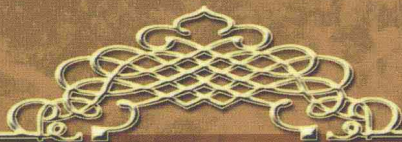




北京市高等教育精品教材立项项目

面向计算机科学与技术专业规范系列教材



软件测试

陈明 编著



Software Testing

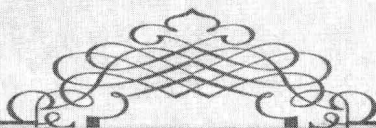


机械工业出版社
China Machine Press



北京市高等教育精品教材立项项目

面向计算机科学与技术专业规范系列教材



软件测试

陈明 编著

Software Testing



机械工业出版社
China Machine Press

本书系统地介绍了软件测试的基本技术及方法，全书分为 11 章，主要包括：软件测试概述、软件测试方法、单元测试、集成测试与确认测试、系统测试与验收测试、系统性能测试、面向对象软件测试、测试的设计与实现、Web 应用测试、软件测试自动化、软件质量保证等。

本书可作为高等学校相关专业软件测试课程的教材，也可作为计算机软件开发人员的参考书。

封底无防伪标均为盗版

版权所有，侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目 (CIP) 数据

软件测试 / 陈明编著. — 北京：机械工业出版社，2011.5

(面向计算机科学与技术专业规范系列教材)

ISBN 978-7-111-33729-4

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

中国版本图书馆 CIP 数据核字 (2011) 第 039941 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：刘立卿

北京京师印务有限公司印刷

2011 年 5 月第 1 版第 1 次印刷

185mm×260mm·12.25 印张

标准书号：ISBN 978-7-111-33729-4

定价：25.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88378991；88361066

购书热线：(010) 68326294；88379649；68995259

投稿热线：(010) 88379604

读者信箱：hzsj@hzbook.com

出版者的话

机械工业出版社华章公司是国内重要的教育出版公司，培生教育集团（拥有 Addison Wesley、Prentice Hall 等品牌）是全球知名的教育出版集团，双方在过去长达十余年的合作中秉承“全球采集内容，服务教育事业”的理念，遴选、移译了国外大量的在计算机科学界享誉盛名的专家名著与名校教材，其中包括 Donald E. Knuth、Alfred V. Aho、Jeffrey D. Ullman、John E. Hopcroft、Dennis Ritchie 等大师名家的经典作品（收录在大理石封面的“计算机科学丛书”中），这些作品对国内计算机教育及科研事业的发展起到了积极的促进作用。

随着国内计算机科学与技术专业学科建设的不断完善、教学研究的蓬勃发展，以及教材改革的逐渐深化，计算机科学与技术专业的优秀课程及教材不仅仅是“引进来”（版权引进），而且需要“走出去”（版权输出）了。

近几年以来，教育部计算机科学与技术专业教学指导分委员会根据我国计算机专业教育的现状以及社会对人才的需求，发布了《高等学校计算机科学与技术专业发展战略研究报告暨专业规范（试行）》（以下简称《规范》）。为配合《规范》的实施推广，同时为落实中央“提高高等教育质量”的最新指导思想，在教育部计算机科学与技术专业教学指导分委员会的指导下，在国内知名高校众多教授的帮助下，我们出版了这套“面向计算机科学与技术专业规范系列教材”。

本套教材的作者在长达数十年的科研和教学经历中积累了大量的知识和经验，也奠定了他们在学术和教学领域的地位，教材的内容体现了他们的教学思想和教学理念，本套教材也是传承他们优秀教学成果的最好载体，是中国版的专家名著和名校教材，相信它们的出版对提高计算机科学与技术专业的教育水平和教学质量能够起到积极的作用。

华章与培生作为专业的出版团队，愿与高等院校的老师共同携手，在这套教材的出版上引进国际先进教材出版经验，在教学配套资源的建设上做出新的尝试，为促进中国计算机科学与技术专业教育事业的发展，为增进中国与世界文化的交流而努力。



