

软件度量

(原书第2版)

Software Metrics A Rigorous and Practical Approach (Second Edition)

(英) Norman E. Fenton 著
(美) Shari Lawrence Pfleeger
杨海燕 赵 巍 张 力 等译



TP311.56

169

软件度量系列

软件度量

(原书第2版)

Software Metrics
A Rigorous and Practical Approach
(Second Edition)

(英) Norman E. Fenton 著
(美) Shari Lawrence Pfleeger

杨海燕 赵巍 张力 等译



机械工业出版社
China Machine Press

软件度量是良好软件工程技术的一个重要组成部分。本书是介绍软件度量有关知识的一本经典教科书，内容丰富，覆盖面广，既包括重要的基础理论，又包括实用的指导准则，还引入了重要的研究成果。

本书内容分为三部分。第一部分是对测量和实验基础知识的介绍，包括测量理论、基于目标的软件测量框架、实证研究、软件度量数据的收集和分析。第二部分是对软件工程测量的各项内容的深入分析，包括：产品规模、结构等内部产品属性和可使用性、可维护性等外部产品属性的测量，可靠性的测量和预测，资源的测量和过程预测。第三部分是从管理层面对软件测量中的一些问题的说明，包括测量程序规划、测量实践和软件工程中的实证研究。书中提供大量实例，并附有习题和部分习题的答案，在附录中汇总一些度量工具的信息和缩略词，最后还列出丰富的附有评注的参考文献。

本书既可作为本科生和研究生学习软件度量的教材，也可作为软件管理人员和开发人员的工作参考手册。

Norman E. Fenton, Shari Lawrence Pfleeger.

Software Metrics: A Rigorous and Practical Approach (Second Edition).

EISBN: 0-534-95425-1

Copyright © 1997 by PWS Publishing Company, a division of Thomson Learning.

Original language published by Thomson Learning (a division of Thomson Learning Asia Pte Ltd). All rights reserved. 本书原版由汤姆森学习出版集团出版。版权所有，盗印必究。

China Machine Press is authorized by Thomson Learning to publish and distribute exclusively this simplified Chinese edition. This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). Unauthorized export of this edition is a violation of the Copyright Act. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

本书中文简体字翻译版由汤姆森学习出版集团授权机械工业出版社独家出版发行。此版本仅限在中华人民共和国境内（不包括中国香港、澳门特别行政区及中国台湾）销售。未经授权的本书出口将被视为违反版权法的行为。未经出版者预先书面许可，不得以任何方式复制或发行本书的任何部分。

981-254-916-1

本书版权登记号：图字：01-2000-2064

图书在版编目（CIP）数据

软件度量（原书第2版）/（英）芬顿（Fenton, N.E.），（美）弗莱格（Pfleeger, S.L.）著；杨海燕等译。
—北京：机械工业出版社，2004.9

（软件工程技术丛书 软件度量系列）

书名原文：Software Metrics: A Rigorous and Practical Approach, Second Edition

ISBN 7-111-14675-1

I. 软… II. ①芬… ②弗… ③杨… III. 软件—测试 IV. TP311.5

中国版本图书馆CIP数据核字（2004）第058019号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：王高翔 姚 蕾

北京牛山世兴印刷厂印刷·新华书店北京发行所发行

2004年9月第1版第1次印刷

787mm×1092mm 1/16·31.5印张

印数：0 001-4 000册

定价：65.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线：（010）68326294

译者序

测量是工程技术领域中一个不可或缺的要素，随着近年来软件工程领域的长足发展，测量技术也逐渐融入到软件工程领域，并成为良好软件工程的一个重要组成部分，因为“凡是不能测量的，便是不能对其进行控制的”。测量有助于对软件开发中的各项活动进行了解和控制，有助于过程和产品的改进。随着CMM的发展及其在国内日益广泛的应用，对软件开发中的过程、产品和资源进行量化的观点已被越来越多的人认同，由此也推动了对软件度量的研究。

本书是介绍软件度量有关知识的一本经典教材，内容丰富，覆盖面广，既包括重要的基础理论，又包括实用的指导准则，还引入了重要的研究成果。本书首先介绍测量、实验、数据收集和分析等方面的基本原理，然后具体介绍软件工程测量的各项内容，其中不乏对各种度量及其使用方法的有益考察，最后从管理层面对软件测量中的一些问题进行说明。本书用易于理解的描述方式向读者提供大量的信息，不仅包含精心挑选的实例和习题，还提供丰富的附有评注的参考文献，这些内容不仅有助于读者理解和巩固书本中的知识，还为读者进行深入研究提供了有益的指导。本书是有志于进行软件度量研究的学者的首选读物之一，也是业界实践人员的重要参考资料。

本书第1章至第8章、第13章及附录由杨海燕翻译，第9、10、14章及参考文献中的评注由赵巍翻译，第11、12、15章由张力翻译，此外，郎小凤、程杜平、尹建松和毛进也承担了本书的部分翻译、校对或文稿整理工作。本书的翻译得到了高仲仪教授的悉心指导，他为本书的翻译提出了很多宝贵的意见和建议，在此表示衷心感谢。

