计算机系列教材

软件测试导论

蔡立志 编著

清华大学出版社

计算机系列教材

蔡立志 编著

软件测试导论

清华大学出版社
北 京

内 容 简 介

本书系统地介绍了软件测试的主要技术和方法，全书共分为8章。在讨论黑盒测试时介绍了边界值、等价类、决策表、因果图等方法，而基于控制流的测试则介绍了语句覆盖、判定、条件覆盖、修正判定覆盖和基本路径覆盖；在组合测试方面包括拉丁方阵、正交表等方法，重点讨论了成对组合测试方法；基于有限自动机的测试涵盖了T方法、D方法、W方法、U方法；面向对象语言测试方面包括类属性、对象属性测试，基于对象创建和销毁、装饰器、多态的软件测试；基于UML的软件测试介绍了基于用例图、类图、活动图、序列图、状态图等的测试方法；最后，本书讨论基于Petri网的软件测试、蜕变测试、基于变异的软件测试以及基于故障树的软件测试。

本书结合当前技术发展的特点和工程实践的需求而编写，内容新颖，实操性强，既适合于软件测试和软件质量保障相关的技术人员在从事软件测试时参考，也适合作为软件工程专业的本科生、研究生学习软件测试课程的教科书。

图书在版编目(CIP)数据

软件测试导论/蔡立志编著. --北京：清华大学出版社，2016
计算机系列教材
ISBN 978-7-302-42821-3

Ⅰ. ①软…　Ⅱ. ①蔡…　Ⅲ. ①软件－测试－教材　Ⅳ. ①TP311.5

中国版本图书馆CIP数据核字(2016)第028541号

责任编辑：白立军　薛　阳
封面设计：常雪影
责任校对：时翠兰
责任印制：何　芊

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795954

印 装 者：三河市春园印刷有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　印　　张：24.5　　字　　数：568千字
版　　次：2016年8月第1版　　印　　次：2016年8月第1次印刷
印　　数：1～2000
定　　价：49.50元

产品编号：059067-01

《软件测试导论》 序

每当人们回顾技术进展时，都会一致认同信息技术是发展最快的技术之一，特别是信息技术的渗透性，几乎在各个领域中都可以看到它的身影，从而使我们的世界变得更加精彩。软件作为信息技术的灵魂，更是扮演了极其重要的角色，在现代社会中已经很难想象没有软件会是什么样？所以软件产业正在全球经济中占据越来越重要的地位，而软件质量已成为软件产品健康发展的关键技术。从20世纪发生软件危机以来，软件工程在过程质量管理、软件产品质量管理中都得到了快速的发展。在软件过程质量模型方面，集成的能力成熟度模型CMMI是软件工程发展的一个重要标志，CMMI模型通过提升过程生产质量，从而提升软件产品质量。生产过程的控制依赖于严格的、规范的流程和文档得以实现。但是在实践中，这种重载的过程管理模式，并不适应需求的变化，所带来的效率负担和产品质量提升的关系并不能令人满意。于是又出现各种敏捷的开发过程模型，以期充分快速地适应需求的变化。在软件产品质量模型方面，也经历GB/T 16260—1996软件质量模型、GB/T 16260—2006软件质量模型、ISO/IEC 25010—2014软件质量模型。质量考察的对象从原来的“软件产品”扩展到“系统与软件工程”。无论是重载的规范模型还是轻量级的敏捷模型，或者哪种软件质量模型，软件测试都在其中发挥了无法替代的作用，也是软件产品发布前的最终检验。

随着软件产业的发展，软件测试的从业人员不断增加，对软件测试知识的学习需求也日益强烈。软件杀虫剂效应使得经典的软件测试方法日益失效，但是软件缺陷并没有因此而消失，无论软件开发技术怎样发展，软件缺陷始终像永远无法驱离的阴霾，潜伏在软件之中，新的开发模式和技术促使新的测试技术的发展。在这个背景下，作者结合该领域研究的最新成果和实践，倾注十余年的测试经验编著了本书。该书具有以下几个方面的特点。

