、软件系统独立性
	依从性	可审计性、文档完备性、自描述性
	准确性	准确性
	互操作性	模块独立性、通信共享性、数据共享性
	安全性	存取控制、存取审查
可靠性	正确性	可跟踪性、一致性、完备性
	完整性	工具性、可审计性
	健壮性	健壮性
	容错性	容错性
	可恢复性	可恢复性
效率	时间性能	存储效率、执行效率
	资源性能	资源性能
可使用性	可理解性	一致性、自描述性、结构性、简明性、易读性
	可学习性	可学习性
	可操作性	可训练性、可通信性
	可吸引性	可吸引性
可维护性	可分析性	自描述性、一致性、简明性、易读性
	可修改性	结构性、可扩充性、简明性、易读性
	稳定性	可扩充性、通用性、自描述性、模块独立性、软件系统独立性、机器独立性
	可测试性	简单性、工具性、自描述性、模块独立性、结构性、可通信性、可审计性
可移植性	适应性	可扩充性、通用性、自描述性、模块独立性、软件系统独立性、机器独立性
	可共存性	数据共享性、通信共享性、模块独立性
	可安装性	可安装性
	可替换性	可替换性

4.2.2 建立软件质量评估指标体系

为了客观地评价软件产品质量，需要一个评价指标体系，软件质量评价指标体系可提供软件实施质量评价的规范化框架，ISO/IEC9126 对软件质量评价过程模型的定义如图 3-1 所示^{[47][48][49]}。

按这个标准评价软件产品质量需要三个阶段：(1)质量需求定义；(2)评价准备阶段；(3)评价过程及结果(接受或拒绝)。

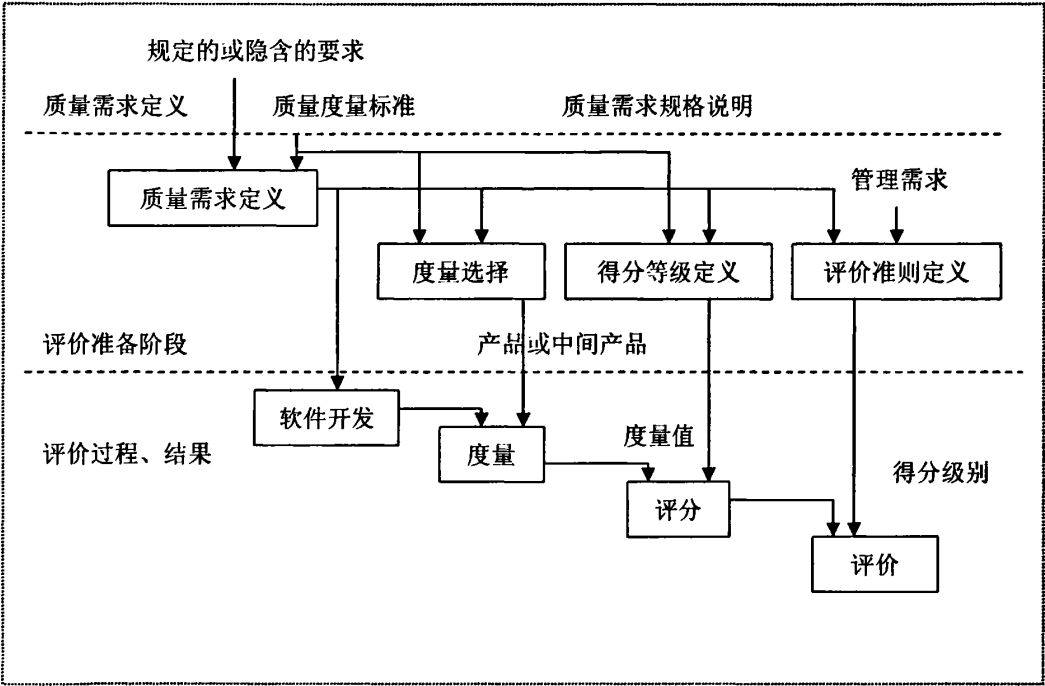


图 4.1 ISO/IEC9126 软件质量评价过程模型

1. 需求定义

软件行业存在这样一个问题，用于描述需求工作的术语没有统一的定义。对同一项的需求，不同的人会有不同的描述。Brian Lawrence 提出，需求是“任何促成设计决策的因素”。很多信息都属于这一范围。

IEEE 的软件工程标准术语表(1990)则将需求定义为：

- (1) 用户为解决某个问题或达到某个目标而须具备的条件或能力
- (2) 系统或系统组件为符合合同、标准、规范或其他正是文档而必须满足的条件或必须具备的能力
- (3) 上述第一项或第二项中定义的条件和能力的文档表达

这一定义既体现了用户对需求的看法，也代表了开发人员的观点。

质量需求反映了特定环境下对软件产品质量的要求，它是评价软件质量的前提条件。在质量需求阶段中，软件使用用户、开发人员、维护人员、测试人员和管理人员等不同类

型的人员对软件质量都有自己的看法。质量需求要求反映各类人员对质量的不同需求。对软件使用用户来说,他关心的是产品是否满足规定或隐含的需求,软件的可使用性,软件的性能以及使用软件后的成效;对开发人员来说,他们既关心软件最终产品的质量,也关心开发过程中阶段产品的质量,为了更好的开发软件产品使之满足质量要求,开发人员常常用多种度量标准来度量产品的同一质量特性。对维护人员来说,他们关心的是文档的规范性,跟源代码的一致性,源代码的可理解性及可修改性;对测试人员来说,他们关心的是软件的可测试性,软件在运行中出现错误的几率,软件的容错能力;对管理人员来说,他们关心的是软件的整体质量,而不是软件的某一个特定质量特性,他们希望在有限的财力、人力和时间内尽可能开发出好的软件。

