新世纪高等学校规划教材 · 软件工程系列

软件测试技术及实践

李海生◎编著

北京师范大学出版集团
BEIJING NORMAL UNIVERSITY PUBLISHING GROUP
北京师范大学出版社

新世纪高等学校规划教[illegible]

软件测试技术及实践

李海生◎编著

北京师范大学出版集团
BEIJING NORMAL UNIVERSITY PUBLISHING GROUP
北京师范大学出版社

图书在版编目(CIP)数据

软件测试技术及实践/李海生编著. —北京：北京师范大学出版社，2019.2
（新世纪高等学校规划教材·软件工程系列）
ISBN 978-7-303-21761-8

Ⅰ.①软… Ⅱ.①李… Ⅲ.①软件—测试—高等学校—教材 Ⅳ.①TP311.55

中国版本图书馆 CIP 数据核字(2016)第 302781 号

营销中心电话 010-62978190 62979006
北师大出版社科技与经管分社 www.jswsbook.com
电子信箱 jswsbook@163.com

出版发行：北京师范大学出版社 www.bnup.com
北京市海淀区新街口外大街 19 号
邮政编码：100875
印 刷：保定市中画美凯印刷有限公司
经 销：全国新华书店
开 本：787 mm×1092 mm 1/16
印 张：15.25
字 数：306 千字
版 次：2019 年 2 月第 1 版
印 次：2019 年 2 月第 1 次印刷
定 价：36.80 元

策划编辑：赵洛育 责任编辑：赵洛育
美术编辑：刘 超 装帧设计：刘 超
责任校对：黄 华 赵非非 责任印制：赵非非

内容简介

本书系统介绍了软件测试的基础知识、原理方法和技术。全书围绕软件测试的流程，详细论述了软件测试的基本原理、软件测试计划与策略、黑盒测试、白盒测试、单元测试、集成测试、系统测试、验收测试、软件自动化测试和软件测试管理等内容。

本书结合教学实例突出基本知识和基本概念的表达，注重软件测试理论与实践相结合，内容详尽，并提供了 8 个软件测试实验，使读者能更好地理解和掌握软件测试的内容，并迅速运用到实际的软件测试工作中去。

本书可作为普通高等院校计算机相关专业的软件测试课程教材，也可作为软件测试技术学习提高的培训教材，还可供从事软件开发和软件测试工作的技术人员参考使用。

前　言

近年来，随着计算机技术的进步，以软件系统为核心的计算机应用已经渗透到各行各业，在国民经济、国防领域以及人们的日常生活中发挥着越来越重要的作用。随着软件规模的不断增长，软件质量问题逐渐成为制约计算机发展和应用的瓶颈之一。各种类型的软件错误层出不穷，不仅给人们日益信息化的日常工作和生活带来极大的不便，在很多关键领域甚至会造成财产损失乃至人身伤害。为了提高软件质量，尽可能地减少软件中的错误，人们尝试使用各种先进的软件开发技术，并辅之以更加合理的生产管理手段，以提高软件产品的质量。其中一个重要手段就是软件测试。

软件测试是软件工程专业、计算机科学与技术专业、信息管理与信息系统、数据科学与大数据等电子信息类专业的一门重要的专业课。软件测试课程涉及软件测试的理论、方法、技术和工具，是一门实践性、应用性很强的课程。软件测试是使用人工或自动手段来运行、检测软件系统的过程，是软件开发必不可少的环节和软件工程实践的重要组成部分。

本书从理论和实践的角度介绍了软件测试与测试技术，所叙述内容反映了当前的软件测试技术，帮助学生熟悉软件测试岗位的工作职责，了解软件测试的基本概念和方法、原则、规范等，掌握软件测试的工作流程、测试技能，培养学生的实际动手操作能力和专业实践能力。

全书共分 10 章，分别介绍了软件测试概述、软件测试计划与策略、黑盒测试、白盒测试、单元测试、集成测试、系统测试、验收测试、软件自动化测试和软件测试管理等内容。并在附录中，结合教师授课需要，设计了 8 个软件测试实验，每个实验相对独立，可供学生在两小时的上机实验中完成。