华章教育 培生教育集团



序 言

近 20 年里，计算机学科有了很大的发展，人们普遍认为，“计算机科学”这个名字已经难以涵盖该学科的内容，因此，改称其为计算学科（Computing Discipline）。在我国本科教育中，1996 年以前曾经有计算机软件专业和计算机及应用专业，之后被合并为计算机科学与技术专业。2004 年以来，教育部计算机科学与技术专业教学指导分委员会根据我国计算机专业教育和计算学科的现状，为更好地满足社会对计算机专业人才的需求，发布了《高等学校计算机科学与技术专业发展战略研究报告暨专业规范（试行）》（以下简称《规范》），提出在计算机科学与技术专业名称之下，构建计算机科学、计算机工程、软件工程和信息技术四大专业方向。《规范》中四大专业方向的分类，在于鼓励办学单位根据自己的情况设定不同的培养方案，以培养更具针对性和特色的计算机专业人才。

为配合《规范》的实施，落实中央“提高高等教育质量”的精神，我们规划了“面向计算机科学与技术专业规范系列教材”。本系列教材面向全新的计算学科，针对我国高等院校逐步向新的计算机科学与技术专业课程体系过渡的趋势编写，在知识选择、内容组织和教学方法等方面满足《规范》的要求，并与国际接轨。本套教材具有以下几个特点：

（1）**体现《规范》的基本思想，满足其课程要求。**为使教材符合我国高等院校的教学实际，编委会根据《规范》的要求规划本套教材，广泛征集在国内知名高校中从事一线教学和科研工作、经验丰富的优秀教师承担编写任务。

（2）**围绕“提高教育质量”的宗旨开发教材。**为了确保“精品”，本系列教材的出版不走盲目扩大的路子，每本教材的选题都将由编委会集体论证，并由一名编委担任责任编委，最大程度地保证这套教材的编写水准和出版质量。

（3）**教材内容的组织科学、合理，体系得当。**本套教材的编写注重研究学科的新发展和新成果，能够根据不同类型人才培养需求；合理地进行内容取舍、组织和叙述，还精心设计了配套的实验体系和练习体系。

（4）**教材风格鲜明。**本套教材按 4 个专业方向统一规划，分批组织，陆续出版。教材的编写体现了现代教育理念，探讨先进的教学方法。

（5）**开展教材立体化建设。**根据需要配合主教材的建设适时开发实验教材、教师参考书、学生参考书、电子参考资料等教辅资源，为教学实现多方位服务。

我们衷心希望本系列教材能够为我国高等院校计算机科学与技术等专业的教学作出贡献，欢迎广大读者广为选用。

“面向计算机科学与技术专业规范系列教材”编委会

面向计算机科学与技术专业规范系列教材

编委会

主任委员： 蒋宗礼

副主任委员： 王志英 钱乐秋

委员：（以姓氏拼音为序）

陈道蓄	陈 明
傅育熙	何炎祥
黄刘生	贾云得
姜守旭	李仁发
李晓明	刘 辰
马殿富	齐 勇
孙吉贵	孙茂松
吴功宜	吴 跃
谢长生	于 戈
张 钢	周兴社

秘书组：温莉芳 刘立卿 姚 蕾

本书责任编辑： 钱乐秋



前言

软件是一种逻辑产品，它广泛应用于各种系统中。计算机软件已成为一种新的驱动力，为决策、现代工程研究等提供解决问题的基础。

随着软件规模的扩大及复杂度的不断提高，必须进行的软件测试的难度也进一步加大，因此更加凸显了软件测试方法的重要性。软件测试是软件工程学科的重要分支，现已成为软件质量保证的关键技术之一。在软件开发过程中，软件测试是不可缺少的重要环节，软件测试工作直接决定了软件产品的质量。

为了保证软件质量，在软件开发过程中需要频繁进行测试，软件测试存在于软件开发过程的各个阶段。虽然软件测试的种类颇多，但一本软件测试教程中不可能介绍所有的测试方法，本书主要阐述基本的测试技术和方法。

本教程以一定的篇幅介绍了系统性能测试技术。性能测试不仅是系统性能测试的过程，也是系统性能优化的过程。完成性能测试需要测试人员的深入分析和思考，需要测试自动化技术的配合。软件测试自动化技术不仅可以帮助提高效率，更重要的是已成为系统性能测试不可缺少的技术，其研究方兴未艾，前景远大。

软件测试自动化技术的实现需要测试工具的支持，软件测试工具是支持测试任务实现而使用的计算机程序，软件测试环境是一组相关的软件测试工具的集合，它们集成在一起支持软件测试。软件测试工具与环境是软件测试的重要组成部分，对于提高软件生产率、改进软件质量有越来越大的作用。