所以对于具体软件来说,各个特性的重要性是不同的,各类人员对其需求也不一样。因此,在对软件进行评价之前,可由用户(包括上述各类人员)根据对软件质量的要求进行调整,软件质量评价的最终结果就是相对于这个质量需求的。

2. 评价准备

评价准备阶段中,要根据应用环境为软件开发的各个阶段和其最终产品分别确定适当的度量标准,建立质量特性、质量子特性、度量特性的映射关系,其中映射关系已在上节中列出。

在确定了质量特性、质量子特性、度量特性之后,下一步是设定每个质量特性的权值。有些质量特性之间存在矛盾的,要根据用户的需求调整质量特性在软件质量中所占的比例。在定义了每个质量特性的权值之后,找出其所有的子特性,然后按照这些子特性在该质量特性中的比重确定它们相应的权值。按照同样的方法给度量特性确定权值。

(1) 权值的确定

权值也叫权重,是以某种数量形式对比、权衡被评价事物总体中诸因素相对重要程度的量值。权值是一个相对的概念,是针对某一特性而言。某一特性的权值是指该特性在整体评价中的相对重要程度。权值表示在评价过程汇总,是对评价对象的各个侧面的重要程度的定量分配,对各评价特性在总体评价中的作用进行区别对待。

确定权值的方法有很多种,常用的有权值因子判断标法、专家直观判定法、层次分析法、排序法等。不同的权值分配对导致不同的评价结果,因此合理的分配权值对评价系统是非常重要的。权值应该能够真实地反映该特性在评价体系中的作用,并能体现评价者对该特性地位的理解程度^[50]。

(2) 权值确定的基本原则

① 系统优化原则。在评价软件质量体系中,每个特性对软件质量都有它的作用和贡献,对软件而言都有它的重要性。所以,在确定它们的权值时,不能只从单个特性出发,而是要处理好各个特性之间的关系,合理分配它们的权值。把整体最优化作为出发点和追求的目标。在这个原则指导下,对软件质量特性进行分析对比,权衡它们各自对整体的作用和效果,然后对它们的相对重要性做出判断。确定各自的权值,即不能平均分配,又不能片面强调某个特性,而忽略其他方面的作用。在实际工作中,应该使每个特性发挥其应

有的作用。

② 民主与集中相结合的原则。由于权值是人们对软件质量评价的重要性的认识，是定性判断的量化，因此往往受个人主观因素的影响。不同的人对同一件事情都有各自的看法，而且经常是不同的，其中有合理的成分；也有受个人价值观、能力和态度造成的偏见。这就需要集中相关人员的意见，互相补充，形成同一的方案。这样可以使考虑问题比较全面，使权值分配比较合理，防止个别人认识和处理问题的片面性。还可以比较客观的协调了评价各方之间意见不统一的矛盾，经过讨论、协商、考察各种具体情况而确定的方案，具有很强的说服力，预先消除了许多不必要的纠纷。事实上，没有重点的评价就不算是客观的评价，每个评价人员的性质和所处的层次不同，其工作的重点也肯定是不能一样的。

3. 评价过程与结果

评价过程实际上是对软件产品实施准备评价的内容，分为三步：

(1) 度量—把选定的质量度量应用到软件产品上

(2) 评级—确定某测量值的等级

(3) 评估—根据评估准则确定产品质量，并依据管理准则判断产品是否可通过验收或是否发行等等。

4.3 软件质量定量综合评估

在对度量数据分析结果的基础上，对质量的程度给予评价。对软件质量的评价有很多方法，常用的有达标法、评级法、评分法等。这些方法一般简单易行，但很难保证客观和正确性。如达标法，要求软件达到全部规定指标者为合格，否则不合格，此方法显然比较粗糙。评分法和评级法克服了达标法的部分缺点，它们运用统计学的方法确定评估指标的具体判断值，但忽略了软件质量评估指标的模糊性。影响软件质量的某些质量属性是模糊的，造成了质量评估的模糊性，由于主观原因，用户对某些影响软件质量的属性褒贬程度不尽相同，很难直接度量而获得具体数值，如何对模糊属性进行量化处理和综合评价是一个关键问题。

4.3.1 建立多层次模糊综合评估模型

模糊集合论为我们提供了对这种模糊属性进行定量描述的工具，为此本文利用模糊综合评价原理对软件质量评估有其科学性和实用价值。现对该方法的数学模型、指标权重的赋值过程等进行论述^{[51][52]}。

1. 评价步骤：

(1) 确定评价对象的因素论域

P 个评价指标， $u = \{u_1, u_2, \dots, u_p\}$ 。

(2) 确定评语等级论域

$v = \{v_1, v_2, \dots, v_p\}$, 即等级集合。每一个等级可对应一个模糊子集。

(3) 建立模糊关系矩阵 R

在构造了等级模糊子集后, 要逐个对被评事物从每个因素 $u_i (i=1, 2, \dots, p)$ 上进行量化, 即确定从单因素来看被评事物对等级模糊子集的隶属度 $(R|u_i)$, 进而得到模糊关系矩阵:

$$R = \begin{bmatrix} R|u_1 \\ R|u_2 \\ \dots \\ R|u_p \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1m} \\ r_{21} & r_{22} & \dots & r_{2m} \\ \dots & \dots & \dots & \dots \\ r_{p1} & r_{p2} & \dots & r_{pm} \end{bmatrix}_{p \times m}$$