本书的编排依据软件测试 V 模型，采用循序渐进的方式，深入浅出地介绍各种软件测试知识，适合学生逐步掌握软件测试的基本方法、了解软件测试设计和管理的精髓。

本书在写作过程中，参考了大量的技术资料和有关书籍、文章和来源于网络的软件测试从业者的经验，并引用了有关书籍、文章里的多幅图表最终写作而成。本书在写作过程

中得到了许多热心朋友的支持和帮助，对于引用的文献名称和作者就不再一一列出，在此一并表示感谢！

本书适合普通高等学校计算机软件相关专业软件测试课程的教材，也可作为软件测试技术学习提高的培训教材，还可供从事软件开发和软件测试工作的技术人员参考使用。

虽然编者力求完美，但由于水平有限，书中难免有疏漏之处，敬请读者不吝赐教。

编　者

2019 年 1 月

目　录

第 1 章　软件测试概述

第 2 章 软件测试计划与策略

第 3 章 黑盒测试

第 4 章 白盒测试

第5章 单元测试

第 6 章 集成测试

第 7 章 系统测试

第 8 章 验收测试

第 9 章 软件自动化测试

第 10 章 软件测试管理

附录 A 软件测试课程实验安排

附录 B 代码评审

第1章 软件测试概述

学习目标 ▸▸▸

- 掌握软件质量、软件缺陷、软件测试、软件测试V模型的概念。
- 熟悉软件测试用例设计的方法。
- 了解软件测试中的误区，软件测试人员应具备的素质。

计算机系统分为硬件系统和软件系统两大部分。在过去的50多年时间里，计算机硬件技术得到了极大的发展，现在已经达到了相当成熟的状态。然而，随着计算机硬件技术的飞速发展，人们对计算机的需求和依赖与日俱增。随之而来的是计算机软件系统的规模和复杂性急剧增加，其软件开发成本以及由于软件故障而造成的经济损失也正在增加，软件的质量问题已成为人们关注的焦点。软件测试是保证软件质量的主要手段，近年来，也受到了人们的广泛关注。社会对软件测试人员的需求迅速增长。

1.1 软件、软件危机与软件工程

什么是软件？这个问题既简单又不太好回答。人们几乎每天都在使用各种各样的软件，如Windows、Office、IE浏览器、媒体播放器等，它们都是人们再熟悉不过的产品了，但是否真正全面地理解什么是软件，大多数人不敢肯定。那软件真正的含义是什么？

现在普遍被人们认可的软件的定义如下：

- 能够完成预定功能和性能的、可执行的指令(计算机程序)。
- 使程序能够适当地操作信息的数据结构。
- 描述程序的操作和使用的文档。

即“软件＝程序＋数据(库)＋ 文档”，在这里给出了软件最基本的组成成分。实际上，还少了一项内容：服务。所以可以用一个简单的公式给出软件的定义：

软件＝程序＋数据(库)＋ 文档＋服务

20世纪60年代以前，计算机刚刚投入实际使用，软件设计往往只是为了一个特定的应用而在指定的计算机上设计和编制，采用密切依赖于计算机的机器代码或汇编语言，软

件的规模比较小，文档资料通常也不存在，很少使用系统化的开发方法，设计软件往往等同于编制程序，基本上是个人设计、个人使用、个人操作、自给自足的私人化的软件生产方式。

20 世纪 60 年代中期，大容量、高速度计算机的出现，使计算机的应用范围迅速扩大，软件开发急剧增长。高级语言开始出现，操作系统的发展引起了计算机应用方式的变化，大量数据处理导致第一代数据库管理系统的诞生。软件系统的规模越来越大，复杂程度越来越高，软件可靠性问题也越来越突出。原来的个人设计、个人使用的方式不再能满足要求，迫切需要改变软件生产方式，提高软件生产率，软件危机开始爆发。

1968 年北大西洋公约组织的计算机科学家在德国召开的国际学术会议上第一次提出了“软件危机”(software crisis)这个名词。软件危机主要表现在以下几个方面。

