

软件质量工程

—— 度量与模型 (第二版)

Metrics and Models in Software Quality Engineering
Second Edition

METRICS AND MODELS
IN SOFTWARE QUALITY
ENGINEERING

SECOND EDITION



STEPHEN H. KAN

[美] Stephen H. Kan 著

吴明晖 应 晶 等译

何志均 审校

国外计算机科学教材系列

TP311.5
K035

软件质量工程

——度量与模型

(第二版)

Metrics and Models in Software Quality Engineering

Second Edition

[美] Stephen H. Kan 著

吴明晖 应 晶 等译

何志均 审校

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是一本关于软件开发本质主题的权威著作。作者通过大量详尽的工业实例,深入浅出地介绍了怎样测量软件质量,以及利用测量来实现软件开发过程的改进。书中通过四个主要的质量度量和模型类别展开论述:质量管理、软件可靠性和预测、复杂性以及客户观点。另外,本书还讨论了测量理论的基础,特殊的质量度量和工具,以及在软件开发过程中应用度量的方法。全书内容丰富,层次分明,并在前一版的基础上进行了大量的更新,补充了许多相关的理论与应用。

本书可作为计算机及相关专业本科生与研究生的教材或参考书,也是每一位从事软件质量工程研究的人员的宝贵资料。

Simplified Chinese edition Copyright © 2004 by PEARSON EDUCATION ASIA LIMITED and Publishing House of Electronics Industry.

Metrics and Models in Software Quality Engineering, Second Edition, ISBN: 0201729156 by Stephen H. Kan. Copyright © 2003. All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Pearson Education, Inc. This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macau).

本书中文简体字翻译版由电子工业出版社和Pearson Education培生教育出版亚洲有限公司合作出版。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有Pearson Education培生教育出版集团激光防伪标签,无标签者不得销售。

版权贸易合同登记号 图字:01-2003-1049

图书在版编目(CIP)数据

软件质量工程——度量与模型(第二版)/〔美〕卡恩(Kan S. H.)著;吴明晖等译. -北京:电子工业出版社,2004.7

(国外计算机科学教材系列)

书名原文:Metrics and Models in Software Quality Engineering, Second Edition
ISBN 7-5053-9912-8

I. 软... II. ①卡... ②吴... III. 软件质量-质量管理-教材 IV. TP311.5

中国版本图书馆CIP数据核字(2004)第062222号

责任编辑:冯小贝

印 刷:北京兴华印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:21.5 字数:605千字

印 次:2004年7月第1次印刷

定 价:35.00元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换;若书店售缺,请与本社发行部联系。联系电话:(010)68279077。质量投诉请发邮件至zltz@phei.com.cn,盗版侵权举报请发邮件至dbqq@phei.com.cn。

出版说明

21 世纪初的 5 至 10 年是我国国民经济和社会发展的关键时期,也是信息产业快速发展的关键时期。在我国加入 WTO 后的今天,培养一支适应国际化竞争的一流 IT 人才队伍是我国高等教育的重要任务之一。信息科学和技术方面人才的优劣与多寡,是我国面对国际竞争时成败的关键因素。

当前,正值我国高等教育特别是信息科学领域的教育调整、变革的重大时期,为使我国教育体制与国际化接轨,有条件的高等院校正在为某些信息学科和技术课程使用国外优秀教材和优秀原版教材,以使我国在计算机教学上尽快赶上国际先进水平。

电子工业出版社秉承多年来引进国外优秀图书的经验,翻译出版了“国外计算机科学教材系列”丛书,这套教材覆盖学科范围广、领域宽、层次多,既有本科专业课程教材,也有研究生课程教材,以适应不同院系、不同专业、不同层次的师生对教材的需求,广大师生可自由选择和自由组合使用。这些教材涉及的学科方向包括网络与通信、操作系统、计算机组织与结构、算法与数据结构、数据库与信息处理、编程语言、图形图像与多媒体、软件工程等。同时,我们也适当引进了一些优秀英文原版教材,本着翻译版本和英文原版并重的原则,对重点图书既提供英文原版又提供相应的翻译版本。

在图书选题上,我们大都选择国外著名出版公司出版的高校教材,如 Pearson Education 培生教育出版集团、麦格劳-希尔教育出版集团、麻省理工学院出版社、剑桥大学出版社等。撰写教材的许多作者都是蜚声世界的教授、学者,如道格拉斯·科默(Douglas E. Comer)、威廉·斯托林斯(William Stallings)、哈维·戴特尔(Harvey M. Deitel)、尤利斯·布莱克(Uyless Black)等。