软件测试是一项富有挑战性的工作，它需要灵感和智慧，需要有软件工程的观念，需要丰富的想象力和创造力。软件测试是一门实践性很强的课程，在学习中既要了解基本的理论知识，又要掌握必要的技能，也就是说，不仅要掌握其理论原则与方法，更重要的是学会熟练应用。因此，在学习软件测试技术的过程中，要注重技术的应用，通过大量的实践和思考，理解软件测试的思想和观念，并运用测试技术和技巧去解决问题。

本教程在内容选择上注重先进性与系统性，在结构上各章呈模块化，在描述中注重理论与实践的结合。学生通过软件测试的理论学习与实践，可以快速掌握软件测试必需的技术和方法，提高软件测试能力。

本书的编写获北京市教育委员会共建项目专项资助。

由于作者水平有限，书中不足之处在所难免，敬请读者批评指正。

陈明

于北京

第1章 概述(2学时)

本章是全书内容的导引,在介绍了软件测试的基础上阐述了有关软件测试技术的最基本概念与定义。通过本章的学习,应该了解软件测试的发展过程,理解软件失效的机理及软件测试的原则、重要性、复杂性和经济性,掌握软件测试的定义、分类、对象、目的、原则和步骤。

第2章 软件测试方法(3学时)

本章介绍软件测试中的最基本方法,通过本章内容的学习,可以了解人工测试、机器测试和敏捷测试的基本概念,理解静态测试和动态测试的主要内容,掌握黑盒测试和白盒测试的基本方法,为后面内容的学习建立坚实的基础。

第3章 单元测试(2学时)

单元测试的目的是检查每个程序单元能否正确实现详细设计说明书中的模块功能、性能、接口和设计约束等要求,发现模块内部可能存在的各种错误。通过本章内容的学习,能够了解单元测试的目的和环境,理解单元测试的定义和策略,掌握单元测试的内容、测试用例和测试过程。

第4章 集成测试与确认测试(3学时)

集成测试与确认测试是软件测试的重要步骤,通过本章内容的学习,要了解集成测试的重要性、集成测试分析、集成测试模式以及确认测试的标准,理解核心系统先行集成测试、高频集成测试、配置复审,掌握自顶向下集成、自底向上集成、混合式集成、回归测试、集成测试用例、集成测试过程、确认测试步骤与内容。

第5章 系统测试、终止测试与验收测试(3学时)

系统测试的目的是将系统已实现的功能与其设计目标进行比较,发现系统与系统定义不符合或不一致的地方。终止测试是决定什么时候停止测试。验收测试是被测软件完成了系统测试之后,在被测软件发布之前所进行的软件测试活动。通过本章的学习,可以了解系统测试、终止测试和验收测试的作用,理解各测试阶段的终止标准、 α 测试和 β 测试,掌握功能测试和非功能测试、终止测试的标准、验收测试的内容等。

第6章 系统性能测试(3学时)

性能需求是软件的基本需求,所以性能测试已成为备受关注的课题。通过本章内容的学习,了解性能测试目的、测试指标和测试分类,理解性能测试的要素、性能测试计划,掌握性能测试的步骤等。

第7章 面向对象软件测试(4学时)

通过本章内容的学习,可以了解面向对象测试层次、面向对象测试顺序和面向对象测试用例,理解面向对象分析的测试、面向对象设计的测试、面向对象编程的测试,掌握类测试、面向对象集成测试、面向对象系统测试等。

第8章 测试的设计与实现(2学时)

通过本章内容的学习,能够了解设计测试计划的目的、测试方案的制定、测试策略的制

定、测试计划的制定及测试的组织，理解测试结果统计、测试结果分析和测试报告编写，掌握建立测试配置、测试用例设计、创建测试任务、执行测试任务和处理软件问题报告等内容。

第 9 章 Web 应用测试(2 学时)

Web 应用系统是应用广泛的系统，学习 Web 测试有其重要实际意义。通过本章内容的学习，可以了解 Web 测试的目的与计划、基于 J2EE 平台的测试、基于 .NET 的 ACT 测试，理解基于 Web 总体架构设计的测试、Web 应用开发测试、Web 应用运行测试，掌握客户端设计的测试和服务器端设计的测试等内容。