矩阵 R 中第 i 行第 j 列元素 r_{ij} , 表示某个被评事物从因素 u_i 来看对 v_j 等级模糊子集的隶属度。一个被评事物在某个因素 u_i 方面的表现, 是通过模糊向量 $(R|u_i) = (r_{i1}, r_{i2}, \dots, r_{im})$ 来刻画的, 而在其他评价方法中多是由一个指标实际值来刻画的, 因此, 从这个角度讲模糊综合评价要求更多的信息。

(4) 确定评价因素的权向量

在模糊综合评价中, 确定评价因素的权向量: $A = (a_1, a_2, \dots, a_p)$ 。权向量 A 中的元素 a_i 本质上是因素 u_i 对模糊子集 {对被评事物重要的因素} 的隶属度。本文使用改进的层次分析法来确定评价指标间的相对重要性次序。从而确定权系数, 并且在合成之前归一化。即

$$\sum_{i=1}^p a_i = 1, \quad a_i \geq 0, \quad i=1, 2, \dots, n.$$

(5) 合成模糊综合评价结果向量

利用合适的算子将 A 与各被评事物的 R 进行合成, 得到各被评事物的模糊综合评价结果向量 B 。即:

$$A \circ R = (a_1, a_2, \dots, a_p) \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ r_{21} & r_{22} & \cdots & r_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ r_{p1} & r_{p2} & \cdots & r_{pm} \end{bmatrix} = (b_1, b_2, \dots, b_m) = B$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ a_{p1} & a_{p2} & \cdots & a_{pp} \end{bmatrix}$$

其中 b_j 是由 A 与 R 的第 j 列运算得到的, 它表示被评事物从整体上看对 v_j 等级模糊子集的隶属程度。

(6) 对模糊综合评价结果向量进行分析

实际中最常用的方法是最大隶属度原则, 但在某些情况下使用会有些很勉强, 损失信息很多, 甚至得出不合理的评价结果。提出使用加权平均求隶属等级的方法, 对于多个被评事物并可以依据其等级位置进行排序。

4.3.2 指标权重的确定

指标权重的确定我们不能一概而论, 软件类型不同, 用户的侧重点也不同。比如, 航空航天用的软件系统, 它的可靠性是一个至关重要的问题; 对于商业领域中的软件, 可维护性则是关键的问题; 在实时系统中, 效率尤为重要。因此我们应该根据软件的类型动态确定指标权重。

权重分配方案的确定方法有专家估测法、层次分析法、模糊逆方程法、环比法等, 本文采用改进的层次分析法, 从定性信息产生定量值的方法。运用改进的层次分析方法能判断基于不完全和不确定知识的可能性大小, 一旦非数值型的输入被转化为合理的标度(包括比例标度和绝对标度), 就可求出质量要素(或属性)的权重^[53]。改进的层次分析法是将层次分析法和 Delphi 法两种方法加以综合利用而得出的。

1. 德尔斐法(Delphi)确定权重

德尔斐法是专家评分法确定权重的一种, 主要形式是通过调查表或个别询问。其主要步骤如下:

(1) 针对预测内容, 选定一批熟悉相关业务的专家。

(2) 以发调查表或者个别征询意见的形式向专家发放调查表, 或者征询专家们意见。

(3) 组织者收到各专家意见后, 进行综合分析整理, 再将意见反馈给专家, 请他们重新考虑并修改自己的意见。

上述步骤可重复多次, 使意见趋于一致。

令 w_{ji} 是第 j 位专家对第 i 项指标评定的权重,则每项指标权重的平均值

$$w'_i = \frac{1}{n} \sum_{j=1}^n w_{ji}, \text{ 每位专家对各指标评定权重与平均值的离差为 } \Delta_{ji} = w_{ji} - w'_i。$$

然后进行数据的后处理,检查指标权重的分歧程度,主要通过以下指标体现:

方差:

$$D(w'_i) = \frac{1}{n-1} \sum_{j=1}^n (w_{ji} - w'_i)^2$$

标准差:

$$\sigma_{w'_i} = \sqrt{D(w'_i)}$$

分歧程度:

$$M_{w'_i} = \frac{\sigma_{w'_i}}{w'_i} \times 100\%$$

一般地,当分歧程度小于或等于 20%时,说明专家们提供的数据符合要求。

2. 层次分析法确定权重

(1) 确定目标和评价因素

P 个评价指标, $u = \{u_1, u_2, \dots, u_p\}$ 。

(2) 构造判断矩阵

在同一层次各个指标,通过两两比较其相对重要性,得到以下判断矩阵:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ a_{p1} & a_{p2} & \cdots & a_{pp} \end{bmatrix}$$

上式中,显然对于任意一个判断矩阵来说, 都有: ① $a_{ij} > 0$; ② $a_{ii} = 1$; ③ $a_{ij} = \frac{1}{a_{ji}}$;