由于时间仓促，译文中难免有不当之处，敬请读者批评指正。

译者

2004年5月

前言

尽管一些人并不承认软件度量 (software metrics) 是良好软件工程技术的关键要素, 但是它确实发挥着十分重要的作用。测量 (measurement) 的用途极其广阔, 如评估状况、跟踪进展情况、评价有效性等。然而, 在如何进行测量和如何才能测量之间存在的差距, 仍然不如想像中那样让人乐观。之所以会在潜在可能性和具体实践之间存在这样的差距, 主要原因在于: 缺乏一个协调的、全面的框架来为理解和使用测量提供指导。通过鼓励对重要问题进行讨论、阐释基本概念和提出新方法来解决长期存在的问题, 1991年出版的本书第1版所介绍的严格的测量框架, 使测量的重要性得到了进一步的增强。

本书的第1版作为介绍软件度量的首批教材之一, 通过向软件工程师介绍测量的相关理论、图论的相关概念和软件可靠性的新方法, 开创了这个领域的新局面。本次出版的这本书对软件度量的最新发展情况进行了全面的阐述。为了能够对软件度量的迅猛发展情况 (尤其是软件度量在行业实践中日益受到人们的重视) 有所反映, 本书在原有基础上进行了全面的重新设计和编写。本书虽然再次介绍了软件度量的原始框架, 但是却把侧重点放在了框架的实际应用上。在测量理论的指导下, 在原来将实体分成产品、过程和资源的基础上, 本书对框架进行了扩展, 新增了过程可见性和面向目标 (goal-directed) 测量的概念。本书还通过补充大量的案例研究和实例, 对每一种想法和技术的应用情况进行了展示。

这个第2版还根据学术界和业界对第1版的反馈, 对有关内容进行了增删。因此, 呈现在大家面前的这本教材引入了大量的案例研究、精心剪裁的实例和习题。本书不但对原有的各个章节进行了完善和更新, 而且还增加了新的章节, 介绍了新的内容 (如过程成熟度和测量、目标-问题-度量范型 (paradigm)、度量的规划、实验、实证研究、面向对象的度量以及度量工具等)。本书在介绍软件度量时, 一如既往地保持了易于理解且力求全面的风格。

本书可供不同层次的读者群阅读。学术界和产业界的读者既可以将其作为软件度量和质量保证课程的初级教材, 也可以将其作为其他软件工程课程的有益参考资料。由于本书的覆盖面很广, 因此能够作为学者和实践人员的主要参考书。本书还介绍了一些只在研究型出版物中才会出现的有益的重要成果。当然, 从事软件度量研究的人士一定会对本书感兴趣, 因为它不但富含报告最新研究成果的材料, 而且还收录了大量附有评注的、有关测量信息的参考文献。另外, 本书还能 为软件管理人员和开发人员提供帮助——为他们建立或扩展测量程序提供指导, 这类人员可以把学习重点放在选择度量标准和规划度量使用等方面的实用指导上。

本书分为三个部分。第一部分对为什么要进行测量以及如何进行测量做了基本介绍, 其中分析并解释了测量、实验、数据收集和分析等方面的基本原理。第二部分深入探讨软件工程测量的各项内容, 全面介绍各种特定度量及其使用方法, 并用详实的实例和案例研究对其进行说明。第三部分从管理层面 对软件测量进行说明, 其中包括: 如何进行测量程序的规划? 其他组织在哪些方面已经获得了成功? 如何将测量用于对技术和工具进行有效性的评价? 此外, 本书还附有带评注的参考文献、术语表以及从主要章节中选出的部分习题的答案。

致 谢

多年以来，有不少人士为本书的编撰工作做出了贡献。在这里，笔者要特别感谢Barbara Kitchenham、Bev Littlewood和Peter Mellor，他们不但为本书的几个章节贡献了自己的智慧，而且还大力支持我们在新版中对其观点进行修改；感谢Tom DeMarco和Tom Gilb，他们不但对本书的原稿进行了认真的审校，而且还向笔者提出了宝贵的意见。