第 10 章 软件测试自动化(2 学时)

通过本章内容的学习，能够了解测试自动化的优点、过程、问题、局限性、设计，理解测试工具的作用、测试工具的分类、典型的软件测试工具和测试工具的选择，掌握测试自动化用例特征、测试自动化用例设计、测试自动化用例生成的优缺点、测试自动化的前处理和后处理等内容。

第 11 章 软件质量与质量保证(2 学时)

通过本章内容的学习，能够了解软件质量的定义、影响软件质量的因素、软件质量保证的概念与策略、软件设计质量的评审内容与程序质量的评审内容、软件质量保证的标准，理解软件质量评价体系与软件质量评价标准、高质量软件的特性与软件质量框架的构成，掌握软件开发质量的定量描述。

目 录

出版者的话

序 言

前 言

教学建议

2.3 人工测试与机器测试 15

2.3.1 软件审查 16

2.3.2 人工测试与机器测试的
比较 17

2.4 黑盒测试 17

2.4.1 黑盒测试的有关概念 17

2.4.2 等价类划分 18

2.4.3 边界值分析 23

2.4.4 错误推测 25

2.4.5 因果图 26

2.5 白盒测试 28

2.5.1 白盒测试的作用 28

2.5.2 程序结构分析 28

2.5.3 逻辑覆盖 29

2.5.4 程序插装 33

2.5.5 符号测试 34

2.5.6 程序变异 34

2.6 白盒测试和黑盒测试的比较 37

2.6.1 白盒测试的特点 37

2.6.2 黑盒测试的特点 37

2.6.3 黑盒测试与白盒测试的
对比 38

2.7 敏捷测试 39

2.7.1 敏捷技术概述 39

2.7.2 敏捷测试的理念 39

2.7.3 敏捷测试的原则 40

2.7.4 敏捷测试的意义 40

小结 40

习题 41

第3章 单元测试 42

3.1 单元测试定义 42

3.2 单元测试的目的 42

3.3 单元测试的环境 43

第1章 概述 1

1.1 软件测试的发展 1

1.2 软件失效机理 2

1.2.1 四个概念及其关系 2

1.2.2 软件错误类型及出现的
原因 3

1.2.3 软件缺陷的主要特征 4

1.3 软件测试的定义 4

1.4 软件测试的分类 5

1.4.1 基于开发阶段的划分 5

1.4.2 基于测试实施组织的
划分 5

1.4.3 基于测试技术的划分 6

1.5 软件测试的对象 6

1.6 软件测试的目的 6

1.7 软件测试的原则 7

1.8 软件测试的重要性 8

1.9 软件测试的复杂性 8

1.10 软件测试的经济性 8

1.11 软件测试步骤 9

1.11.1 结构化软件测试步骤 9

1.11.2 面向对象软件测试
步骤 10

小结 10

习题 11

第2章 软件测试方法 12

2.1 静态分析 12

2.2 动态测试 14

3.4	单元测试的内容	43	5.2.1	终止测试的标准	68
3.5	单元测试用例	44	5.2.2	各个测试阶段的终止标准	69
3.6	单元测试的策略	45	5.3	验收测试	69
3.7	单元测试的过程	45	5.3.1	验收测试的内容	70
3.7.1	单元测试的三个阶段	46	5.3.2	α 测试	71
3.7.2	单元测试的步骤	47	5.3.3	β 测试	71
3.7.3	单元动态测试方法	47	小结		72
3.8	单元测试问题	47	习题		72
小结		48	第6章	系统性能测试	74
习题		48	6.1	性能测试概述	74
第4章	集成测试与确认测试	49	6.1.1	性能测试的目的与指标	74
4.1	概述	49	6.1.2	性能测试的分类	75
4.1.1	集成测试的重要性	50	6.1.3	性能测试的要素	76
4.1.2	集成测试分析	50	6.2	性能测试的步骤	78
4.1.3	集成测试模式	51	6.2.1	收集需求	78
4.2	自顶向下集成测试	51	6.2.2	设计性能测试用例	78
4.3	自底向上集成测试	52	6.2.3	性能测试用例的特点	78
4.4	混合式集成测试	52	6.2.4	执行性能测试用例	78
4.5	核心系统先行集成测试	54	6.2.5	分析性能测试结果	79