式中 $i, j = 1, 2, \dots, p$ 。

判断矩阵中的 a_{ij} 是专家和分析者根据各指标的性质和相对关系,按表4.8给出的。

表4.8 判断矩阵标度法

标度值	对应的重要性含义
1	a 与 b 相比, a 和 b 具有同等的重要性
3	a 与 b 相比, a 比 b 比较重要
5	a 与 b 相比, a 比 b 重要
7	a 与 b 相比, a 比 b 明显重要
9	a 与 b 相比, a 比 b 极端重要
2, 4, 6, 8	介于相邻奇数判断之间的折中情况
倒数	a 对 b 的标度为 x, 则 b 对 a 的标度为 x 的倒数

(3) 计算判断矩阵

用Mathematic软件计算判断矩阵S的最大特征根 λ_{\max} ，及其对应的特征向量A，此特征向量就是各评价因素的重要性排序，也即是权系数的分配。

(4) 一致性检验

为进行判断矩阵的一致性检验，需计算一致性指标 $CI = \frac{\lambda_{\max} - n}{n - 1}$ ，平均随机一致性指标RI。它是用随机的方法构造500个样本矩阵，构造方法是随机地用标度以及它们的倒数填满样本矩阵的上三角各项，主对角线各项数值始终为1，对应转置位置项则采用上述对应位置随机数的倒数。然后对各个随机样本矩阵计算其一致性指标值，对这些CI值平均即得到平均随机一致性指标RI值。当随机一致性比率 $CR = \frac{CI}{RI} < 0.10$ 时，认为层次分析排序的结果有满意的一致性，即权系数的分配是合理的；否则，要调整判断矩阵的元素取值，重新分配权系数的值。

3. 改进的层次分析法

Delphi法优点是能充分发挥各位专家的作用，集思广益：能把各位专家意见的分歧点表达出来，取各家之长，避各家之短，集中了众多专家的意见；计算比较简单。缺点是通过打分直接给出各指标权重，主观性太强，精度不够，难以保持权重的合理性，有时会因随意性太大，导致评估指标间相对重要性得不到合理体现，因而会带来评估失衡的问题。

层次分析法最大的优点就是实现了定量与定性相结合，精度高，综合考虑评价指标体系中各层因素的重要程度而使各指标权重趋于合理，比完全凭个人经验和知识来主观确定更科学、更合理，还可以通过计算机编程将复杂的矩阵计算轻易地实现，可以提高评估效率；缺点是在构造各层因素的权重判断矩阵时，一般采用分级定量法赋值，容易造成同一系统中一个因素是另一因素的5倍、7倍，甚至9倍，从而影响权重的合理性。

本文就在结合Delphi法和层次分析法两者优点的基础上提出改进的层次分析法。判断

矩阵是层次分析法的核心，判断矩阵建立的不合理，将直接影响评价结果的得出。如果采用分级定量法赋值，过于死板。如果将判断矩阵中的指标对比重要值通过 Delphi 法来确定，集中多位专家的智慧，将使判断矩阵更加合理。

利用 Delphi 法确定判断矩阵步骤与利用 Delphi 法确定权重相似，这里就不再赘述。

4.3.3 确定评价指标的隶属度

一个软件产品，对其进行评价时，领域专家、高层管理人员(决策者)和用户一起根据所确定的评价集对其评价要素进行评价，评价结果用隶属度矩阵 R 表示如下： $R = (r_{ij})_{m \times k}$ ，在矩阵 R 中 r_{ij} 表示在第 i 个评价要素上，对它第 j 等级评定的人数占全部专家组人数的百分比，即 $r_{ij} = d_{ij} / d$ ， d_{ij} 表示第 i 个评价要素上，对它做出 V_j 评语的次数， d 表示全部专家组人数。 m 为评价要素的个数， k 为评语数。

隶属度的得出，本文采用问卷调查法，调查对象为软件爱好者，即对软件功能及使用都很熟悉的人。

4.3.4 评估结果分析

根据度量结果（得分和等级）软件能达到全部规定指标者为合格，否则不予通过。度量值本身常常并不能绝对地说明什么，需要充分考虑被度量项目的特性和内容，根据度量需求来分析实际度量值，对于当今的软件而言，并不存在关于度量值的通用的标准尺度，需要在类似度量对象之间比较，从而给出相对意义下的关于度量结果的解释。度量值的判断可以通过计划值和期望值作比较，实际上软件度量值同期望值的偏离可能由许多不同的原因引起，有些可能是坏的，而其中有些偏离却是好的，不能一概而论，而应该正确区分和认识造成偏离的根本原因，从而正确地评价软件质量。