另外，笔者还要特别感谢南部银行工艺学校（South Bank Polytechnic）极富创造力的METKIT小组，尤其是Martin Bush、Ros Herman、David Mole、Meg Russell和Robin Whitty，他们为第1版的撰写、编辑和修改做出了很多贡献。下列同事为笔者提出了有益的建议：Nick Ashley、Richard Bache、Bill Bail、Vic Basili、Sarah Brocklehurst、Grubstake小组（Albert Baker、Antonia Bertolino、Jim Bieman、Dave Gustafson、Austin Melton、Linda Ott和Robin Whitty）、Tracy Hall、Les Hatton、Bill Hetzel、Gillian Hill、Chuck Howell、Darrel Ince、Agnes Kaposi、Chris Kemerer、Taghi Khoshgoftaar、Bob Lockhart、Bob Malcolm、Tom Maibaum、Steve Minnis、Margaret Myers、Martin Neil、Niklas Ohlsson、Ron Prather、James Robertson、Suzanne Robertson、Gerald Robinson、Terry Rout、Roger Shaw、Martin Shepperd、Claude Stricker、Lorenzo Strigini、Dale Tiller、Alan Todd、Klaas van den Berg、Hans van Vliet、Dave Wright、Marv Zelkowitz和Horst Zuse。Carol Allen和Basi Isaacs为笔者提供了办公支持，Chuck Pfleeeger和Stella Page为笔者提供了宝贵的技术和情感支持。笔者还要感谢Christof Ebert、Bob Grady、Armstrong Takang、Aimo Torn、Faridah Yahya所做的工作，以及指出本书第1版所存在问题（我们已经在本书中对这些问题做了相应的改正）的各方人士。

笔者还要再次感谢欧洲委员会，由欧洲信息技术研究战略计划（ESPRIT）资助的METKIT和PDCS项目为本书第1版的撰写提供了部分支持，而PDCS 2项目则为本书第2版的撰写提供了部分支持；英国电信研究实验室（British Telecom Research Laboratory）和NATO也为本书的初期工作提供了一些支持。我们还要衷心感谢英国工程与自然科学研究委员会（UK Engineering and Physical Sciences Research Council）和英国工贸部（UK Departement of Trade and Industry），他们的DESMET项目中的工作构成了本书第4章、第5章和第6章的核心内容，本书的大量实例很多都是选自其SMARTIE和DATUM项目。