(1)软件开发费用和进度失控。费用超支、进度拖延的情况屡屡发生。有时为了赶进度或压成本不得不采取一些权宜之计，这样又往往严重损害了软件产品的质量。

(2)软件的可靠性差。尽管耗费了大量的人力、物力，而系统的正确性却越来越难以保证，出错率大大增加，由于软件错误而造成的损失十分惊人。

(3)生产出来的软件难以维护。很多程序缺乏相应的文档资料，程序中的错误难以定位，难以改正，有时改正了已有的错误又引入新的错误。随着软件的社会拥有量越来越大，软件维护占用了大量人力、物力和财力。

(4)软件成本在计算机系统总成本中所占的比例居高不下，且逐年上升。由于微电子学技术的进步和硬件生产自动化程度不断提高，硬件成本逐年下降，性能和产量迅速提高。然而软件开发需要大量人力，软件成本随着软件规模和数量的剧增而持续上升。

(5)软件生产不能满足日益增长的软件需求，软件生产率远低于硬件生产率和计算机应用的增长率，出现了软件供不应求的局面。更为严重的是，软件生产效率随软件生产规模的增加和软件复杂性的提高而急剧下降。

(6)软件系统实现的功能与实际需求不符。软件开发人员对用户需求缺乏深入的理解，往往急于编写程序，闭门造车，最后完成的软件与用户需求相距甚远。

出现软件危机的原因，一方面是与软件本身的特点有关；另一方面是与软件开发和维护的方法不正确有关。从宏观上看，软件危机的实质是软件产品的供应赶不上需求的增长；从微观上简单地说，软件危机就是开发的软件有错误，软件质量达不到要求，软件项目无法按时完成，软件项目的花费超预算。

为了解决软件危机，既要有技术措施，又要有必要的组织管理措施。1968 年秋季，北大西洋公约组织的科技委员会召集了近 50 名一流的编程人员、计算机科学家和工业界巨头，讨论和制定摆脱“软件危机”的对策。在那次会议上第一次提出了软件工程(software engineering)这个概念。软件工程正是从技术和管理两方面研究如何更好地开发和维护计算机软件的一门学科。

软件工程主要研究软件生产的客观规律性，建立与系统化软件生产有关的概念、原则、方法、技术和工具，指导和支持软件系统的生产活动，以期达到降低软件生产成本、改进软件产品质量、提高软件生产率水平的目标。软件工程学从硬件工程和其他人类工程中吸收了许多成功的经验，明确提出了软件生命周期的模型，发展了许多软件开发与维护阶段适用的技术和方法，并应用于软件工程实践，取得良好的效果。

1.2　软件质量与质量模型

1.2.1　软件质量

软件质量是一个模糊的、捉摸不定的概念。人们常常听说：××软件好用；××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作是软件质量评价，特别不能算作是软件质量科学的定量评价。但是，软件质量，乃至于任何产品质量，都是一个很复杂的事物性质和行为。对于什么是产品质量，可以从以下几个观点来看。

(1)透明性观点：质量是产品可以认识但不可定义的性质。

(2)使用者观点：质量是产品满足使用目的之程度。

(3)制造者观点：质量是产品性能和规格要求的符合度。

(4)产品观点：质量是联结产品固有性能的纽带。

(5)基于价值观点：质量依赖于顾客愿意付给产品报酬的数量。

而有关软件质量的定义又有不同的描述。

1979 年，Fisher 和 Light 将软件质量定义为：表征计算机系统卓越程度的所有属性的集合。

1982 年，Fisher 和 Baker 将软件质量定义为：软件产品满足明确需求一组属性的集合。

又如，ANSI/IEEE Std 729－1983 定义软件质量为“与软件产品满足明确的和隐含的需求的能力有关的特征或特性的全体”，实际上反映了三方面的问题：

(1)软件需求是度量软件质量的基础。

(2)只满足明确定义的需求，而没有满足应有的隐含需求，软件质量也无法保证。

(3)不遵循各种标准定义的开发规则，软件质量就得不到保证。

按照《GB/T 16260－1996(idt ISO/IEC9126：1991)信息技术软件产品评价质量特性及其使用指南》，软件质量(software quality)是与软件产品满足明确的或隐含的需求的能力有关的特征和特性的总和。