4.4 实例分析

本文以北京某广告公司的一款广告信息管理软件为例，对其进行质量度量。

4.4.1 度量指标的选择

根据前章的度量模型，本文选定 6 个质量特性以及 19 个子特性构成评价体系。如下表 4.9 所示。

表 4.9 北京某广告公司广告信息管理软件两级评价指标

质量特性	质量子特性	描述
功能性(0.202)	可重用性(0.183)	程序或程序的一部分在其他程序中应用的程度
	依从性(0.128)	遵循有关规定的程度
	准确性(0.210)	软件在计算和控制时能否得到正确结果的能力
	互操作性(0.174)	连接一个系统与另一个系统的难易程度
	安全性(0.199)	保护和控制软件和数据的能力
	可恢复性(0.106)	软件在失效发生后能否恢复到失效前的数据的能力
可靠性(0.156)	正确性(0.213)	软件满足需求及用户目标的程度
	完整性(0.321)	对未授权人员访问软件或数据行为的控制程度
	健壮性(0.285)	软件中的错误能随着时间自行消除的能力
	容错性(0.181)	软件遇到错误时仍能维持性能的能力
效率(0.165)	时间性能(1)	软件的处理时间和响应时间
可 使 用 性 (0.202)	可理解性(0.217)	用户理解怎样使用软件的难易程度
	可学习性(0.285)	用户学习使用该软件的难易程度
	可操作性(0.246)	用户操作软件的难易程度
	可吸引力(0.252)	界面的友善程度
可 维 护 性 (0.109)	可分析性(0.474)	诊断缺陷或失效原因的难易程度
	可修改性(0.526)	对软件中的错误进行修改的难易程度
可 移 植 性 (0.166)	适应性(0.429)	软件在改变环境后说具有的适应能力
	可共存性(0.571)	软件安装后与其他软件的共存能力

注：表中括号内的数值为权值，需要通过计算才能得出。

4.4.2 评价集的建立

利用语义学标度分为4个测量等级：好、良好、一般、差。为了便于计算，我们将主观评价的语义学标度进行量化，并依次赋值为4、3、2及1。主观测量是用四级语义学标度。

所设计的评价定量标准见表4.10。

表 4.10 评价定量分级标准

评价值	评语	定级
$x_i > 3.5$	好	E1
$2.5 < x_i \leq 3.5$	良好	E2
$1.5 < x_i \leq 2.5$	一般	E3
$x_i \leq 1.5$	差	E4

4.4.3 指标权重的求解

(1) 确定评价对象集

P = 北京某广告公司广告信息管理软件质量。

(2) 构造评价因子集

$$u = \{u_1, u_2, \dots, u_6\} = (\text{功能性, 可靠性, 效率, 可使用性, 可维护性, 可移植性})$$

(3) 确定评语等级论域

确定评语等级论域，即建立评价集 v 。

$$v = \{v_1, v_2, \dots, v_4\} = \{\text{好, 良好, 一般, 差}\}$$

(4) 一级指标权重的计算

6个一级指标因子权重，我们采用改进的层次分析的方法求出指标权重。构造判断矩阵 $S = (u_{ij})_{p \times p}$ 即：

$$S = \begin{bmatrix} 1 & \frac{4}{3} & \frac{5}{4} & 1 & \frac{9}{5} & \frac{6}{5} \\ \frac{3}{4} & 1 & \frac{9}{10} & \frac{8}{9} & \frac{7}{5} & \frac{8}{9} \\ \frac{4}{5} & \frac{10}{9} & 1 & \frac{4}{5} & \frac{3}{2} & 1 \\ 1 & \frac{9}{8} & \frac{5}{4} & 1 & 2 & \frac{5}{4} \\ \frac{5}{9} & \frac{5}{7} & \frac{2}{3} & \frac{1}{2} & 1 & \frac{4}{6} \\ \frac{5}{6} & \frac{9}{8} & 1 & \frac{4}{5} & \frac{6}{4} & 1 \end{bmatrix}$$

用Mathematic软件计算判断矩阵 S 的最大特征根得 $\lambda_{\max} = 6.00589$ 。为进行判断矩阵的一致性检验，需计算一致性指标：

$$CI = \frac{\lambda_{\max} - n}{n - 1} = \frac{6.00589 - 6}{6 - 1} = 0.001178$$

平均随机一致性指标 $RI = 1.24$ 。随机一致性比率：

$$CR = \frac{CI}{RI} = \frac{0.001178}{1.24} = 0.00095 < 0.10$$

因此认为层次分析排序的结果有满意的一致性，即权系数的分配是非常合理的。

其对应的特征向量为：

$$A_0 = (1.21372, 0.935715, 0.9911, 1.21138, 0.634379, 1.0)$$

再作归一化处理得：

$$A = (0.202, 0.156, 0.165, 0.202, 0.109, 0.166)$$

(5) 计算二级指标权重

同理，我们仍采用改进的层次分析的方法来求出指标权重。分别对各个二级指标构造其各自的判断矩阵，再用Mathematic软件计算最大特征根和一致性检验。得出合理的权系数。

功能性六个指标的权重，其特征向量为

$$(1.72669, 1.20273, 1.97386, 1.64237, 1.87383, 1.0)$$

归一化得：(0.183, 0.128, 0.210, 0.174, 0.199, 0.106)

可靠性指标的权重：(0.213, 0.321, 0.285, 0.181)

可使用性指标的权重：(0.217, 0.285, 0.246, 0.252)

可维护性评价指标的权重：(0.474, 0.526)

可移植性指标的权重：(0.429, 0.571)

4.4.4 某广告公司信息管理软件质量的综合评价

(1) 软件质量的加权平均模糊合成综合评价

利用加权平均 $M(\bullet, \oplus)$ 模糊合成算子将 A 与 R 合成得到模糊综合评价结果向量 B 。模糊综合评价中常用的取大取小算法，在因素较多时，每一因素所分得的权重常常很小。在模糊合成运算中，信息丢失很多，常导致结果不易分辨和不合理(即模型失效)的情况。所以，针对上述问题，这里采用加权平均型的模糊合成算子。计算公式为：

$$b_j = \sum_{i=1}^p (a_i \cdot r_{ij}) = \min \left(1, \sum_{i=1}^p a_i \cdot r_{ij} \right), j=1, 2, \dots, m$$