目 录

译者序

前言

致谢

第一部分 测量的基本原理与实验

第1章 测量：何谓测量？为何测量？2

1.1 日常生活中的测量2

1.1.1 测量是什么3

1.1.2 使事物可测量4

1.2 软件工程领域的测量6

1.2.1 软件工程领域中测量的疏漏7

1.2.2 软件测量的目标8

1.2.3 测量的作用：了解、控制与改进10

1.3 软件度量的范围11

1.3.1 成本和工作量估计11

1.3.2 生产率模型和度量12

1.3.3 数据收集12

1.3.4 质量模型和度量13

1.3.5 可靠性模型13

1.3.6 性能评价和模型13

1.3.7 结构和复杂性度量14

1.3.8 通过度量实现管理14

1.3.9 方法和工具的评估15

1.3.10 能力成熟度评估15

1.4 小结15

1.5 习题16

第2章 测量基础17

2.1 测量的表示理论17

2.1.1 经验关系18

2.1.2 映射规则22

2.1.3 测量的表示条件22

2.2 测量和模型28

2.2.1 定义属性28

2.2.2 直接和间接测量30

2.2.3 用于预测的测量32

2.3 测量标度和标度类型35

2.3.1 标称标度37

2.3.2 序数标度37

2.3.3 间隔标度39

2.3.4 比率标度40

2.3.5 绝对标度41

2.4 测量中的意义42

2.4.1 度量的统计运算45

2.4.2 客观度量和主观度量48

2.4.3 扩展的数字系统中的测量48

2.4.4 间接测量和意义52

2.5 小结54

2.6 习题54

2.7 补充读物56

第3章 基于目标的软件测量框架58

3.1 软件度量的分类58

3.1.1 过程59

3.1.2 产品61

3.1.3 资源64

3.2 确定对什么进行测量65

3.2.1 目标-问题-度量范例65

3.2.2 测量和过程改进68

3.2.3 GQM与过程成熟度相结合73

3.3 框架的应用75

3.3.1 成本和工作量估计75

3.3.2 生产率度量和模型76

3.3.3 数据收集76

3.3.4 质量模型和度量77

3.3.5 可靠性模型77

3.3.6 性能评估和模型78

3.3.7 结构和复杂性度量78

3.3.8 能力成熟度评估	78	5.2.3 故障	127
3.3.9 通过度量实现管理	79	5.2.4 改动	131
3.3.10 方法和工具的评估	79	5.3 如何收集数据	132
3.3.11 数学家对度量的看法	80	5.4 何时收集数据	137
3.4 软件测量的确认	81	5.5 如何存储和提取数据	139
3.4.1 预测系统的确认	81	5.5.1 原始数据库结构	139
3.4.2 度量的确认	82	5.5.2 提取数据	140
3.5 软件测量确认实践	83	5.6 小结	141
3.5.1 更严格的确认需求	83	5.7 习题	142
3.5.2 确认和不精确定义	85	5.8 补充读物	145
3.5.3 如何才能不进行确认	85	第6章 软件测量数据分析	147
3.5.4 选择适当的预测系统	86	6.1 引言	147
3.6 小结	86	6.2 实验结果分析	147
3.7 习题	87	6.2.1 数据的性质	148
3.8 补充读物	88	6.2.2 实验的目的	154
第4章 实证研究	90	6.2.3 设计上的考虑因素	156
4.1 调查研究四项原则	90	6.2.4 决策树	156
4.1.1 选择研究方法	90	6.3 简单分析技术举例	157
4.1.2 提出假设	92	6.3.1 盒形图	157
4.1.3 对变量保持控制	93	6.3.2 散点图	159
4.1.4 使研究工作有意义	94	6.3.3 控制图	161
4.2 正规实验规划	96	6.3.4 关联的度量	163
4.2.1 实验的规程	96	6.3.5 稳健相关性	163
4.2.2 实验设计的原则	100	6.3.6 线性回归	165
4.2.3 实验设计的类型	103	6.3.7 稳健回归	166
4.2.4 实验设计的选择	106	6.3.8 多元回归	167
4.3 案例研究规划	114	6.4 高级方法	168
4.3.1 姐妹项目	114	6.4.1 分类树分析	168
4.3.2 基线	114	6.4.2 变换	169
4.3.3 随机选择	115	6.4.3 多元数据分析	171
4.4 小结	115	6.4.4 多准则决策支持	173
4.5 习题	115	6.5 统计检验概述	180
4.6 补充读物	116	6.5.1 单组检验	180
第5章 软件度量数据收集	118	6.5.2 双组检验	181
5.1 良好的数据应具备哪些条件	118	6.5.3 两个以上分组的比较	182
5.2 如何定义数据	119	6.6 小结	183
5.2.1 存在问题的问题	120	6.7 习题	183
5.2.2 失效	123	6.8 补充读物	185

第二部分 软件工程测量

第7章 测量产品内部属性: 规模	188
7.1 软件规模的各个方面	188
7.2 长度	190
7.2.1 代码	190
7.2.2 规格说明与设计	198
7.2.3 预测长度	198
7.3 重用	199
7.4 功能性	201
7.4.1 Albrecht方法	202
7.4.2 COCOMO 2.0方法	206
7.4.3 DeMarco方法	207
7.5 复杂性	208
7.5.1 测量算法的效率	208
7.5.2 测量问题的复杂性	213
7.6 小结	214
7.7 习题	215
7.8 补充读物	216
第8章 测量产品内部属性: 结构	217
8.1 结构度量的类型	217
8.2 控制流结构	218
8.2.1 结构的流图模型	219
8.2.2 层次化度量	226
8.2.3 测试覆盖度量	231
8.3 模块性和信息流属性	236
8.3.1 模块性和信息流模型	236
8.3.2 全局模块性	237
8.3.3 形态	238
8.3.4 树杂度	239
8.3.5 内部重用度	241
8.3.6 耦合	242
8.3.7 内聚	244
8.3.8 信息流	245
8.3.9 信息流—测试覆盖度量	247
8.4 面向对象度量	248
8.5 数据结构	250
8.6 全面“复杂性”测量的困难	251
8.7 小结	253
8.8 习题	254
8.9 补充读物	256
8.10 本章附录	259
第9章 测量产品外部属性	263
9.1 软件质量建模	263
9.1.1 早期模型	264
9.1.2 定义你自己的模型	267
9.1.3 ISO 9126标准质量模型	267
9.2 测量质量的几个方面	268
9.2.1 基于缺陷的质量度量	269
9.2.2 可使用性度量	274
9.2.3 可维护性度量	276
9.3 小结	280
9.4 习题	280
9.5 补充读物	281
第10章 软件可靠性: 测量和预测	282
10.1 可靠性理论基础	282
10.2 软件可靠性问题	287
10.3 参数式可靠性增长模型	290
10.3.1 Jelinski-Moranda模型	291
10.3.2 基于Jelinski-Moranda模型 的其他模型	293
10.3.3 Littlewood模型	293
10.3.4 Littlewood-Verrall模型	294
10.3.5 非齐次Poisson过程模型	294
10.3.6 对模型的普遍评论	295
10.4 预测的准确性	295
10.4.1 处理偏倚: u-图	296
10.4.2 处理噪声	299
10.4.3 prequential似然函数	299
10.4.4 选择最佳的模型	302
10.5 软件可靠性增长预测的再校准	303
10.6 运行环境的重要性	307
10.7 软件可靠性的其他方面	308
10.8 小结	310
10.9 习题	311

10.10 补充读物	312
第11章 资源测量：生产率、 团队和工具	314
11.1 生产率的含义	314
11.2 要测量哪些生产率	316
11.3 生产率测量	317
11.4 团队、工具和方法	320
11.4.1 团队结构	321
11.4.2 人员经验	322
11.4.3 方法和工具	323
11.5 小结	325
11.6 习题	326
11.7 补充读物	326
第12章 过程预测	328
12.1 优良的估计	328
12.1.1 什么是估计	328
12.1.2 评价估计的准确性	330
12.2 成本估计：问题和方法	332
12.2.1 成本估计问题	332
12.2.2 成本估计的通行方法	333
12.2.3 自底向上或自顶向下的估计	333
12.3 工作量和成本模型	334
12.3.1 基于回归技术的模型	335
12.3.2 COCOMO模型	336
12.3.3 Putnam的SLIM模型	340
12.3.4 多项目模型	341
12.4 现有建模方法存在的问题	342
12.4.1 模型结构	342
12.4.2 极度复杂的模型	343
12.4.3 产品规模估计	344
12.5 解决目前所采用估计方法 存在的问题	344
12.5.1 本地数据定义	344
12.5.2 校准	345
12.5.3 独立估计小组	345
12.5.4 减少输入的主观性	345
12.5.5 初步估计和重新估计	345