1. 系统性

较为系统地介绍了软件测试的基本理论、基本方法和应用例子，涵盖了组合测试、基于有限状态机、面向对象结构测试、基于UML的测试等。

2. 新颖性

在介绍传统的黑盒测试和控制流测试的基础上，立足于当前国内外的最新研究成果，介绍当前测试领域发展的新趋势、新技术。例如，基于Petri网的测试、蜕变测试等。

3. 实用性

软件测试既是一门理论性较强的学科，更是一门实践性和应用性特别强的技术。作者既注重理论的探索，更注重实践的应用，对于每一个测试方法都给出了具体应用的例子，在条件许可的情况下，给出了运行的 Python 程序。

4. 简明性

对当前流行的软件测试技术做了深入浅出的阐述，但没有涉及自动化测试和系统信息安全测试等，读者就能简洁明了地掌握软件测试技术的核心内容。

在软件测试领域，还有很多值得探索的研究课题，例如，测试的 Oracle 问题、测试充分性问题等。这些课题的解决，尚需几代人的不断努力。2011 年，国务院文件（[国发〔2011〕4 号]）中第十八条指出："鼓励软件企业大力开发软件测试和评价技术，完善相关标准，提升软件研发能力，提高软件质量，加强品牌建设，增强产品竞争力。"将软件测试纳入国家软件产业发展的高度进行鼓励。

该书问世，将有益于读者掌握一门重要的技术，有益于提升我国软件产品的质量，有益于推动软件测试的研究、教学、生产的进一步发展，该书是一本值得推荐的优秀著作。

朱三元

2016 年 4 月

《软件测试导论》 前 言

自从20世纪软件危机以来，软件工程得到了快速发展，日益受到学术界和产业界的重视。国际标准化组织ISO和国际电工委员会IEC第一联合技术委员会第七分技术委员会ISO/IEC JTC1/SC7，即软件和系统工程分技术委员会，专注于软件工程国际标准的研制。为了适应软件技术发展的需求，SC7成立了两个和软件测试密切相关的工作组WG6和WG26。WG26专注于研究软件测试标准化工作，2013年发布了ISO/IEC 29119《软件和系统工程 软件测试》系列标准，从概念与定义、测试流程、测试文档、测试技术4个部分加以阐述。WG6专注于软件质量的研究，2014年发布了ISO/IEC 25051《系统与软件工程 系统与软件质量要求和评价(SQuaRE) 就绪可用软件产品(RUSP)的质量要求和测试细则》标准。中国信息技术标准化委员会软件工程分技术委员会专门成立了软件质量与测试工作组，推进软件测试标准化的各项工作。2011年，国务院学位委员会“关于印发《学位授予和人才培养学科目录(2011年)》的通知”(学位[2011]11号)文件确定软件工程学为一级学科(080835)，标志着软件工程学科进入了一个新的发展阶段，是软件工程学科发展的一个重要的里程碑。在2000年国务院18号文《鼓励软件产业和集成电路产业发展的若干政策》(国发[2000]18号)的背景下，全国成立一批第三方独立的软件测试机构从事第三方软件测试业务，为创建良好的软件产业氛围做出了重要贡献。2005年，在北京成立了全国第三方软件评测机构联盟，截至2015年，成员单位由成立之初的十多家发展到2015年的五十多家，软件测试技术发展需求极其旺盛。

自从1946年出现第一个Bug以来，软件测试作为软件质量保证的重要手段之一，在软件工程领域发挥着越来越重要的作用。软件测试自身也逐渐发展成为相对独立的软件工程过程，包含测试策划、测试设计、测试实现、测试执行、测试分析等阶段。但是随着开发技术的发展，带来了日益明显的软件测试杀虫剂效应。工程技术人员将软件测试所发现的软件缺陷的共性特征加以归纳整理，以组件内在属性的方式出现。这些组件在应用之前都已经单独通过测试与验证，例如边界判定、电话号码、身份证的信息有效性验证。基于这些组件开发的软件产品，利用边界值分析、等价类划分等测试方法已经难以发现有价值的缺陷。另一方面，面向方面编程AOP技术使得核心业务和辅助功能分离成为可能，例如日志记录、参数有效性验证等独立于核心业务进行开发，进一步加剧了软件测试的杀虫剂效应，使得经典的测试技术发现软件缺陷的能力在日益降低。动态语言、函数式编程带来了软件编程全新的思维，Z语言、Petri网、UML等新型建模技术和工具应用日益广泛，都促使产业界和学术界不断探索和发展新的测试技术和方法。作者从事软件测