式中， b_j ， a_i ， r_{ij} 分别为隶属于第 j 等级的隶属度、第 i 个评价指标的权重和第 i 个评价指标隶属于第 j 等级的隶属度。

(2) 多级模糊综合评价结果向量

将来源于抽样调查的统计数据代入建立的模型中，计算各级模糊综合评价的向量。

① 功能性指标的评价向量

$$A_1 = a \circ R$$

$$= (0.183, 0.128, 0.210, 0.174, 0.199, 0.106) \circ \begin{pmatrix} 0.154 & 0.404 & 0.410 & 0.032 \\ 0.006 & 0.272 & 0.500 & 0.223 \\ 0.053 & 0.756 & 0.191 & 0.000 \\ 0.107 & 0.368 & 0.354 & 0.170 \\ 0.373 & 0.408 & 0.189 & 0.030 \\ 0.164 & 0.436 & 0.313 & 0.087 \end{pmatrix}$$

$$= (0.150309, 0.458948, 0.311525, 0.079172)$$

归一化后的综合评价向量：(0.150, 0.459, 0.312, 0.079)

② 可靠性指标的评价向量

$$B_1 = (0.213, 0.321, 0.285, 0.181) \circ \begin{pmatrix} 0.058 & 0.277 & 0.556 & 0.110 \\ 0.160 & 0.489 & 0.310 & 0.041 \\ 0.040 & 0.328 & 0.466 & 0.167 \\ 0.041 & 0.225 & 0.499 & 0.236 \end{pmatrix}$$

$$= (0.082535, 0.225865, 0.498879, 0.197111)$$

归一化得(0.084,0.226,0.499,0.197)

③ 效率指标的评价向量

$$C_1 = (0.035, 0.370, 0.511, 0.084)$$

④ 可使用性指标的评价向量

$$D_1 = (0.217, 0.285, 0.246, 0.252) \circ \begin{pmatrix} 0.049 & 0.417 & 0.441 & 0.093 \\ 0.024 & 0.224 & 0.483 & 0.270 \\ 0.034 & 0.203 & 0.532 & 0.231 \\ 0.025 & 0.296 & 0.543 & 0.137 \end{pmatrix}$$

$$= (0.032137, 0.278859, 0.50106, 0.188481)$$

归一化得(0.032,0.279,0.501,0.188)

⑤ 可维护性指标的评价向量

$$E_1 = (0.474, 0.526) \circ \begin{pmatrix} 0.022 & 0.277 & 0.493 & 0.208 \\ 0.031 & 0.320 & 0.499 & 0.150 \end{pmatrix}$$

$$= (0.026734, 0.299618, 0.496156, 0.177492)$$

归一化得(0.027,0.300,0.496,0.177)

⑥ 可移植性指标的评价向量

$$F_1 = (0.429, 0.571) \circ \begin{pmatrix} 0.036 & 0.349 & 0.524 & 0.091 \\ 0.008 & 0.143 & 0.406 & 0.444 \end{pmatrix}$$

$$= (0.020012, 0.231374, 0.456622, 0.292563)$$

归一化得 (0.020,0.231,0.457,0.292)

⑦ 综合评价向量

$$A = (0.202, 0.156, 0.165, 0.202, 0.109, 0.166) \circ \begin{pmatrix} 0.150 & 0.459 & 0.312 & 0.079 \\ 0.048 & 0.226 & 0.499 & 0.197 \\ 0.035 & 0.370 & 0.511 & 0.084 \\ 0.032 & 0.279 & 0.501 & 0.188 \\ 0.027 & 0.300 & 0.496 & 0.177 \\ 0.020 & 0.231 & 0.457 & 0.292 \end{pmatrix}$$

$$= (0.05629, 0.316428, 0.456311, 0.166291)$$

归一化得 $A' = (0.057, 0.318, 0.458, 0.167)$

⑧ 对综合评分值进行等级评定

$$V_A = 4 \times 0.150 + 3 \times 0.459 + 2 \times 0.312 + 1 \times 0.079 = 2.68$$

$$V_B = 4 \times 0.084 + 3 \times 0.226 + 2 \times 0.499 + 1 \times 0.197 = 2.09$$

$$V_C = 4 \times 0.035 + 3 \times 0.370 + 2 \times 0.511 + 1 \times 0.084 = 2.356$$

$$V_D = 4 \times 0.032 + 3 \times 0.279 + 2 \times 0.501 + 1 \times 0.188 = 2.155$$

$$V_E = 4 \times 0.027 + 3 \times 0.300 + 2 \times 0.496 + 1 \times 0.177 = 2.177$$

$$V_F = 4 \times 0.020 + 3 \times 0.231 + 2 \times 0.457 + 1 \times 0.292 = 1.979$$