12.5.6 成本估计的可选择的规模度量	347
12.5.7 本地开发的成本模型	348
12.6 过程预测的含义	349
12.7 小结	351
12.8 习题	351
12.9 补充读物	352

第三部分 测量与管理

第13章 测量程序规划	354
13.1 何谓度量计划	354
13.2 “为什么”与“什么”：开发目标、 问题和度量	355
13.2.1 其他测量框架	356
13.3 何时及何处：将度量映射到活动	364
13.4 如何：测量工具	365
13.5 谁：测量者、分析者和受影响者	366
13.5.1 不同视角	367
13.5.2 角色与职责	367
13.6 修订计划	369
13.7 小结	370
13.8 习题	370
13.9 补充读物	371
第14章 测量实践	372
14.1 成功准则	372
14.2 小型测量	375
14.3 大型测量	381
14.3.1 美国国家航空航天局 —戈达德航天中心	381
14.3.2 惠普	382
14.3.3 Contel	385
14.3.4 西门子	385
14.3.5 日立	388
14.4 经验教训	389
14.4.1 度量	391
14.4.2 人员	391
14.4.3 程序	391
14.4.4 实现	391
14.5 小结	392

14.6 习题	392	15.4.1 净室研究	404
14.7 补充读物	394	15.4.2 面向对象	405
第15章 软件工程实证研究	395	15.4.3 审查	405
15.1 实证研究问题	395	15.5 测量的现在和未来	406
15.1.1 经验主义与倡导者	396	15.5.1 大量减少度量数量	406
15.1.2 优良的实验设计	396	15.5.2 技术引入效果的测量	407
15.1.3 模拟环境与真实环境	397	15.5.3 组织实证研究体系	408
15.1.4 差的测量与优良的测量	397	15.6 小结	409
15.1.5 短期审视与长期审视	398	15.7 习题	409
15.2 研究产品	398	15.8 补充读物	409
15.2.1 软件结构和维护	398	附录A 部分习题答案	411
15.2.2 规模、结构和质量	399	附录B 度量工具	426
15.3 研究资源	402	附录C 缩略词和术语表	428
15.4 研究过程	404	参考文献(附评注)	430

第一部分

测量的基本原理与实验

第1章 测量：何谓测量？为何测量？

第2章 测量基础

第3章 基于目标的软件测量框架

第4章 实证研究

第5章 软件度量数据收集

第6章 软件测量数据分析

第 1 章

测量：何谓测量？为何测量？

曾几何时，软件测量还是一项高深莫测的工作；到如今，它已经演变成成为良好软件工程技术的一个不可或缺的要害。许多优秀的软件开发人员都要通过测量（measure）软件的特性，了解下列内容：需求是否一致和完整，设计的质量是否较高，代码是否为即将进行的测试准备就绪；精干的项目经理都要通过对过程和产品的属性进行测量，来确定软件的发布时间和预算的执行情况；聪明的客户都要通过对最终产品的某些方面进行测量，来判断产品是否满足需求、以及产品的质量是否达到了让人放心的程度；另外，维护人员也都需要通过测量工作来对当前产品进行评估，以做出是否需要对产品进行升级和改进的判断。

本书对上述所有问题——但是，并不仅限于上述问题——都进行了探讨。首先，在前六章，对测量的基本原理和实验进行了讨论和说明，以便使读者能对下述内容有一个大致的了解：进行测量的意义何在？测量是如何为软件工程工具和技术的使用及有效性研究提供支持的？然后，在第7章到第12章，对软件工程测量的相关问题进行了深入细致的探讨，并附带地介绍了部分具体的度量（metrics）及其使用情况。最后，在第13章、第14章和第15章，从管理视角对软件测量进行了说明，分析了软件测量和度量的当前实践情况和未来发展趋势。总而言之，这些章节不但有着广泛的内容——几乎覆盖了软件工程测量的各个方面，而且有着足够的深度——使读者能够将书中的度量应用到自己的过程、产品和资源中。如果本书的读者还是一名学生，而且对于与集体成员一道开发项目、以解决商业和研究问题一无所知，那么，这本书将会告诉他：测量工作为何能够成为日常的开发和维护活动的一个和谐而有用的部分。