20 世纪 90 年代，Norman、Robin 等将软件质量定义为：表征软件产品满足明确的和隐含的需求的能力的特性或特征的集合。

1994 年，国际标准化组织公布的国际标准 ISO 8042 综合将软件质量定义为：反应实体满足明确的和隐含的需求的能力的特性的总和。

综上所述，软件质量是产品、组织和体系或过程的一组固有特性，反映它们满足顾客和其他相关方面要求的程度。如 CMU SEI 的 Watts Humphrey 指出：“软件产品必须提供用户所需的功能，如果做不到这一点，什么产品都没有意义。其次，这个产品能够正常工作。如果产品中有很多缺陷，不能正常工作，那么不管这种产品性能如何，用户也不会使用它。”而 Peter Denning 强调：“越是关注客户的满意度，软件就越有可能达到质量要求。程序的正确性固然重要，但不足以体现软件的价值。”

GB/T 11457－2006《软件工程术语》中定义软件质量为：

(1)软件产品中能满足给定需要的性质和特性的总体。

(2)软件具有所期望的各种属性的组合程度。

(3)顾客和用户觉得软件满足其综合期望的程度。

(4)确定软件在使用中将满足顾客预期要求的程度。

简言之，软件质量是软件一些特性的组合，它仅依赖于软件本身。也就是说，为满足软件的各项精确定义的功能、性能需求，符合文档化的开发标准，需要相应地给出或设计一些质量特性及其组合，作为在软件开发与维护中的重要考虑因素。如果这些质量特性及其组合都能在产品中得到满足，则这个软件产品质量就是高的。软件质量反映了以下三方面的问题：

(1)软件需求是度量软件质量的基础，不符合需求的软件就不具备质量。

(2)在各种标准中定义了一些开发准则，用来指导软件人员用工程化的方法来开发软件。如果不遵守这些开发准则，软件质量就得不到保证。

(3)往往会有一些隐含的需求没有明确地提出来。例如，软件应具备良好的可维护性。如果软件只满足那些精确定义了的需求而没有满足这些隐含的需求，软件质量也不能保证。

综上，软件质量是各种特性的复杂组合。它随着应用的不同而不同，随着用户提出的质量要求不同而不同。用户主要感兴趣的是如何使用软件、软件性能和使用软件的效果。通常用户关心软件的以下特性。

(1)软件是否具有所需要的功能。

(2)软件可靠程度如何。

(3)软件效率如何。

(4)软件使用是否方便。

(5)环境开放的程度如何(即对环境、平台的限制，与其他软件连接的限制)。

1.2.2 质量模型

目前，围绕软件质量模型的研究主要分为两个方向：一是根据经验提出软件质量模型，二是给出一种构建软件质量模型的方法。前者主要是建立各种通用的质量模型来描述软件或信息系统，从而对软件的质量进行预测、度量以及评估，其中有人把质量分解成各种因素，有人把质量分成多个层次。但是，过去的这些质量模型总是想要以单个的模型广泛地应用于所有软件和信息系统的开发。很明显，这与质量本身的特征多样性是相背驰的。因此，如何针对具体的软件或信息系统建立合适的软件模型依然是一个开放的课题。后者主要是提供了如何建立质量模型的方法，包括如何建立质量属性之间的联系以及如何分析质量属性等。

迄今为止，研究人员已经建立了多个质量模型来帮助理解、度量和预测软件的质量，这些模型是研究人员根据多年的软件工程实践经验提出来的，是有效地组织质量保证活动的基础。目前，主流的软件质量模型分为两类：层次模型和关系模型。比较著名的层次模型包括 McCall 模型、Boehm 模型和 ISO 9126 质量模型；比较著名的关系模型包括 Perry 模型和 Gillies 模型。

