免费提供电子教案

高等院校规划教材
软件工程系列

软件测试案例教程

周元哲 张庆生 王伟伟 刘 海 编著

机械工业出版社
CHINA MACHINE PRESS

高等院校规划教材 软件工程系列

软件测试案例教程

周元哲　张庆生　王伟伟　刘　海　编著

机械工业出版社

本书较为全面、系统地涵盖了当前软件测试领域的理论和实践知识，反映了当前最新的软件测试理论、标准、技术和工具，展望了软件测试的发展趋势。本书内容主要包括：软件测试概述、软件测试基本知识、黑盒测试、白盒测试、性能测试、软件测试管理和移动终端测试，并将软件测试案例以及当今最新的测试工具与每章内容结合，做到理论与实践相结合。

本书可作为高等院校相关专业软件测试的教材或教学参考书，也可供从事计算机应用开发的各类技术人员参考，或用做全国计算机软件测评师考试、软件技术资格与水平考试的培训资料。

本书配套授课电子课件，需要的教师可登录 www.cmpedu.com 免费注册、审核通过后下载，或联系编辑索取（QQ：2399929378，电话：010-88379753）。

图书在版编目（CIP）数据

软件测试案例教程 / 周元哲编著. —北京：机械工业出版社，2013.6
高等院校规划教材·软件工程系列
ISBN 978-7-111-42474-1

Ⅰ. ①软… Ⅱ. ①周… Ⅲ. ①软件－测试－高等学校－教材
Ⅳ. ①TP311.5

中国版本图书馆 CIP 数据核字（2013）第 098231 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
责任编辑：郝建伟 吴超莉
责任印制：张 楠

北京诚信伟业印刷有限公司印刷

2013 年 6 月·第 1 版第 1 次印刷
184mm×260mm·13.5 印张·334 千字
0001－3000 册
标准书号：ISBN 978-7-111-42474-1
定价：32.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务	网络服务
社 服 务 中 心：（010）88361066	教 材 网：http://www.cmpedu.com
销 售 一 部：（010）68326294	机工官网：http://www.cmpbook.com
销 售 二 部：（010）88379649	机工官博：http://weibo.com/cmp1952
读者购书热线：（010）88379203	**封面无防伪标均为盗版**

出 版 说 明

计算机技术在科学研究、生产制造、文化传媒、社交网络等领域的广泛应用，极大地促进了现代科学技术的发展，加速了社会发展的进程，同时带动了社会对计算机专业应用人才的需求持续升温。高等院校为顺应这一需求变化，纷纷加大了对计算机专业应用型人才的培养力度，并深入开展了教学改革研究。

为了进一步满足高等院校计算机教学的需求，机械工业出版社聘请多所高校的计算机专家、教师及教务部门针对计算机教材建设进行了充分的研讨，达成了许多共识，并由此形成了教材的体系架构与编写原则，策划开发了“高等院校规划教材”。

本套教材具有以下特点：

1）涵盖面广，包括计算机教育的多个学科领域。

2）融合高校先进教学理念，包含计算机领域的核心理论与最新应用技术。

3）符合高等院校计算机及相关专业人才培养目标及课程体系的设置，注重理论与实践相结合。

4）实现教材“立体化”建设，为主干课程配备电子教案、素材和实现实训项目等内容，并及时吸纳新兴课程和特色课程教材。

5）可作为高等院校计算机及相关专业的教材，也可作为从事信息类工作人员的参考书。

对于本套教材的组织出版工作，希望计算机教育界的专家和老师能提出宝贵的意见和建议。衷心感谢广大读者的支持与帮助！

机械工业出版社

前　言

随着软件的规模和复杂性的大幅度提升，如何保证软件质量的可靠性变得日益突出。软件测试是保证软件质量的关键技术之一，也是软件开发过程中的一个重要环节，其理论知识和技术工具都在不断革新。

本书结合作者多年从事软件工程的教学经验，并邀请软件公司有关专家一起编写，注重基本理论和基本技能的教学。学习本书，需要读者掌握一些先行知识，如掌握一门高级语言（例如 Visual Basic 或 Java 语言）、数据库、数据结构以及软件工程的基本理论知识等。

本书面向初中级读者，书中介绍了软件测试的基本理论和当前流行的一些软件测试工具的应用，较为全面地涵盖了当前软件测试领域的专业知识，反映了当前最新的软件测试理论、标准、技术和工具，展望了软件测试的发展趋势。

本书内容主要包括：软件测试概述、软件测试基本知识、黑盒测试、白盒测试、性能测试、软件测试管理和移动终端测试，并以案例——“软件工程管理”课程网站测试为例，讲解了当今最新的测试工具，将理论与实践相结合。

本书由周元哲、王伟伟、刘海、张庆生编写。其中，张庆生参与了移动终端测试章节的编写，上海泽众软件科技有限公司王伟伟软件测试工程师参与了 AutoRunner、TestCenter、PerformanceRunner 和 CodeAnalyzer 等测试软件工具相关内容的编写。刘海编写了 Jira 相关内容。其余章节由周元哲编写。由周元哲负责本书大纲拟订与统稿工作。