4.6	高频集成测试	54	6.2.6	性能调节	79
4.7	回归测试	55	6.2.7	基准测试	79
4.8	集成测试用例	56	6.2.8	能力策划	79
4.9	集成测试过程	57	6.3	性能测试计划	80
4.10	确认测试	58	6.4	性能测试的注意事项	80
4.10.1	确认测试的步骤	58	小结		81
4.10.2	确认测试的标准	59	习题		81
4.10.3	确认测试的内容	59	第7章	面向对象软件测试	82
4.10.4	配置复审	59	7.1	面向对象测试基础	82
小结		60	7.1.1	面向对象测试层次	82
习题		60	7.1.2	面向对象测试顺序	82
第5章	系统测试、终止测试与验收测试	62	7.1.3	面向对象测试用例	82
5.1	系统测试	62	7.2	面向对象测试模型	83
5.1.1	功能测试	63	7.2.1	面向对象分析的测试	84
5.1.2	非功能测试	64	7.2.2	面向对象设计的测试	85
5.1.3	系统测试与单元测试、集成测试的区别	68	7.2.3	面向对象编程的测试	86
5.2	终止测试	68	7.3	类测试	87
			7.3.1	类测试概述	88

7.3.2 类测试技术	90	9.3.1 代码测试	126
7.3.3 UML 在类测试中的应用	97	9.3.2 组件测试	126
7.4 面向对象集成测试	100	9.4 Web 应用运行的测试	126
7.5 面向对象系统测试	101	9.4.1 功能测试	127
7.6 面向对象测试与传统测试的 比较	102	9.4.2 性能测试	127
小结	103	9.4.3 用户界面测试	128
习题	103	9.4.4 兼容性测试	129
第 8 章 测试的设计与实现	104	9.4.5 安全性测试	129
8.1 测试计划	104	9.4.6 接口测试	130
8.1.1 设计测试计划的目的	104	9.4.7 易用性测试	131
8.1.2 测试方案的制定	105	9.5 基于 J2EE 平台的测试	131
8.1.3 测试策略的制定	106	9.5.1 J2EE 概述	131
8.1.4 测试计划的制定	106	9.5.2 基于 J2EE 的单元测试 技术	132
8.1.5 测试的组织	108	9.5.3 Servlet 的单元测试	134
8.2 测试设计	111	9.5.4 JSP 单元测试	135
8.2.1 建立测试配置	111	9.5.5 数据库访问层的单元 测试	135
8.2.2 测试用例设计	113	9.6 基于 .NET 的 ACT	135
8.3 测试执行	117	9.6.1 ACT 概述	135
8.3.1 创建测试任务	117	9.6.2 ACT 创建测试	136
8.3.2 执行测试任务	118	9.6.3 ACT 测试举例	138
8.3.3 处理软件问题报告	118	小结	140
8.4 测试总结	119	习题	140
8.4.1 测试结果的统计	119	第 10 章 软件测试自动化	141
8.4.2 测试结果的分析	119	10.1 测试自动化的概念	141
8.4.3 测试报告的编写	120	10.2 测试自动化的优点	142
小结	120	10.3 测试自动化的过程	142
习题	120	10.4 测试自动化的问题	144
第 9 章 Web 应用测试	122	10.5 测试自动化的局限性	144
9.1 Web 测试概述	122	10.6 测试自动化设计	145
9.1.1 Web 系统的结构	122	10.6.1 测试自动化的基本 架构	145
9.1.2 Web 测试的目的与计划	123	10.6.2 测试自动化的方法	146
9.1.3 Web 系统的测试策略	124	10.6.3 测试自动化的层次	148
9.2 Web 应用设计的测试	124	10.7 测试自动化用例	149
9.2.1 总体架构设计的测试	124	10.7.1 测试自动化用例特征	149
9.2.2 客户端设计的测试	125	10.7.2 测试自动化用例设计	149
9.2.3 服务器端设计的测试	125		
9.3 Web 应用开发的测试	126		