本章从对日常生活中常见测量活动的讨论入手，在第一节中首先说明了测量是如何成为理解、控制和改善环境的常见而必须的活动。在这部分内容里，读者将会明白：测量工作为什么需要严谨的作风？在第二节中讨论了测量在软件工程中发挥的作用——其中重点介绍了：测量需求是如何与进行软件开发时所设定的目标和所必须回答的问题相关的？然后，对软件工程测量和其他工程学科中的测量进行了一番比较，并提出了软件测量的具体目标。在最后一节中，对本书其余章节将要讨论到的、与测量有关的各个主题做了一个提纲挈领的说明。

1.1 日常生活中的测量

在主宰我们生活的许多系统中，测量活动始终处于核心地位。比如，经济领域内的测量活动对价格的涨跌、薪水的增减起着决定作用；雷达系统中的测量使我们能够探测到用肉眼看不到的飞行器的存在；医疗系统中的测量能够帮助医生诊断出某类疾病；大气探测系统中的测量是天气预报的基础活动。如果没有测量的存在，技术所能发挥的作用微乎其微。

但是，测量活动并不仅仅存在于专业技术领域。在日常生活中，我们每个人都会用到测量。比如，价格就是商店里商品价值的一种度量（measure），通过计算购物总额，可以判断店主是

否给了我们正确的找零；通过对身高和尺码等进行测量，我们可以判断出一件衣服是否合身；在计划某次旅行时，我们可能会计算距离、选择路线、测量速度、预计抵达目的地的时间（也许还会预计需要加油的时间）。测量不但有助于我们了解自己所处的世界，以便与自己周围的环境相适应，而且有助于我们提高自己的生活质量。

1.1.1 测量是什么

从上面介绍的这些例子中，我们可以看到，人们使用测量的方式千差万别。但是，在上文所描述的所有活动中，存在着一条贯穿其中的相同线索：在每种情况下，都要为事物的某个方面设定一个描述符（descriptor），通过这个描述符，我们可以在该描述符与其他描述符之间进行比较。例如，在商店里，我们可以对一件商品的价格和另一件商品的价格进行比较；在服装店里，我们可以对各种尺码进行比较；在旅途中，我们可以对已经走过的路程和剩下的路程进行比较。在这些例子中，虽然设定描述符和进行比较的规则并不是十分明了，但是，我们显然是根据一组定义良好的规则来进行比较和计算的。让我们通过下面对测量的正式定义来加深对这一概念的理解：

所谓**测量**（measurement），是指这样一个过程，即通过这一过程以某种方式为真实世界实体的属性赋予数字或者符号，以便能够按照明确定义的规则对属性进行描述。

所以，通过测量，能够捕获关于实体属性的信息。所谓**实体**（entity），是指真实世界中的某个对象（如一个人、一个房间）或者某个事件（如一次旅行、软件项目的测试阶段）。人们总是希望能够通过识别出那些对区分不同实体起着重要作用的特性来描述实体。所谓**属性**（attribute），是指实体的特征或性质，比如，（房间的）面积和颜色、（旅行的）花费和（测试阶段）耗费的时间，就是一些具有代表性的属性。从通常意义上讲，实体和属性之间并没有严格的区分界限。比如，我们说“今天的天气很冷”，其实也是在说今天的空气温度很低；再如，我们说“她长得比他高”，实际上也是指“她的身高超过了他的身高”。虽然这种不严格的术语在日常口语中能够得到人们的认可，但是，从科学的角度来说，它却是不正确、不恰当的。所以，“测量事物”和“测量属性”的说法都是错误的——事实上，人们“测量的是事物的属性”。“测量房间”的说法之所以是模棱两可的，是因为我们其实测量的是房间的长度、面积和温度。同理，“测量温度”的说法之所以是含糊不清的，是因为我们测量的是在特定条件下、特定地理位置的温度。换句话说，在工程师和科学家的眼中，日常会话中司空见惯的表达方式可能是不正确的。

当我们使用属性来描述实体的时候，我们通常要用数字或者符号来定义属性。这样一来，价格就用美元或者英镑的数量来定义，而高度则采用英寸或者厘米来定义。同理，衣服的尺码可以说成是“小号”、“中号”和“大号”，而燃油的品质则可以说成是“普通”、“优质”和“超优”。这类数字和符号是人们反映自己感知真实世界所采用的抽象物。比如，在定义这些数字和符号的时候，人们会尽其所能，以期能够真实反映自己所感知到的、实体之间的关系。所以，六英尺高的人比五英尺高的人要高；同理，“中号”的T恤衫比“大号”的T恤衫要小。这类数字或符号不但非常重要，而且还非常有用。假定我们没有见过Herman这个人，但是我们知道他身高七英尺。在这种情况下，即使不能亲眼见到Herman，我们也能推算出自己和他之间的高矮关

系。而且，从他非常高这一点判断，他进入我们办公室时肯定需要弯一下腰。由此可见，人们能够仅仅凭借对实体属性的掌握和分析，对实体做出相关的判断。

测量是一个过程，人们对其定义仍然存在不小的分歧。由于在这个问题上存在多如牛毛且五花八门的权威观点，使得对测量构成要素的解释出现了因人而异的局面。如果要搞清楚测量的概念，就必须弄懂许许多多的难以回答的问题。兹举例如下：