McCall 模型是最早的质量模型之一，属于层次模型的类型。J. A. McCall 等人将其模型分为三层：因素、准则、计量，并对软件质量因素进行了研究，认为软件质量是由正确性、可靠性、效率等构成的函数，而正确性、可靠性、效率等被称为软件质量因素，或软

件质量特征，它表现了系统可见的行为化特征。每一因素又由一些准则来衡量，而准则是跟软件产品和设计相关的质量特征的属性。例如，正确性由可跟踪性、完全性、相容性来判断；每一准则又有一些定量化指标来计量，指标是捕获质量准则属性的度量。McCall 认为软件质量可从两个层次去分析，其上层是外部观察的特性，下层是软件内在的特性。McCall 定义了 11 个软件外部质量特性，称为软件的质量要素，这些质量特性分别是面向软件产品的运行、修正和转移的，如图 1-1 所示，它们是正确性、可靠性、易用性、效率、完整性、可维护性、可测试性、灵活性、互联性、可移植性和复用性。同时，还定义了 23 个软件的内部质量特征，称之为软件的质量属性，它们是完备性、一致性、准确性、容错性、简单性、模块性、通用性、可扩充性、工具性、自描述性、执行效率、存储效率、存取控制、存取审查、可操作性、培训性、通信性、软件系统独立性、机器独立性、通信通用性、数据通用性和简明性，软件的内部质量属性通过外部的质量要素反映出来。然而，实践证明以这种方式获得的结果会有一些问题。例如，本质上并不相同的一些问题有可能会被当成同样的问题来对待，导致通过模型获得的反馈也基本相同。这就使得指标的制定及其定义的结果变得难以评价。

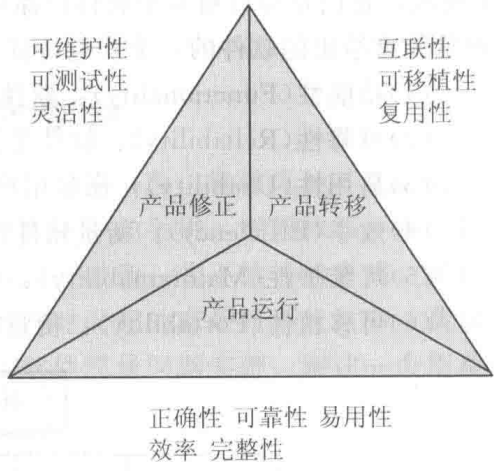

图 1-1　McCall 质量模型

Boehm 模型是由 Boehm 等在 1978 年提出来的质量模型，在表达质量特征的层次性上它与 McCall 模型是非常类似的，如图 1-2 所示。Boehm 提出的概念的成功之处在于它包含了硬件性能的特征，这在 McCall 模型中是没有的。但是，其中与 McCall 模型类似的问题依然存在。

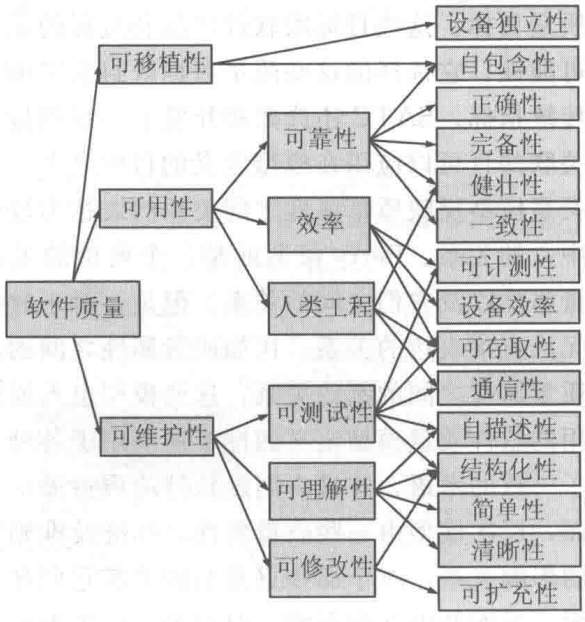

图 1-2　Boehm 质量模型

ISO 9126 质量模型是另一个著名的质量模型，它描述了一个由两部分组成的软件产品