10.7.3 测试自动化用例生成的优缺点	151	11.5.1 设计质量的评审内容	168
10.8 测试自动化的前处理和后处理	151	11.5.2 程序质量的评审内容	173
10.9 软件测试工具	153	11.6 软件质量保证的标准	175
10.9.1 测试工具的作用	153	11.7 软件质量评价	176
10.9.2 测试工具的分类	154	11.7.1 软件质量评价体系	176
10.9.3 典型的软件测试工具	155	11.7.2 软件质量评价标准	177
10.9.4 测试工具的选择	161	11.8 软件质量框架	179
小结	162	11.8.1 高质量软件的特性	179
习题	162	11.8.2 软件质量框架的组成	179
第 11 章 软件质量与质量保证	164	11.9 软件开发质量的定量描述	181
11.1 软件质量的定义	164	11.9.1 基本的定量估算	181
11.2 影响软件质量的因素	164	11.9.2 对软件需求的估算	181
11.3 软件质量保证	165	11.9.3 估算验收测试阶段预期发现的缺陷数	182
11.3.1 软件质量保证的概念	165	11.9.4 针对维护活动的度量	182
11.3.2 软件质量保证的策略	166	11.9.5 软件可用性的计算	183
11.3.3 SQA 小组的任务	166	11.9.6 利用植入故障法估计	183
11.4 软件质量保证活动	167	小结	183
11.5 软件评审	168	习题	183
		参考文献	184

概 述

学习要点：

- 软件缺陷
- 软件测试的定义
- 软件测试的对象
- 软件测试的目的

计算机科学技术的飞速发展，促进了软件产品的广泛应用。不论是软件的生产者还是软件的使用者，都在激烈的竞争中求生存、求发展，软件产品的质量至关重要，并已经成为关注的焦点。软件开发者为了占有市场，必须把产品质量作为企业的重要目标之一，进而才可以确保在激烈的竞争中获得胜利。而为了保证软件产品的质量，软件测试成为必不可少的重要过程与手段。

在开发大型软件系统的过程中，面对着极其错综复杂的问题，软件开发者的主观认识不可能完全符合客观事实，而且与工程密切相关的各类人员之间的沟通和配合也不可能完美无缺，因此，在软件生命周期的每个阶段不可避免地会产生错误。尽管力求在每个阶段结束之前通过严格的技术审查，尽可能极早地发现并纠正差错，但是，审查并不能发现所有错误，而且，在纠正这些错误的过程中还可能引入新的错误。如果在软件投入实际运行之前没有发现并纠正它存在的错误，那么这些错误将在运行过程中暴露出来，到时候不仅要为改正这些错误而付出高昂的代价，而且系统很可能产生严重后果。

软件测试在软件生命周期中分为两个阶段。在编写出每个模块的代码之后就对它做测试，称为单元测试，模块的编写者和测试者是同一个人，编码和单元测试在软件生命周期中属于同一个阶段。在这个阶段结束之后，对软件系统还应该进行集成测试、确认测试和系统测试等，这是软件生命周期中的另一个独立的阶段，由专门的测试人员承担这项工作。

大量统计结果表明，软件测试的工作量占软件开发总工作量的 40% 以上，在特殊情况下，例如对关系到人的生命安全的软件要进行的测试所花费的成本，可能相当于软件工程其他开发步骤总成本的 3~5 倍。因此，必须重视软件测试，绝不要以为编写出程序之后软件开发工作就完成了，实际上，几乎还需要完成与开发工作同样多的工作量。

1.1 软件测试的发展

随着社会化生产应运而生的测试技术涉及多方面。在许多应用领域，测试都是保证产品质量的关键。软件测试是软件工程的重要组成部分。

随着计算机的产生与发展，软件开发和软件测试就已相继出现。由于早期的计算机性能比较差，软件的可编程范围也比较狭窄，所以在这一阶段并没有系统的软件测试，更多的是是一种调试性测试，测试的目的主要是为了证实系统的可运行性。

20 世纪 50 年代后期到 20 世纪 60 年代，许多高级语言相继诞生并且得到了广泛的应用，

测试的对象逐渐转入到用高级语言书写的系统。但是,由于受到硬件系统发展瓶颈的限制,软件测试位于次要地位,软件的正确性和可用性主要由编程人员的水平所决定。因此,软件测试理论和方法的发展缓慢。