- 我们已经注意到，颜色是房间的一个属性。那么，对于一个墙壁为蓝色的房间来说，“蓝色”是房间颜色的一个度量吗？
- 人的身高是一个公认的可以进行测量的属性。但是，人的其他属性，如智力，是否可以测量呢？用IQ测试的分数来测量人的智力是否充分？类似的问题，酒可以用酒精含量（“低”）来测量，但是，酒的质量是否能用专家评定的方法来测量呢？
- 度量（measure）的准确性对测量仪器和测量的定义有很大的依赖性。例如，只要尺子的刻度精确，而且其使用方法正确，测出的长度就是准确的。但是，有的测量似乎不可能做到准确，这有两方面的原因：可能是因为测量过程不准确，也可能是因为准确性依赖于进行测量的人的判断水平。比如，前面提到的“人的智力”和“酒的质量”的度量就存在着一个误差范围。这能否成为否认其作为名副其实的测量的理由呢？
- 即使测量装置是可靠的、其使用方法也是正确的，即使是在测量理解得非常透彻的物理属性时，也仍然会存在误差范围。比如，同一个人的身高可能会产生许多种度量结果，因为它不仅取决于是否允许人穿鞋，而且还与人的站立姿势有关。这样一来，应该如何决定“多大的误差范围是可以接受的、多大的误差范围是不能接受的”？
- 高度的测量标度可以是米、英寸或英尺，这些不同标度测量的是同一个属性。当然，高度的测量标度也可以是英里和公里——这类标度适于用来测量卫星距离地面的高度，但不适于用来测量人的身高。对于标度的用途来说，它在什么情况下是可接受的呢？
- 一旦得到测量的结果，就需要对它进行分析，并为被测实体下一个结论。人们能够对测量结果进行何种处理？例如，为什么我们可以说“Fred的身高是Joe的两倍”，但却不能说“今天是昨天的两倍热”？为什么计算某一组高度的平均值意义重大（比如，伦敦建筑群的平均高度为200米），而计算某个足球队运动衫上的号码的平均值却毫无意义？

为了回答上述问题以及其他方面的诸多问题，本书在第2章对测量科学进行了深入的分析——之所以要采用这种严谨的研究方法，其目的在于：为读者将测量概念应用到软件工程的实际问题中打下牢固的基础。但是，在介绍测量理论之前，让我们首先来分析一下：什么样的事物是可测量的？

1.1.2 使事物可测量

“正是那些不可测量的事物，才使可测量得以诞生”。

这句名言流传在关于测量科学家的民间传说中，相传是由伽利略（1564–1642）悟出的（Finkelstein, 1982）。有人认为，科学的目标之一，便是为人们感兴趣的事物寻找测量其属性的方式。这句话所蕴含的思想是，测量能够使概念变得更直观，进而变得更容易为人所理解和控制。因此，作为科学家，我们所肩负的历史使命是：发明能够测量我们周围的世界的方法；在

已经能够进行测量的地方，将测量工作开展得更好。

在物理学、医学、经济学、甚至是某些社会科学领域中，人们现在已经能够对一些曾经认为是不可测量的属性进行测量。而且，无论人们喜欢与否，对属性的测量，比如人的智力、空气质量、经济的通货膨胀等的测量，已经成为影响人们在日常生活中进行的重要决策的基础性因素。当然，有些测量并不如人们所想象的那么精细（refined）（这是从第2章“进行精确处理”的意义上说的）；人们把物理学奉为良好测量的典范，一直在改进度量的道路上不断地追求着完美。但是，千万不要忘了：远古的人们曾经一度认为，时间、温度和速度等概念是不可测量的；而在今天，对这些概念的测量不但是一件司空见惯的事情，而且任何人都能够轻而易举地进行测量工作——这类测量早已成为人类存在有机体的一个组成部分。

如果要提高软件工程领域中测量的严格性，就不应该限制测量的类型和范围。事实上，通过对不可测量的事物进行测量，不但可以增进人们对某些实体和属性的了解程度，而且还可以使软件工程与其他工程学科具有同样的强大能力。虽然人们对如何能够测量某个属性尚不明了，但是，提出这类测量的活动本身将会引起激烈的讨论，这样的讨论有助于人们更加深刻地认识问题。尽管部分软件工程师坚持认为，某些重要的软件属性，如可信性（dependability）、质量、可使用性（usability）和可维护性等，是绝对不可能进行量化的，但是，人们仍然愿意通过尝试使用测量来增进自己对这种属性的了解。