试十多年，越来越深刻地体会到软件测试的变化。为了适应软件技术发展的趋势，结合这些年来软件测试工作实践，将软件测试的工程实践和科学研究的成果编著成《软件测试导论》这本书，与从业者分享。

书中所有的程序代码都采用 Python 语言编写。Python 语言的简洁性，使得读者能够将注意力集中在测试的核心思想。在阐述测试方法和原理方面，努力避免受程序设计语言特性所影响，但非常遗憾，无法做到完全避免语言的相关性。例如，第 6 章面向对象结构的测试，若抛开特定语言，抽象的测试概念会降低读者的感性认识。尽管在不同程序设计语言中，面向语言特性的实现中存在一定的差异，例如，是否存在接口、是否支持多重继承等，第 6 章还是采用 Python 语言相关的特性来描述。在 8.3 节关于变异测试的内容中，许多变异规则是和静态语言特性相关联的，在 Python 语言中并不适用，为了不失一般性，本书描述了这些变异规则。

受篇幅所限，本书没有阐述包括软件质量保证过程、软件测试成熟度模型、软件自动化测试、软件系统的信息安全测试等内容，也不涉及某些特定对象测试，例如移动 APP 测试、Web 测试。这些内容和本书所讨论的软件测试存在密切的关系，但是这些内容都是独立的体系，都可以单独出版。本书仅涵盖当前流行的软件测试技术。

第 1 章，主要介绍软件测试的历史和发展、测试术语、软件缺陷管理、软件质量模型和测试，并探讨了软件测试的局限性及其分类。

第 2 章，主要介绍边界值、等价类、决策表、因果图等传统的黑盒测试方法。

第 3 章，在介绍图论、控制流图知识的基础上，以 Python 语言为例介绍了基于语句覆盖、判定、条件覆盖、修正判定覆盖和基本路径覆盖的测试方法。

第 4 章，在介绍基于拉丁方阵、正交表测试方法的基础上，重点介绍组合测试的要求及其测试用例的生成方法，讨论了可变强度组合测试以及约束对组合测试的影响。

第 5 章，介绍有限自动机的定义、特性以及故障模型，在此基础上，探讨了有限自动机的测试方法，包括 T 方法、D 方法、W 方法、U 方法。

第 6 章，以 Python 语言为例，介绍面向对象语言测试，包括类属性、对象属性测试、基于对象创建和销毁的测试，以及基于装饰器和多态的软件测试。

第 7 章，重点介绍基于用例图、类图、活动图、序列图、状态图的测试，对于每一类 UML 图都涵盖了概念、覆盖准则和测试用例设计三个部分。

第 8 章，重点介绍基于 Petri 的软件测试、蜕变测试、基于变异的软件测试以及基于故障树的软件测试。

本书第 2 章由郑阳和陆佳文协助撰写，第 7 章由陆佳文协助撰写，陆佳文、张杨、刘振宇等人协助对全文进行校对，感谢他们的真诚付出。

最后，感谢清华大学出版社在本书出版过程中付出的所有努力和帮助。感谢上海计算机软件技术开发中心的同事们，本书的出版离不开他们的共同努力。

由于作者才疏学浅，时间匆忙，书中难免存在疏漏或者不足，恳请读者批评指正。反馈意见请发邮件到 clz@ssc. stn. sh. cn。

蔡立志

2016 年 4 月

FOREWORD

《软件测试导论》目 录

第1章 绪 论

随着软件应用的日益深入和广泛，各个行业都已经离不开软件的支撑，软件质量问题引起的损失也日益增大。本章介绍了软件测试的发展以及历史上著名的软件质量问题。软件缺陷管理在整个软件生存周期中发挥着极其重要的作用。软件缺陷生存周期、软件缺陷的属性是软件缺陷管理的核心内容。软件质量模型经历了多个版本，为软件测试提供了有价值的依据。软件测试模型描述了软件测试和开发过程之间的关系。最后本章分析了软件测试局限性和分类。