为确保教材的选题质量和翻译质量,我们约请了清华大学、北京大学、北京航空航天大学、复旦大学、上海交通大学、南京大学、浙江大学、哈尔滨工业大学、华中科技大学、西安交通大学、国防科学技术大学、解放军理工大学等著名高校的教授和骨干教师参与了本系列教材的选题、翻译和审校工作。他们中既有讲授同类教材的骨干教师、博士,也有积累了几十年教学经验的老教授和博士生导师。

在该系列教材的选题、翻译和编辑加工过程中,为提高教材质量,我们做了大量细致的工作,包括对所选教材进行全面论证;选择编辑时力求达到专业对口;对排版、印制质量进行严格把关。对于英文教材中出现的错误,我们通过与作者联络和网上下载勘误表等方式,逐一进行了修订。

此外,我们还将与国外著名出版公司合作,提供一些教材的教学支持资料,希望能为授课老师提供帮助。今后,我们将继续加强与各高校教师的密切联系,为广大师生引进更多的国外优秀教材和参考书,为我国计算机科学教学体系与国际教学体系的接轨做出努力。

电子工业出版社

教材出版委员会

主任 杨芙清 北京大学教授
中国科学院院士
北京大学信息与工程学部主任
北京大学软件工程研究所所长

委员 王 珊 中国人民大学信息学院院长、教授
胡道元 清华大学计算机科学与技术系教授
国际信息处理联合会通信系统中国代表
钟玉琢 清华大学计算机科学与技术系教授
中国计算机学会多媒体专业委员会主任

谢希仁 中国人民解放军理工大学教授
全军网络技术研究中心主任、博士生导师

尤晋元 上海交通大学计算机科学与工程系教授
上海分布计算技术中心主任

施伯乐 上海国际数据库研究中心主任、复旦大学教授
中国计算机学会常务理事、上海市计算机学会理事长

邹 鹏 国防科学技术大学计算机学院教授、博士生导师
教育部计算机基础课程教学指导委员会副主任委员

张昆藏 青岛大学信息工程学院教授

译者序

软件产业是信息产业的发展重点。近几年来,我国的软件产业保持着较快的增长,软件与信息服务市场的份额持续扩大,但也存在着企业规模、核心技术和软件出口能力等方面的问题,尤其是软件企业的生产和流程管理水平存在差距,经常出现开发费用超支、开发周期拖延、项目管理混乱等问题。这使得软件质量难以得到保证。随着软件开发过程与活动的日益复杂,软件系统的规模和复杂程度亦快速增长。如何保证软件系统的质量并有效地降低软件开发的風險,这将是软件工程领域的研究焦点与迫切需求。

这是一本关于软件开发本质主题的权威著作。本书通过大量详尽的工业实例,深入浅出地说明了怎样测量软件质量,以及利用测量来实现软件开发过程的改进。书中通过四个类别的度量和模型展开讨论:(1)质量管理模型;(2)软件可靠性和预测模型;(3)复杂性度量和模型;(4)客户观点度量、测量和模型。这些度量和模型覆盖了整个软件开发过程:从高层设计到测试和维护,以及关于可靠性的所有阶段。另外,本书还讨论了测量理论的基础、特殊的质量度量和工具,以及在软件开发过程中应用度量的方法。本书在第一版的基础上进行了大量的更新工作,并结合软件质量工程领域的最新发展和需求,新增了软件测试的过程中度量、面向对象度量、可用性度量、过程度量评价、软件项目评价、过程改进的注意事项以及测量软件过程改进等一些内容。通过理论、技术和实例三者的完美结合,作者系统地阐述了软件开发中最重要的主题之一:质量工程。这本书无论是对于学术界的科研工作,还是对于软件开发组织的计划以及管理实际项目的度量、分析和改进质量都具有很高的参考价值。

本书的第1章~第10章以及第17章~第19章由吴明晖翻译,第11章~第16章由应晶教授翻译。全书由最后吴明晖负责统稿,并由何志均教授进行了审校。在本书的翻译过程中,美国Michigan State University 计算机系的吴明博士以及浙江大学计算机系的邬惠峰、余春艳、陈羨、李钢、方敏、柯海丰、栾雪剑、黄晟盛、胡君、曹立、付超、张琦也参与了大量的工作,并提出了很多的意见和建议。由于译者水平有限,错误之处在所难免,敬请各位读者不吝赐教。

第二版序

五十多年以来,软件一直都是一个充满问题的学科。软件的问题非常多,包括被取消、引起诉讼、费用超支、进度延迟、高昂的维护费用,以及较低的客户满意度水平等。有关软件的问题经常会频繁出现。我公司的研究表明,大约一半以上的大型软件项目会遇到某种形式的延迟、超支或运作失败。