值得注意的是，从严格意义上讲，只存在如下两种类型的量化：测量和计算。测量（measurement）是一种直接的量化方法，比如，一棵树的高度和一车砖的重量，就属于测量的范畴。而计算（calculation）则是一种间接的量化方法；在这里，首先需要进行测量，然后再将得到的测量结果组合成一个可以反映人们希望了解其数值的属性的定量项。例如，城市检查人员在给房屋估价时（根据房屋的估价决定房主的应缴税额），需要通过使用某个计算公式将许多个因素（包括房间的间数、取暖方式与散热方式以及居住面积）组合在一起。估价是一种量化方法，但却不是测量；它是以数字形式表现出来的，这使得它比单纯进行定性评估要有用得多。本书用直接（direct）和间接（indirect）来区分测量和计算，详见第2章的相关内容。

体育运动为我们用较为客观的方式测量抽象属性（如质量）提供了许多宝贵的经验。该领域所采用的度量已经得到了广泛的认可——尽管偶尔还会出现关于修改和完善这些度量的讨论。下面的例子在强调有关测量的各种概念的同时，对如何在软件工程中发挥其作用进行了说明：

例1-1：在十项全能运动中，需要测量的内容包括：运动员跑完不同距离分别花费的时间，以及在不同跳跃项目中跳跃的距离。然后，以各个度量的重要性为基础，采用加权的方法，计算出一个总成绩——有时需要根据项目的相对重要性或度量的变化情况，对其中的权值进行相应的修改。用这样的总成绩来衡量一个运动员的综合能力的做法已经得到了广泛的认可。事实上，十项全能比赛的奥运冠军，通常都被认为是世界上最好的运动员。

例1-2：英国足联采用记分制来评选一个赛季的最佳球队。1981年，足联对记分制进行了修改：胜一场得3分、而不再是2分，打平仍然得1分。做这样的修改，是反映大多数人的共同意见，因为从性质上讲，胜一场比赛和平一场比赛之间的差距比平一场比赛和输一场比赛之间的差距要大。

例1-3: 在最佳足球运动员的评选方面,至今还没有一个公认的度量(当然,对于前锋来说,进球数算得上是一种比较准确的度量)。尽管很多球迷和球员都认为,球员的质量是一种不可测量的属性——这个问题早在1994年美国世界杯之前就已经提了出来。为了能够用某种客观(可测量的)的方法来评选“最佳球员”,人们提出了几种新的测量方法:

“为了帮助国际足联(FIFA)评选最佳足球运动员,有必要增加球场标志竿的数量,以便在球场上形成纵向和横向间隔距离为10米的分割线。这样,就可以测出球场上所有球员的准确传球码数、边路传球码数、盘球码数和头球码数。”

翻译摘录自“1994年世界杯的比赛规则可能会发生的变化”
Nouveaux FIFA d'Arbitres (FIFA Referees News), 1990年3月。

有人建议加入这些测量,并与进球个数,触球、救球或断球情况,以及传球(各种类型的)、盘球和头球的频率和距离一起进行加权处理。需要注意的是,如果要采用这里提出的有关球员质量的新度量,就需要对举行比赛的物理环境进行变更。

存在于软件工程领域内的类似情况不胜枚举。在很多情况下,人们都希望得到这样一个总体评分:该评分由若干个度量组合而成,并且能够反映开发和维护情况的“全貌”。人们都希望自己能够通过一组度量(每个度量都各自表征着“良好性”的某个方面)来判断软件产品质量的好坏。而且,人们都希望能够测量出一个组织所具备的、生产优质软件的能力,并且测量出一个模型对软件开发过程进行准确预测的能力。这种复合度量之所以会引起争议,一方面是因为组成复合度量的各个度量的缘故,另一方面是因为赋予的权值的缘故。

与前述情况相似,当我们希望获得与软件工程的某个方面相关的定性信息时,常常会引发激烈的争论——不同的专家持不同的观点,在有些时候甚至无法达成一致的意见。

最后,在某些情况下,为了测量新事物、或者采用某种新方式来测量事物,需要对现有的环境和实践方式进行适当的修改。在修改过程中,引入下列方法都是可行的:采用新的工具(来计算代码行数和评价代码结构)、在过程中增加新的步骤(来报告工作量)、以及采用新的方法(使得测量变得更简单)。变更往往都是让人难以接受的,究其原因,是因为:无论是测量程序已经得到实施、还是做过修改,都会碰到管理方面的问题(本书的其他章节还会对这个问题进行专门的讨论)。

1.2 软件工程领域的测量

通过前面的介绍,我们已经认识到,测量是人们日常生活中的基本要素,测量已经成为一项司空见惯的活动,而且,人们对测量已广泛认可。本节将通过对软件工程领域的分析,帮助读者了解:软件工程领域为什么需要测量?

软件工程(software engineering)所描述的是这样一些技术,即将工程方法应用于软件产品的构造和支持中。软件工程活动一般包括如下一些内容:管理、成本计算、计划、建模、分析、详细说明(specifying)、设计、实现、测试和维护。所谓“工程方法”,是指每个活动都是得到了充分的理解和控制,以致于在对软件进行详细说明、设计、构建和维护时,很少会有意外发生。既然计算机科学为软件的构建提供了理论基础,所以,软件工程应该是致力于:以一种可