1.1 软件测试的历史和发展

1.1.1 软件测试的起源

在软件测试领域，软件的缺陷通常被称为 Bug。Bug 的意思是小虫。为什么将软件缺陷称为 Bug？有一段历史典故。1946 年，Grace Hopper 从职位退休以后，加入了哈佛大学的计算实验室，继续 Mark Ⅱ和 Mark Ⅲ的研究工作。1947 年 9 月 9 日，Mark Ⅱ计算机遇到一个错误。经调查发现这个错误是由于 F 面板(Panel F)上第 70 号继电器中间飞入了一个飞蛾而导致电路信号错误。这个飞蛾被贴在了日志上，日志上写着："First actual case of bug being found."，如图 1-1 所示，这就是软件缺陷的起源。该日志目前保存在史密森尼博物院中。

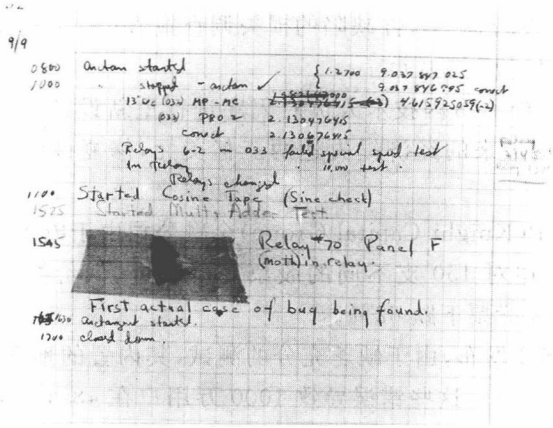

图 1-1 Mark Ⅱ上的 Bug

早期的软件是用机器语言编写的，机器语言是内置在计算机电路中的指令，由 0 和 1 组成。不同的计算机使用不同的机器语言，程序员必须记住每条语言指令及其二进制数

字组合。后来在机器指令的基础上出现了汇编语言，使用助记符表示每条机器语言指令，例如 ADD 表示加，SUB 表示减，MOV 表示移动数据。汇编语言需要大量的记忆指令，并且在表达复杂逻辑方面的能力非常弱，通常程序的功能也非常有限。

在那个时期，软件规模一般都很小、复杂程度低，没有形成成熟的软件开发过程体系，测试的含义比较狭窄。一般将测试等同于“调试”，目的是纠正软件中已经知道的故障，常常由开发人员自己完成这部分工作。

软件缺陷(Defect)是程序本身存在的问题，是程序原来就具有的。“缺陷”一词更能反映事情的本质，因为 Bug 是从外面爬进去的，是外部因素，并非程序本身存在的问题。

1.1.2 软件质量问题

随着软件技术不断发展，软件在各个领域都得到了非常广泛的应用，其作用也越来越重要。软件由于存在缺陷而导致的损失也越来越大，历史上曾经出现过几个著名的软件质量事故案例，涉及航空、航天、金融、工业控制、电子商务等各个领域，每一个质量事故都造成了重大的经济损失。

1996 年 6 月 4 日，欧洲阿丽亚娜 5 型运载火箭的发射系统代码直接重用了 4 型的相应代码，而 4 型的飞行条件和 5 型的飞行条件截然不同。软件引发的问题导致火箭在发射 39s 后偏轨，从而激活了火箭的自我摧毁装置，此次事故造成了 3.7 亿美元经济损失。