但软件并非一定要有那么多问题。实际上一些复杂的软件项目能够赶上发布进度,符合代价目标,并且使用时也很正常。在我的职业生涯中,一直致力于研究成功的软件项目和失败项目之间的差异。

当比较类似的项目时,很容易确定带来软件成功的主要因素,特别是当一部分成功而另一部分失败的时候。成功的软件项目在软件质量控制方面取得了很大的成绩,这取决于它们良好的软件质量测量。

尽管好的软件质量控制代价昂贵,但它却在投资方面产生了很好的回报。当我们研究失败或取消的软件项目时可以发现,它们有一些相似的模式:早期阶段处理不够谨慎,没有足够的需求分析或设计审查。在匆忙度过早期阶段之后,看上去进度是提前了,但是许多问题却发生在编码和测试时期。当测试开始时,检测出了重大的问题以至于进度和代价目标不能达到。实际上,一些软件项目有太多的严重问题——称为缺陷,以至于没有完成就被取消。

而成功的项目在开始阶段会更加谨慎仔细。相关人员将仔细地分析需求,并且正式地审查设计。虽然这些工作需要花费一些时间,并且增加了一些开销,不过一旦开始编码和测试阶段,良好的质量控制就会使项目更加迅速地朝着成功的方向迈进。

Stephen Kan 和我都了解软件质量控制以及 IBM 好的软件质量度的重要性。尽管 Stephen 和我几十年来在 IBM 的不同实验室工作,但我们还是拥有共同的观点。经验数据使我们了解,没有出色的软件质量控制,大的系统开发是非常危险且容易失败的。我们还知道,有效的软件质量控制建立在使用精确度量进行仔细测量的基础上。

Stephen 和我都能访问 IBM 的内部质量数据——这是世界上最好和最大的软件数据库之一。但这是一个公司的内部信息并不足以令人信服,除非其他组织也提供了这些类似信息。Stephen 的著作得到了广泛的认可,这本书提供了来自于其他许多公司和政府机构的宝贵信息,其中引用了惠普、摩托罗拉、NASA 和其他一些成功解决了软件问题的组织的研究成果。

我在 1995 年第一次读到本书的第一版。我发现这部著作包含了详尽的研究成果,覆盖了许多与软件质量工程相关的问题,并且非常清楚地阐述了它们之间的关系。

这个新的版本保留了早期版本的精华,而且加入了许多新的信息,例如关于面向对象方法的质量测量的章节。Stephen Kan 的新版本很好地考虑了相关内容的深度和广度的权衡。这是我在软件质量工程和度量方面见过的最好的一本书。

Capers Jones

Chief Scientist Emeritus

Software Productivity Research, Inc.

第一版序

质量管理和工程在过去几年中有着大量的应用。例如：

- 一个教学医院的系统，保守预算为 1780 万美元，通过五年的质量管理，节约了 250 万美元的投资。
- 美国空军空运指挥系统 (U.S. Air Force Military Airlift Command) 通过详尽的质量改进，使得在海湾战争中的空运性能有了很大的提高，因此避免了部署民用航班 (因此也避免了延迟第二天的邮件运送等)。
- 美国劳工统计局 (U.S. Bureau of Labor Statistics) 减少了产生月消费价格指数 (CPI) 的时间，其中涉及到五个部门的 650 人，共减少了 33% 的时间，而且保持了原来的精确度。
- 宾州大学通过一个降低邮寄费用的项目，每年节省了 6 万多美元的开销。

这样的例子还有很多，遍布各行各业，如电子通信、健康保险、法律、医院、政府、医药、铁路和学校。TQM 成功的领域多得几乎让人迷惑。

虽然整个世界以惊人的速度正朝着质量改进的方向迈进，但是软件开发人员和工程师往往落后于潮流，他们还沉浸在度量和过程模型、方法论和 CASE 工具方面的辩论之中。部分原因包括在定义用户需求方面的特殊困难，许多软件企业中普遍存在的“领导”意识，以及软件工程应用的相对不成熟性。尽管前两个原因本身也是有趣的话题，但 Stephen Kan 的著作主要关注的是第三个原因。

请设想一下只利用很少的一部分航空工程知识来设计飞机，忽略机械工程知识而去设计汽车，没有任何化学工程知识来运作加工精炼厂。这样，对于通常来自于提供软件质量解决方案的顾问处的建议是首先应用已经证实的软件工程方法，我们就不会感到奇怪了。正是由于这些方法在软件开发领域中接受得很慢，因此也不能很快地应用质量工程的方法。