20世纪70年代以后,随着计算机系统速度的提高,软件在整个系统中变得越来越重要。在这个阶段,软件的规模越来越大,可视化的编程环境、日益完善的软件分析设计方法以及新的软件开发过程模型的出现使得大型软件的开发成为可能;另一方面,由于软件规模和复杂性的迅速增加,软件面临着巨大的危机,软件测试受到重视,测试技术的研究也逐步展开。

20世纪70年代中期,对软件测试技术的研究达到高潮。J. B. Goodenough 和 S. L. Gerhart 首先提出了软件测试的理论,从而把软件测试这一实践性很强的学科提高到理论的高度。1982年6月在美国北卡罗来纳大学召开了首次软件测试的技术会议,这次会议讨论了软件测试问题,是软件测试技术发展中的一个重要里程碑。

随着软件产业的发展,对软件的成本、开发进度和质量都提出了更高的要求,对软件质量的控制已不再是传统意义上的软件测试。传统的测试一般在软件开发后期才介入,然而,大量研究结果表明,设计活动引入的错误占软件开发过程中出现的所有错误的50%~65%。因此,测试就已经不再是一个编码后才进行的活动,而是一个贯穿软件开发整个生命周期的质量控制活动。

目前,在测试理论、测试方法、测试过程和测试工具等方面的研究取得了大量的进展。这不仅使软件的质量有了基本的保证,也使软件测试的工作量占到了软件开发总工作量的40%以上,软件测试日益受到重视。

1.2 软件失效机理

由于软件内部逻辑错综复杂,运行环境动态变化,不同的软件差异很大,所以软件的失效机理具有不同的表现形式。通常将软件失效机理描述为软件错误、软件缺陷、软件故障和软件失效。

1.2.1 四个概念及其关系

1. 软件错误

软件错误是指在软件生命周期内不希望出现和不可接受的人为错误,其结果是导致软件缺陷的产生。软件错误的产生是一种人为过程,相对于软件本身,它是一种外部行为。

错误在整个软件开发周期中很可能扩展,在需求阶段发生的错误,在设计期间有可能被放大,在编写代码时还会进一步扩大。

2. 软件缺陷

软件缺陷是存在于软件(文档、数据、程序)之中的那些不希望出现或不可接受的偏差,例如少一个逗号、多一条语句等。缺陷导致的结果是软件运行某一特定条件时出现软件故障,也称为软件缺陷被激活。

3. 软件故障

软件故障是软件运行过程中出现的一种不可接受或不希望出现的内部状态。例如,当软件处于一种死循环时,就是出现了软件故障,如果无有效措施及时处理,则产生软件失效。软件故障是一种动态行为。

4. 软件失效

软件失效是指软件运行时产生的一种不希望出现和不可接受的外部行为结果。

软件错误是一种人为错误。一个软件错误必定产生一个或多个软件缺陷。当一个软件缺陷被激活时，便产生了软件故障。同一个软件缺陷在不同的条件下被激活，可能产生不同的软件故障。对于软件故障，如果没有容错措施或及时处理，便不可避免地导致软件失效。同一软件故障在不同条件下可能产生不同的软件失效。