1999 年，NASA 在制造其火星气候轨道探测器时，使用的是英制单位，而不是预定的公制单位，由于测试不充分导致该缺陷未被及时发现。该错误造成探测器的推进器无法正常运作，探测器从距离火星表面 130 英尺的高度垂直坠毁。此项工程成本耗费 3.27 亿美元。

2002 年 6 月 28 日，美国国立标准技术研究所(National Institute of Standards and Technology，NIST)发表的有关软件缺陷的损失调查报告指出：据推测，由于软件缺陷而引起的损失额每年高达 595 亿美元。

2006 年，美国国税局 IRS 因技术人员对程序进行重新设计，导致电子诈骗系统不能正常运行。该错误直接带来的经济损失达两三亿美元，修复该错误的成本高达 2100 万美元。

2012 年，美国 KCP(Knight Capital Group)金融公司由于电子交易系统出现故障，交易算法出错，导致该公司对 150 支不同的股票高价购进、低价抛出，直接给公司带来了 4.4 亿美元的损失，当天股票下跌 62%。

2012 年，苹果 iOS 6 发布，由于缺乏充分的测试，其内置的地图服务存在许多地点和定位的错误，如图 1-2 所示。这些错误导致 1000 万用户在 48 个小时内纷纷涌向 Google 地图。

2014 年，由于软件设计上的缺陷，黑客组织“蜻蜓 Dragonfly”对石油管道运营商、发电企业和其他能源工控设备提供商发起攻击。在 18 个月内，全球有 84 个国家的工业控制系统受到了攻击，1018 座发电站感染恶意程序。

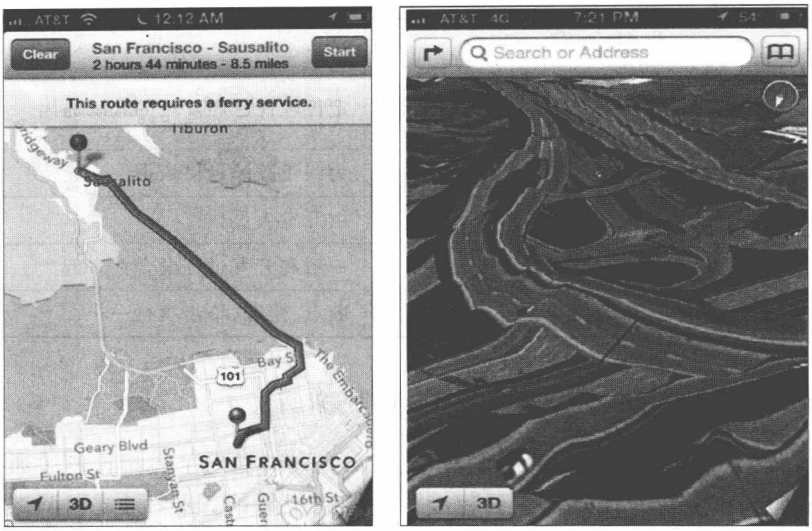

图 1-2 苹果地图错误

1.1.3 软件测试的发展

从 Mark Ⅱ上出现的第一个缺陷(尽管这个缺陷不是人为造成的)开始,软件测试经过近七十年的发展,详细信息见附录 A。在这个过程中,形成 4 种不同的软件测试观点。

1. 软件测试等同于调试活动

这种观点存在于早前的阶段,认为软件测试的目的是定位或者纠正软件中存在的故障。一般情况下,软件项目中没有独立的测试人员,执行软件测试任务的是软件开发人员。软件测试没有形成独立阶段,也不是独立的任务。

但是软件测试和调试还是存在很大区别。调试的基本手段包括在程序中设置断点、观察内容变量的变化、逐步跟踪程序的执行等。尽管在调试时,也会设计一定的输入参数,但是其重点在于是否能够定位已经发现缺陷的原因,而不会考虑软件全面的特性。表 1-1 给出了软件测试和调试之间的比较。

表 1-1 软件调试和测试的比较

	软 件 测 试	调 试
相同点	关心软件参数的输入	
	和软件故障相关联	
	测试发现的缺陷,可以成为调试的依据	
	依赖于软件运行	