这种接受缓慢的现象的一个主要原因是，尽管文献描述了在主要大型软件开发企业中的实际应用，但却没有给出清晰的基础知识描述。Stephen Kan 编写的这本著作，是朝着这个需求迈进的一大步。

Kan 博士为软件质量工程领域提供了一个很全面的参考，他很好地权衡了需要的技术细节以及模型与技术的实际应用。这本书提供了很多工业实例，这些例子不仅仅来自于 MBNQA 的得主 IBM Rochester，还包括了 NEC、惠普、摩托罗拉、NASA 以及 IBM Federal Systems Division 的内部实例。利用软件工业实例来解释概念和理论，使得本书的内容非常丰富。

Joseph Juran 博士，现代质量管理和工程的主要奠基者之一，描述了“质量堤坝后的生活” (life behind the quality dikes)。随着社会越来越依赖于相关技术，技术上的失败将会带来更大的负面影响。质量可以帮助社会远离这些危险。软件开发中质量的地位比商业竞争更加重要，因此也更有理由来阅读、理解和应用本书中的观点。

Brian Thomas Eck, Ph. D.

Vice President

Juran Institute, Inc.

Wilton, Connecticut

前言

从历史的角度来看软件工程, 20 世纪 60 年代或更早可以看做是它的基础时代, 20 世纪 70 年代是发展时期, 20 世纪 80 年代是代价时期, 20 世纪 90 年代以及后来是质量和效率时期。在 20 世纪 60 年代, 我们学会了如何应用信息技术来满足机构的需求, 并开始将软件与机构的日常事务联系起来。在 20 世纪 70 年代, 工业界的特点是大规模的进度延迟和代价超支, 重点放在软件项目的计划和控制上。基于阶段的生命周期模型被引入和分析, 如人-月的分析开始涌现。在 20 世纪 80 年代, 硬件价格开始下降, 信息技术进入到机构的每一个方面, 并且开始走向个人。随着工业界竞争的日益激烈, 各个组织广泛实行了低代价应用。软件开发生产率的重要性大大增加, 开发并使用了许多的工程代价模型。到了 20 世纪 80 年代晚期, 质量的重要性开始得到人们的广泛认识。

20 世纪 90 年代及后来明显进入到质量时代。随着现有技术能够提供丰富的功能, 用户要求更高的质量。社会对软件日益增强的依赖也进一步强化了对质量的需求。账单错误, 大规模地受到破坏的电话服务, 甚至近些年来出现的导弹发射失败都可以追溯到软件开发中的有关问题。在这个时代, 质量已经放到了软件开发过程的中心。从软件销售商的角度来看, 质量不再是市场的一个有利因素, 而是变成了公司成功竞争的必要条件。

从 20 世纪 90 年代中期开始, 涌现出了两大主要因素, 被证明不仅对软件工程而且对全球商业环境都有重大影响: 为了提高效率的业务再工程 (reengineering) 以及 Internet。软件开发必须更加有效, 产品的质量级别必须要满足需求而且能够成功, 特别是对于关键性任务的应用。低质量的负面影响更加严重和广泛, 软件应该提供的质量“堤坝”至关重要。这些因素将会不断影响软件工程许多年。

测量在有效的软件开发中起到了关键的作用, 同时也为软件工程成为一个真正的工程学科提供了科学的基础。这本书描述了软件质量工程的度量和模型: 质量计划, 过程改进和质量控制, 过程中质量管理, 产品工程 (设计和代码复杂度), 可靠性预测, 以及客户满意度的分析。许多有关测量的书籍采用了百科全书的方法, 其中包括了每一种可能的软件测量。但这本书将其范围限制于软件质量的度量和模型中, 没有包括如代价预测、生产率、人员调度、性能测量等已经有大量文献的领域。

在本书的第二版中, 加入了 7 个全新的章节, 覆盖了软件测试的过程中度量, 面向对象度量, 可用性度量, 过程度量评价, 软件项目评价, 过程改进的注意事项, 以及测量软件过程改进。在第二版中删除了描述 AS/400 软件质量管理系统的章节。新版本在原先的章节中进行了一些更新和修改, 并加入了一些新的内容和图表。

其中有两章的新章节来自于两个专家的贡献, 这是新版本的一个特点。有关软件过程改进中的注意事项一章由 Patrick O'Toole 撰写。作为一个被广泛认同的过程改进专家且有着 20 多年的实际经验, Patrick 给这本书带来了关于过程改进方面的全新观点。这个角度是基于实际经验、面向项目并与商业策略一致的。有关测量软件过程改进一章是由 Capers Jones 提供的。Capers 是软件度量、生产率研究、软件质量控制以及软件评价的先驱, 他的著作在国际上都享有盛名。Capers 在软件评价和基准研究方面的基于数据和事实的方法是非常先进的。基于来自一万多个项目的经验和数据, 他为读者带来了用于软件过程改进的实践方法, 以及有关软件过程改进的主要发现。功能点度