1.2.2 软件错误类型及出现的原因

1. 软件错误类型

根据软件错误的性质不同，可以把软件错误分为下述几种类型。

1) 需求错误。软件需求指定得不合理或不正确，需求不完整，需求中含有逻辑错误，需求分析的文档有误等。

2) 功能与性能错误。功能或性能存在错误，或是遗漏了某些功能，或是规定了某些冗余的功能；为用户提供的信息有错，或信息不确切；对异常情况处理有误等。

3) 软件结构错误。程序控制流或控制顺序有误，处理过程有误等。

4) 数据错误。数据定义或数据结构有误，数据存取或数据操作有误等。例如，动态数据与静态数据混淆，参数与控制数据混淆等。

5) 实现和编码错误。编码错误包括语法错误、数据名错误、局部变量与全局变量混淆或者程序逻辑有误等。

6) 集成错误。软件的内部接口、外部接口有误，软件各相关部分在时间配合、数据吞吐量等方面不协调等。

7) 系统结构错误。操作系统调用错误或使用错误、恢复错误、诊断错误、分割及覆盖错误以及引用环境错误等。

8) 测试定义与测试执行错误。测试的错误包括测试方案设计与测试实施的错误、测试文档的问题、测试用例不够充分等。

由于软件结构错误、数据错误和功能与性能错误出现得较普遍，所以这类错误检测特别受到重视。

2. 出现错误的原因

软件出现错误的原因是多方面的，归纳起来主要有如下几点：

1) 交流不够、交流上有误解或者根本没有进行交流，在不清晰应该做什么或不应该做什么的情况下进行了应用开发。

2) 图形用户界面（GUI）、客户/服务器结构、分布式应用、数据通信、超大型关系型数据库以及庞大的系统规模等，使得软件复杂性呈指数增长。

3) 程序设计错误。在软件设计阶段出现的错误，主要包括概要设计、详细设计和编码步骤出现了错误。

4) 需求不断变化。需求变化的后果可能造成系统重新设计、项目日程重新安排、已经完成的工作部分重做甚至完全抛弃等。如果有许多小的改变或者一次大的变化，项目各部分之间已知或未知的关系会相互影响，进而导致出现更多问题，还可能影响项目参与者的积极性。

5) 时间压力。软件项目的日程表很难做到准确，很多时候需要预计和猜测。当最终期限到来之际，由于时间紧迫，容易出现错误。

6) 代码文档不完整。在一些团队中，不鼓励程序员为代码编写文档，也不鼓励程序员将

代码写得清晰和容易理解；相反，他们认为少写文档可以更快地进行编码，无法理解的代码更易于工作的保密。显然，这是一种错误的认识。

7) 软件开发工具。当软件产品的开发依赖于某些工具时，这些工具本身隐藏的问题可能会导致产品的错误。因此，应该选择比较成熟的开发工具，而不是追求最先进的开发工具。

3. 软件错误与缺陷的分布

据统计分析，软件错误的分布大致为：需求分析阶段占 56%，软件设计阶段占 27%，编码阶段占 7%，其他占 10%。

软件开发阶段的早期错误较多，到后期将被放大。所有的错误都将付出代价，包括没有被发现的错误和在开发过程中很晚才发现的错误。没有被发现的错误，在系统中迁移、扩散，最终导致系统失效。甚至很晚才发现的错误会造成返工。

1.2.3 软件缺陷的主要特征

软件错误有多种类型，在一些关键系统中，出现软件错误时，其后果是灾难性的。而在非关键性系统中，出现错误的后果可能并不像前一种情况那样明显，但难以察觉。在通常情况下，利用软件缺陷描述软件错误。软件缺陷的主要特征如下：

- 1) 软件未达到软件产品需求说明书指明的要求；
- 2) 软件出现了软件产品需求说明书中指明不应出现的错误；
- 3) 软件功能超出软件产品需求说明书指明的范围；
- 4) 软件未达到软件产品需求说明书未指明但应达到的要求；
- 5) 软件测试人员认为难以理解、不易使用、运行速度慢或最终用户认为不好。

考虑到设计等方面的因素，软件缺陷还包括软件设计不符合规范，未能在特定的条件下（资金、范围等）达到最佳等。但是，更多的是把软件缺陷看成软件运行时出现的各种问题。

统计结果表明：大多数软件缺陷并非源自编码错误，导致软件缺陷的最大原因是需求分析错误，其次是设计错误，还有编码错误和测试错误等。

1.3 软件测试的定义

软件测试的定义是：软件测试是为了发现错误而执行程序的过程。这个定义明确指出寻找错误是测试的目的。

软件测试是软件工程过程的一个重要阶段，在软件投入运行前，对软件需求分析、设计和编码各阶段产品的最终检查，是为了保证软件开发产品的正确性、完全性和一致性，从而进行检测错误以及修正错误的过程。软件开发的目的是开发出满足用户需求的高质量、高性能的软件产品，而软件测试以检查软件产品内容和功能特性为核心，是软件质量保证的关键步骤，也是成功实现软件开发目标的重要保障。

从用户的角度来看，希望通过软件测试找出软件中隐藏的错误，所以软件测试应该是为了发现错误而执行程序的过程。软件测试应该根据软件开发各阶段的规格说明和程序的内部结构而精心设计测试用例（即输入数据及其预期的输出结果），并利用测试用例去运行程序，以发现程序中隐藏的错误。

软件测试的主要作用如下：

- 1) 测试是执行一个系统或者程序的操作；
- 2) 测试是带着发现问题和错误的意图来分析和执行程序；
- 3) 测试结果可以检验程序的功能和质量；