由上述计算可知,对照表 4.10 的评价分级标准可得北京某广告公司广告信息管理软件
的“功能性”评价指标的评价结果为“良好”属于 E2 级,其它 5 个指标的评价结果都均
为“一般”,属于 E3 级。按照各个指标的评分等级的大小可以对其排序,其中“可靠性”、
“可移植性”的评价要比其它指标都要低一点。而对总体的综合评判分值为:

$$V = 4 \times 0.057 + 3 \times 0.318 + 2 \times 0.458 + 1 \times 0.167 = 2.65$$

说明北京某广告公司广告信息管理软件总体质量为“良好”,属于 E2 级。

实际中最常用的方法是最大隶属度原则,但此方法的使用是有条件的,存在有效性问

题，可能会得出不合理的评价结果。根据此问题提出加权平均原则求隶属等级的方法，对于采用加权平均原则对上述各级评价指标的评价结果进行分析。

4.5 本章小结

针对前一章提出的新的软件质量度量模型进行详细的分析，通过实例介绍了新的度量模型的运用。

第五章 总结及展望

5.1 结论

本文的研究工作主要是在 ISO/IEC9126 标准的基础上提出了一种新的软件质量度量模型，确定了各项指标的权重，完成了软件质量的评价。主要工作总结如下：

- (1) 阅读有关软件质量度量的国内外研究资料，着重分析和归纳了软件质量度量模型。
- (2) 通过对比国内外的研究模型，提出了一种新的软件质量度量模型。该模型为二层结构，可对软件质量进行定量化评估，是对 ISO/IEC9126 模型的扩充和细化。
- (3) 针对提出的度量模型，确立了度量框架以及度量指标的具体内容，分析了不同层次度量指标之间的关联关系，对度量的过程进行了公式化阐述。
- (4) 以度量模型为原型，按照软件质量度量框架，运用多层次模糊综合评估方法，对已知软件的质量进行了度量。

5.2 待解决问题

新的度量模型虽已能够对软件质量进行度量，但还存在一些有待于进一步研究和讨论的工作：

- (1) 如果度量指标的权重经过计算，发现不合理，如何进行合理重新分配。
- (2) 给出的例子是进行的二层指标的度量，如果度量指标层数为 N 层，能否进行度量。

参考文献

- [1] 郑人杰. 实用软件工程[M]. 北京: 清华大学出版社, 1996 年 2 月, 第二版
- [2] 罗伯特格拉斯著, 陈河南等译. 软件开发的滑铁卢[M]. 北京: 电子工业出版社, 2002
- [3] 蔡开元. 软件可靠性工程基础[M]. 北京: 清华大学出版社, 1995 年 9 月, 第一版
- [4] 秦娥, 魏红娟. 加强软件质量管理的必要性[J]. 平原大学学报, 2003
- [5] 李健, 金茂忠. 中小型企业软件过程改善方法研究[J]. 计算机工程与应用 2001(19): 107~112
- [6] 罗圣仪. 计算机软件质量保证的方法和实践[M]. 北京: 北京科学出版社, 1999
- [7] 李健, 金茂忠. 有效改善软件过程方法研究[M]. 北京: 人民邮电出版, 1999
- [8] 贾宗宣, 龚晓峰. 基于 CVS 的软件项目管理的研究[J]. 微计算机信息, 2006, 22(2-3): 222~224
- [9] 李怀璋, 王青. 基于 ISO9000 和 CMM 的软件质量管理系统的研究[J]. 中国科学院研究所, 2001, 25~27
- [10] Gillies A C. Software Quality: Theory and Management. (Second edition). [M]. Thomson Computer Press, 1997
- [11] ChidungLac, Jean-LUC Raffy. A tool for software quality[J]. IEEE Trans on Knowledge and Data Engineering, 1992, 4(2): 115~130
- [12] Grady, R and Caswell, D. Software Metrics: Establishing a Company-Wide Program. Englewood Cliffs, NJ, Prentice-Hall, 1987
- [13] 李华荣. 软件企业质量管理模型的研究[J]. 山西科技, 2002(4): 27~28
- [14] 刘孟仁等译, 卡耐基梅隆大学软件工程研究所编著. 能力成熟度模型(CMM): 软件过程改进指南[M]. 北京: 电子工业出版社, 2004
- [15] 方德英, 李敏强. 信息系统项目监理机制的经济学分析[J]. 管理工程学报, 2003(4): 98-102
- [16] 黄锡滋. 软件可靠性、安全性与质量保证[M]. 北京: 电子工业出版社, 2002
- [17] 邓良松, 刘海岩, 陆丽娜. 软件工程[M]. 西安: 西安电子科技大学出版社, 2000
- [18] 单银根. 软件能力成熟度模型(CMM)与软件开发技术[M]. 北京: 北京航空航天大学出版社, 2003
- [19] 马闰娟, 梁成才. CMM二级KPA软件质量保证的一个实施方案[J]. 计算机工程, 2003, 29(2)
- [20] Steven R Rakitin 著, 于秀山, 包晓露, 焦跃译. 软件验证与确认的最佳管理方法[M]. 北京: 电子工业出版社, 2002
- [21] Wiegers, KE 著, 刘伟琴, 刘洪涛译. 软件需求(第 2 版) [M]. 清华大学出版社, 2004
- [22] 张根耀, 惠小静. 软件质量保障技术的新发展[J]. 延安大学学报(自然科学版), 2002,