量的价值通过分析及一些发现成果而得到了展现。这一章对于那些关注测量软件过程改进的软件过程专业人士来说是必不可少的。

这个版本的另一个新特点是,它为那些资源有限而又打算实现度量计划的小型团队和小型组织提供了一系列的建议和意见。这些建议分布在9章之中,用阴影标识。本书的一些例子是基于小型项目的,许多适用于大型项目的方法和技术也都适用于小型项目。这些建议从小型组织的角度出发,目的在于使用少量的度量来改进软件开发。

这本书针对软件质量专业人士、软件项目管理人员、软件产品管理人员、软件开发管理人员、软件工程师、软件产品保证人员,以及软件工程、管理信息系统、系统工程、质量工程和管理专业的学生。对于教师来说,本书能为本科高年级或研究生课程提供一个理论基础。美国和其他地区的一些软件工程、计算机科学、质量工程专业的教学计划已经采用了本书的第一版作为教材。

本书的主题

本书有几个主题。第一,权衡理论、技术和实例之间的关系,它为软件开发中质量工程的实践提供了一些可行的指导原则。尽管引用了一些公式,但重点在于理解和应用度量,而不是数学推导。本书引用的很多例子来自于 IBM Rochester 软件开发实验室(AS/400 和 IBM eServer iSeries 计算机系统的开发基地)以及软件工业中的其他一些公司。IBM Rochester 在 1990 年赢得了 Malcolm Baldrige 美国国家质量奖(MBNQA)。本书中的很多度量都是在那时使用的,从那时起许多度量经过了不断的开发与修改。所有的度量都得到了大量实践经验的验证。IBM Rochester 每年开发并发布许多不同大小和类型的项目,包括非常大且复杂的项目和小型项目;类型包括从固件、操作系统到中间件与应用。

第二,作者试图包括软件质量工程领域中尽可能多的度量和模型。除了关于度量和技术的一般讨论之外,这本书包括了四个类别的度量和模型:(1)质量管理模型;(2)软件可靠性和预测模型;(3)复杂性度量和模型;(4)客户观点度量、测量和模型。这些度量和模型覆盖了整个软件开发过程,从高层设计到测试和维护,以及关于可靠性的所有阶段。而且,尽管本书不是关于全面质量管理(TQM)的一本论著,但它在讨论度量时重点考虑了这一概念。TQM 的原则在于为了取得长远的成功,将产品质量和客户满意度联系在一起。TQM 可以解释为什么除了有关产品和过程度量的章节之外,本书还有两章是关于客户观点度量和测量的。在本书的其他一些讨论中,也将在适当的时候讲解客户观点。

第三,通过将度量和模型与质量改进策略和改进措施相结合,我们试图侧重于应用度量,而非描述度量。书中解释了过程度量和评估过程度量质量状态的基本框架,并提出了工作量/成果模型。我们说明了开发质量策略建议和缺陷移除模型之间的直接联系,并给出了与一些特定度量和分析有关的措施实例。而且,为了说明度量,本书使用了很多的图表。这也反映了在实际的项目和质量管理中,清晰的可视化表示形式通常能够增进理解并提高度量的有效性。

第四,在介绍了质量和过程改进之后,本书讨论了过程中的质量评价过程、一种评价软件项目的方法、过程改进中的注意事项和软件过程改进的定量分析。这些章节与关于度量和模型的章节之间的联系就是来自工业项目的实践经验。

本书的组织结构

下面,我们详细描述每一章的重点。

- **第1章，软件质量的定义：**讨论了质量和软件质量的定义，该定义强调了客户的地位。本章讨论了质量属性和它们的关系，给出了TQM的框架及其定义，以及客户对质量的观点，这是本书的一个重点。
- **第2章，软件开发过程模型：**介绍了许多在软件工业中使用的开发过程模型。本章大致介绍了软件过程成熟度评估的两种方法——SEI的过程能力成熟度模型（CMM）和SPR评价方法。我们还概括了两套质量管理标准——Malcolm Baldrige 美国国家质量奖评价标准和ISO 9000。
- **第3章，测量理论基础：**介绍了测量理论基础，这一点对软件测量非常重要。本章通过实例讲解了操作定义的概念和它在测量中的重要性，并且讨论了测量的级别、一些基本测量和Six Sigma的概念。测量质量、可靠性和有效性的两个关键标准，以及测量误差的有关问题，本章对它们的重要性都进行了说明。这一章还讨论了相关性，以及基于观测数据建立因果关系的必要标准。
- **第4章，软件质量度量概述：**给出了与软件生命周期相关的三个度量类别的例子：终端产品、过程中和维护阶段。本章描述了几家大型软件公司的度量计划，并讨论了软件工程数据的采集。
- **第5章，软件开发中七种基本质量工具的应用：**描述了质量控制的基本统计工具的应用，软件开发中称其为Ishikawa的七种工具。本章讨论在软件环境中应用控制图表的潜在作用和相关难点，并且介绍了用于显示复杂因果关系的定量工具——关系图。
- **第6章，缺陷移除效率：**这是描述软件开发质量问题的5章中的第一章。通过两种类型的模型，可以对质量管理模型、软件可靠性和预测模型、软件开发质量进行计划、工程化、管理与预测。这一章介绍了缺陷移除效率的核心概念、相关的测量，以及它在质量计划中的地位。
- **第7章，Rayleigh模型：**描述了模型以及它作为可靠性和预测模型的实现。Rayleigh模型作为质量管理模型的使用将在第9章讨论。
- **第8章，指数分布和可靠性增长模型：**讨论了指数分布和主要的软件可靠性增长模型。这些模型，如Rayleigh模型，在软件交付给用户之前（开发完成前）用于质量预测。这些模型还用于维护计划，从而对故障模式或领域中的缺陷到达模式进行建模。
- **第9章，质量管理模型：**描述了贯穿整个开发周期的几种质量管理模型。本章提出并讨论了过程度量和支撑模型的报告，而且给出了解释过程度量和评估过程质量状态的框架——工作量/成果模型。
- **第10章，软件测试的过程中度量：**本章是第9章的一个延续，侧重于软件测试的度量。工作量/成果模型通过在测试阶段的应用而得到了说明。本章还讨论了用于评估外包软件的验收测试的候选度量，以及关于如何得知产品可以交付的关键问题。
- **第11章，复杂性度量和模型：**讨论了软件工程中的第三类度量和模型。尽管质量管理模型和可靠性及预测模型是用于项目管理和质量管理，但是复杂性度量和模型的目的是为了使得软件工程师能够改进他们的设计以及软件开发的实现。
- **第12章，面向对象项目中的度量和经验教训：**覆盖了面向对象开发中的设计和复杂性度量、生产率度量、质量和质量管理度量，以及从开发和部署面向对象项目中得到的经验与教训。前面的内容可以看做是复杂性度量和模型讨论的延续，其他部分可以融入质量和项目管理的框架中。

- **第13章，可用性度量：**讨论系统可用性和中断度量，研究了可用性、可靠性和传统缺陷率测量之间的关系。可用性度量和客户满意度测量是第四类度量和模型——面向客户的度量。
- **第14章，测量和分析客户满意度：**讨论了客户满意度的数据采集和测量，分析客户满意度数据的模型和技术。从第3章到这一章，涵盖了度量和模型的整个范围。
- **第15章，执行过程中质量评价：**描述了过程质量评价作为好的项目管理的一个因素。质量评价基于定量的指标（如前面章节讨论过的）和定性的信息。
- **第16章，执行软件项目评价：**在另一个层次进行讨论，这一章提出了一种软件项目评价方法。本章的侧重点在项目层次，讨论来自于一个实践者的观点。
- **第17章，软件过程改进中的注意事项：**由 Patrick O' Toole 提供，为软件过程改进的专业人员提出了实际的建议。本章可以联系第2章的过程成熟度的讨论。
- **第18章，使用功能点度量来测量软件过程改进：**由 Capers Jones 提供，讨论了软件过程改进的六个阶段。基于大量的经验数据，本章研究了过程改进的代价和效果，其中显示关于代价、时间、进度、生产率和质量的定量分析。本章说明了功能点度量的价值，并联系了第2章的过程成熟度的讨论。
- **第19章，总结：**提供了对于一般软件测量以及一些特别的软件质量度量和模型的几个观察结果，并给出了关于软件工程测量未来前景的预示。
- **附录：**展现了一个真实的项目评价问卷的实例。对于第16章的每一种方法和技术，读者可以根据自己的项目评价工作量来定制问卷。