21(1): 23~25

- [23] 刘夏. 软件质量度量与质量保证的研究[J]. 聊城大学学报: 自然科学版, 2004, 17(2): 97~99, 101
- [24] 陈健明, 王海峰. 软件质量模型及其评价[J]. 微电子学与计算机, 2003(5): 64~68
- [25] 程杜平, 钱红兵. 软件质量与度量[J]. 计算机工程与设计, 2002(7): 80~83
- [26] 林俊兴. 软件质量量化指标及实现方法的探讨[J]. 计算机工程与应用, 2005(45): 64~68
- [27] Brooks FP. The mythical man-month[M]. Reading(MA): Addison-Wesley, 1975
- [28] 石柱, 何新贵. 软件质量的模糊属性及其表示[J]. 计算机工程与设计, 2001, 22(1-4)
- [29] M.Hitz, B.Montazeri.Chidamber and Kemerer's Metrics Suite: A Measurement Theory Perspective, IEEE Transactions on Software Engineering, Vol.22, No.4, April 1996
- [30] Standards Coordinating Committee of the IEEE Computer Society, IEEE Standard Glossary of Software Engineering Terminology, IEEE-STD-610, 12-1990, IEEE, New York, 1991
- [31] 赵一鸣. 基于 ISO-9126 质量模型的软件质量评价方法[J]. 计算机工程与应用, 2002, 93~94
- [32] 彭芸. 软件质量控制和定量评估的研究[D]. 泰安: 山东科技大学学位论文, 2004
- [33] 李心科. 软件故障分析及质量评估方法的研究[D]. 合肥: 合肥工业大学博士论文, 2001
- [34] ISO/IEC JTC1/SC7/WG6, ISO/IEC9126-1. Information Technology Software Quality Characteristics and Metrics-Part 1: Quality Model[S]
- [35] Sutcliffe A, Maiden N. The domain theory for requirements engineering [J]. IEEE Transactions on Software Engineering, 1998, 24(3):760~773
- [36] Weiner, LR. Digital Woes: Why We Should Not Depend on Software[J]. Reading, MA: Addison-Wesley, 1993, 4~15
- [37] 赵雪峰. 通用应用软件质量评价指标体系研究[D]. 杭州: 浙江大学硕士学位论文, 2006
- [38] Leveson, NG, and CS Turner. An Investigation of the Therac-25 Accidents[J]. IEEE Computer, April 1987, 10~19
- [39] Information Technology: Software Product Evaluation: Quality Characteristics and Guidelines for Their Use, ISO/IEC9126, 1991
- [40] Xavier Franch and Juan Pablo Carvallo. Universitat Politecnica de Catalunya, Using Quality Models in Software Package Selection. IEEE SOFTWARE, IEEE Computer Soc. Press, Los Alamitos, Calif. January/February 2003: 34~41
- [41] 叶言苓, 朱三元. 软件质量评价体系及其实现[J]. 计算机应用与软件, 2001(1): 26~33
- [42] 王超, 梁义芝. 软件质量的一种定量评价方法[J]. 情报指挥控制系统与仿真技术, 2002 (10): 54~57
- [43] 陈建明, 王海峰. 软件质量模型及其评价[J]. 微电子学与计算机, 2003 (5): 64~67

- [44] 肖鸿民. 软件质量评价的一种定量化方法[J]. 西北师范大学学报, 2000 (4): 31~35
- [45] 李宗军, 滕延燕. 软件质量模糊评价模型及其算法[J]. 青岛大学学报, 2004 (2): 86~90
- [46] 陈颖洁. 军用软件质量评价工具的开发与设计[D]. 上海: 华东师范大学学位论文, 2006
- [47] Deutsch M, Willis R. Software Quality Engineering [M]. Englewood Cliffs, NJ Prentice-Hall, 1988
- [48] R.J.Ruber, R.D Hartwick. Quantitative Measurement of Program Quality [M]. Proceedings of ACM National Conference, 1968
- [49] Wong SKM, Pawan Lingras. Representation of qualitative user preference by quantitative belief functions [J]. IEEE Trans. on Knowledge and Data Engineering, 1994, 6(1):72~78
- [50] 袁瑞跃, 董明. 一种医院管理系统软件综合评审的方法[J]. 中国医疗器械杂志, 1999(1)
- [51] 石柱. 基于模糊技术的软件质量评价及可靠性评估[D]. 北京: 北京航空航天大学学位论文, 2000
- [52] 王胜之, 鲜明, 蒋兴才. 基于判断矩阵的软件评价指标权重分配方法[J]. 计算机应用与软件, 2003 (5): 40~42
- [53] 杨宗奎. 基于综合评价方法的软件质量评价系统研究[D]. 西安: 西安电子科技大学学位论文, 2004

攻读硕士学位期间发表的科研论文

完成学术论文

- [1] 汪勇, 张刚, 张百栈. 改进的进化规划及在采购方案优化中的应用. 系统工程理论与实践, 2008

致 谢

时光的飞快流逝，让我越来越深切的感受到岁月如梭。回首学习生涯，有荆棘，也有坦途；有欢笑，也有汗水。但更让我感到无法忘怀并永远心存感激的，是一路上辛勤培育我们的老师。我们的学习生涯得到了众多老师的关怀和帮助，这里我要由衷的向勉励和支持我的人表示诚挚的谢意。

首先要感谢尊敬的导师汪勇老师。汪老师的深厚学识和严谨的治学态度无时无刻不在影响着我，激励着我。学习上，汪老师给予我悉心指导，使我的学术水平有所提高；学业上，汪老师对我的论文提出了宝贵建议，不厌其烦的与我讨论，使这篇论文日渐完善。

感谢学校的老师和同学，是他们的关心和帮助，让我顺利完成了研究生的学习生涯。

感谢我的父母和亲人，他们是我最大的精神支柱。

感谢所有关心、爱护我的朋友们。

张刚

武汉科技大学管理学院