建议的阅读方式

读者应该从头至尾阅读本书的每一个章节，后面的章节将会引用前面章节提到的概念和讨论结果。同时，每一章都阐述了一个不同的主题，可以成组地学习有关章节。有些读者可以选择不同的主题阅读。例如，那些对质量定义、过程模型和测量基础不感兴趣的读者可以从第4章（“软件质量度量概述”）开始阅读，那些想要很快进入缺陷移除、质量计划、管理和预测度量的中心主题的读者可以从第6章（“缺陷移除效率”）开始学习。一般来说，作者建议读者按如下方式成组地阅读本书的章节。

- 第1章到第3章
- 第4章
- 第5章
- 第6章到第10章
- 第11章到第12章
- 第13章到第14章
- 第15章到第18章
- 第19章

目 录

第1章 软件质量的定义	1
1.1 质量：常规观点	1
1.2 质量：专业观点	1
1.3 软件质量	2
1.4 全面质量管理	4
1.5 小结	6
第2章 软件开发过程模型	8
2.1 瀑布开发模型	8
2.2 原型方法	12
2.3 螺旋模型	13
2.4 迭代开发过程模型	15
2.5 面向对象的开发过程	17
2.6 净室方法	20
2.7 缺陷预防过程	22
2.8 过程成熟度框架和质量标准	25
2.9 小结	33
第3章 测量理论基础	36
3.1 定义、操作定义和测量	36
3.2 测量的层次	38
3.3 一些基本度量	39
3.4 可靠性和有效性	45
3.5 测量误差	46
3.6 相关性的注意事项	49
3.7 因果关系的标准	51
3.8 小结	52
第4章 软件质量度量概述	54
4.1 产品质量度量	54
4.2 过程中质量度量	63
4.3 软件维护中的度量	67
4.4 度量规划的实例	70
4.5 采集软件工程数据	75
4.6 小结	79
第5章 软件开发中七种基本质量工具的应用	83
5.1 Ishikawa 的七种基本工具	83

5.2	检查列表	85
5.3	Pareto 图	86
5.4	直方图	88
5.5	游程图表	89
5.6	散点图表	90
5.7	控制图表	92
5.8	因果关系图	98
5.9	关系图	99
5.10	小结	100
第 6 章	缺陷移除效率	103
6.1	文献回顾	103
6.2	缺陷移除效率的进一步观察	106
6.3	缺陷移除效率和质量计划	111
6.4	阶段缺陷移除的代价效力	114
6.5	缺陷移除效率和过程成熟度级别	116
6.6	小结	118
第 7 章	Rayleigh 模型	120
7.1	可靠性模型	120
7.2	Rayleigh 模型	120
7.3	基本的假设	123
7.4	实现	124
7.5	可靠性和预测的有效性	130
7.6	小结	131
第 8 章	指数分布和可靠性增长模型	133
8.1	指数模型	133
8.2	可靠性增长模型	135
8.3	模型假设	139
8.4	模型评价的标准	139
8.5	过程建模	140
8.6	测试压缩因子	143
8.7	预测随时间的缺陷总数分布	144
8.8	小结	146
第 9 章	质量管理模型	150
9.1	Rayleigh 模型框架	150
9.2	代码集成模式	154
9.3	PTR 子模型	156
9.4	PTR 到达和积压预测模型	159
9.5	可靠性增长模型	161
9.6	模型评价标准	163
9.7	过程中度量和报告	164

9.8	正交缺陷分类	169
9.9	小结	172
第 10 章	软件测试的过程中度量	173
10.1	软件测试的过程中度量	173
10.2	过程度量和质量管理	187
10.3	外包软件的验收测试中的可能度量	191
10.4	如何了解产品是否达到交付标准	193
10.5	小结	194
第 11 章	复杂性度量和模型	196
11.1	代码行	196
11.2	Halstead 的软件科学	197
11.3	环形复杂性	198
11.4	语法构造	200
11.5	结构度量	201
11.6	一个实践中的模块设计度量的例子	203
11.7	小结	206
第 12 章	面向对象项目中的度量和经验教训	208
12.1	面向对象的概念和构造	208
12.2	设计和复杂性度量	209
12.3	生产率度量	215
12.4	质量和质量管理度量	217
12.5	面向对象项目的经验教训	220
12.6	小结	223
第 13 章	可用性度量	225
13.1	系统可用性的定义和测量	225
13.2	可靠性、可用性和缺陷率	227
13.3	为质量改进收集用户中断数据	229
13.4	中断和可用性的过程度量	233
13.5	小结	233
第 14 章	测量和分析客户满意度	235
14.1	客户满意度调查	235
14.2	分析满意度数据	238
14.3	对公司的满意度	243
14.4	客户满意度的目标	243
14.5	小结	245
第 15 章	执行过程中质量评价	248
15.1	准备阶段	248
15.2	评估阶段	250
15.3	总结阶段	253

15.4	推荐和风险缓解	254
15.5	小结	255
第 16 章	执行软件项目评价	257
16.1	审计和评价	257
16.2	软件过程成熟度评价和软件项目评价	258
16.3	软件过程评价周期	259
16.4	一种提出的软件项目评价方法	261
16.5	小结	270
第 17 章	软件过程改进中的注意事项	272
17.1	测量过程成熟度	272
17.2	测量过程能力	274
17.3	分阶段与连续——有争议的领域	274
17.4	只测量级别是不够的	274
17.5	建立一致性原则	275
17.6	花时间做到更快	276
17.7	保持简单化——或面对分解复杂度	277
17.8	测量过程改进的价值	278
17.9	测量过程的采纳	278
17.10	测量过程的依从性	279
17.11	庆祝旅程，不仅是终点	279
17.12	小结	280
第 18 章	使用功能点度量来测量软件过程改进	281
18.1	软件过程改进的顺序	282
18.2	过程改进经济学	284
18.3	基于活动的过程改进测量	286
18.4	小结	289
第 19 章	总结	291
19.1	数据质量控制	291
19.2	从软件度量计划开始	292
19.3	软件质量工程建模	294
19.4	软件开发中的统计过程控制	297
19.5	测量和其将来的发展	299
附录	项目评价问卷	302
术语表		318

第1章 软件质量的定义

如果要提高软件项目的质量,首先就需要定义质量和测量质量。然而,在质量工程和管理中,一个主要问题就是质量这个词是模棱两可的,以至于常常被错误地理解。产生这种概念混淆的原因主要有如下几个。首先,质量不是一个单一的概念,而是一个多维的概念。质量包括了问题中的实体、对实体的看法以及实体的质量特性。第二,对于任何概念来说,都具有几个抽象的层次。当人们说到质量时,一方面可能指广义的质量,另一方面可能指狭义的质量。第三,质量是我们日常用语的一部分,它的常规用法和专业用法也许存在着不同之处。

这一章主要讨论了人们对质量的一般看法,包括由质量专家定义的正式的质量概念及其含义,在软件中质量的意义和特殊用法,以及在全面质量管理中使用的方法和关键问题。

1.1 质量: 常规观点

通常我们认为质量是一个无形的特性——可以对其进行讨论、感知和判断,但不能进行测量。有些术语,如“好的质量”、“坏的质量”、“生活质量”都说明了人们如何谈论一些模糊的概念,他们并不想对这些概念进行定义。这种现象表明了人们使用不同的方式来观察和解释质量。这意味着不能对质量进行控制和管理,同时也不能将其定量化。这个观点同质量工程领域对质量的专业看法形成了鲜明的对比,专业领域认为可以且应该对质量进行实际的定义、度量、监控、管理和改进。

另一种常见的看法认为质量意味着豪华、等级和品位。人们通常认为昂贵、精致且较复杂的产品比那些在各个方面差一点的产品具有更好的质量。因此,凯迪拉克(Cadillac)是一种质量很高的汽车,但雪弗莱(Chevrolet)就稍差一些,无论两者的可靠性和维修记录如何。或者再举一个例子,一个环绕立体声音响系统的质量是好的,而单声道收音机的质量则不行。根据这种观点,高质量只局限于一小部分昂贵的产品,复杂的功能决定着产品的级别;而简单廉价的产品却很难成为质量好的产品。

1.2 质量: 专业观点

质量的常见看法通常很模糊并且将导致错误的理解,因此不能有助于工业领域中质量的提高。对于工业领域来说,质量必须有一个可行的定义。Crosby(1979)将质量描述为“与要求的一致性”,而Juran和Gryna(1970)则将质量描述为“良好的实用性”。下面我们将要看到,这两种定义相互关联且一致,许多质量专家都采用了这些定义。

“与要求的一致性”意味着要求必须清楚地表明到不能被误解的程度。然后,在开发和生产过程中,进行定期的测量来确保与这些要求的一致性。与要求不一致则被认为是缺陷——缺乏质量。例如,某种收音机的一个要求是必须可以接收到30英里(1英里=1.609 3千米)以外广播源的某些频率。假如收音机不能实现这一点,那么就说明它没有达到质量的要求,应拒绝使用。或者,如果凯迪拉克汽车符合了人们对该品牌的所有要求,那么这就是一辆质量好的汽车。如果雪弗莱汽车达到了品牌的要求,它同样也是一辆质量好的汽车。这两辆汽车也许在款式、性能和价格上有所不同,但都达到了各自的要求,所以它们都是质量好的汽车。