在本书编写过程中，西安邮电学院王曙燕、舒新峰、胡滨、孟伟君对本书的编写给予了大力的支持并提出了指导性意见。西安龙旗控股有限公司项目经理张金、西安绿点信息科技有限公司总经理周勇龙、上海泽众软件科技有限公司销售经理钟惠民等对本书的写作大纲、写作风格等提出了很多宝贵的意见。在本书编写过程中我们参阅了大量中外文的专著、教材、论文、报告以及网上资料，由于篇幅所限，未能一一列出。本书得到了陕西省教育厅专项科研计划（编号 12JK0723）的资助。在此，一并向帮助和支持本书编写工作的人员及有关参考文献的作者表示衷心的感谢。

由于作者水平有限，书中难免有不足之处，恳请广大读者批评指正。本书作者的电子信箱是zhouyuanzhe@163.com。

编　者

目　录

第1章　软件测试概述

本章介绍软件的发展历史、软件测试的目的、原则和分类，并对软件测试工具以及自动化测试技术进行了说明，为学习本书后续内容做必要准备。

1.1　软件测试发展历程

测试（Test）一词最早出于古拉丁字，它有“罐”或“容器”的含义。到现在测试一词已经普遍使用，在工业生产和制造业中测试被当做一个常规的生产活动，它常常和产品的质量检验密切相关。在这些行业中测试的含义似乎是明确的，但在计算机软件领域内则不然。比如，什么是程序测试？什么是软件测试？它们之间有什么差别？测试和调试是一回事吗？它们之间又有什么差别？对于这些概念的不同解释可能会涉及测试的目的和方法。为了弄懂这些问题，让我们首先回顾一下软件测试的发展历程。

在20世纪60年代计算机发展早期，程序设计是少数聪明人做的事。在“手工作坊”式的软件开发过程中，程序的编写者和使用者往往是同一个人。由于规模小，复杂程度低，软件开发过程相当无序，也没有什么系统化的方法可依，对软件开发工作更没有进行任何管理，软件测试的含义也比较窄，基本等同于“调试”，目的是纠正软件的故障，常常由软件开发人员自己进行。对测试的投入极少，测试介入也晚，常常是等到形成代码，产品已经基本完成时才进行测试。这种个体化的软件开发环境，使得软件设计往往只是人们头脑中隐含进行的一个模糊过程，除了程序清单之外，根本没有其他文档资料保存下来。随着软件规模的扩大，程序复杂性的增加，维护难度的增加，出现了程序开发质量低下，进度延误，费用剧增等问题，导致了“软件危机”的产生。

软件测试是软件工程的一个重要分支，是解决“软件危机”的重要手段之一，是软件质量保证的重要基础。直到1957年，软件测试工作才开始与软件编程人员的调试行为区分开来，作为一种独立、客观地查找软件缺陷的活动。软件测试作为一门独立的学科开始萌芽。

1972年，测试学科先驱Bill Hetzel博士认为：“测试是目的在于鉴定程序或系统的属性或能力的各种活动，它是软件质量的一种度量”。在他的召集下，北卡罗来纳大学举行了首届软件测试会议，一批关于软件测试的理论专著相继出版。这些专著构成了软件测试学科的最初理论框架，标志着软件测试学科的诞生。其中，J.B. Goodenough和Susan Gerhart在IEEE上发表的《测试数据选择的原理》一文中确定了软件测试是软件的一种研究方向，并且认为测试除了要考虑正确性以外，还应关心程序的效率、健壮性等因素，为程序调试提供更多的信息。

1979年，Glenford Myers在《软件测试艺术》一书中，提出“测试是为发现错误而执行的一个程序或者系统的过程”。

20世纪80年代早期，软件和IT行业进入了大发展阶段，软件趋向大型化、高复杂度，对软件的质量要求也越来越高。一些软件测试的基础理论和实用技术开始形成，软件开发的

方式也逐渐由混乱无序的开发过程过渡到结构化的开发过程，以结构化分析与设计、结构化评审、结构化程序设计以及结构化测试为特征，软件测试性质和内容也随之发生变化，测试不再是一个单纯的发现错误的过程，而是具有软件质量评价的内容。1983 年，Bill Hetzel 在《软件测试完全指南》中指出，测试是以评价一个程序或者系统属性为目标的任何一种活动，是对软件质量的度量。同年，IEEE829 将软件测试定义为："使用人工或自动手段来运行或测定某个软件系统的过程，其目的在于检验它是否满足规定的需求或弄清预期结果与实际结果直接的差别"。这个定义明确指出，软件测试的目的是检验软件系统是否满足需求，软件测试不再是一个一次性的工作，也不只是在开发后期的活动，而是与整个开发流程融合成一体。

20 世纪 90 年代，软件测试工具开始运用。1996 年，测试成熟度模型 TMM、测试支持度模型 TSM 等一系列与软件测试相关的理论提出。到了 2002 年，Rick 和 Stefan 在《系统的软件测试》一书中对软件测试作了进一步描述：测试是为了度量和提高软件的质量，对软件进行工程设计、实施和维护的整个生命周期过程。2003 年，Brett Pettichord 发表文章将软件测试思想划分为 5 个学派。

近 20 年来，随着计算机和软件技术的飞速发展，软件测试技术的研究也取得了很大的突破。许多测试模型（如 V 模型等）的产生，自动化测试涌现了大量的软件测试工具，如功能测试工具、Web 测试工具、性能测试工具、测试管理工具、代码测试工具等。以软件测试服务为主导的软件测试产业正在兴起。

1.2 软件测试的目的

软件测试的主要内容包括软件验证技术（Software Verification Technology）、软件确认技术（Software Validation Technology）和软件测试管理技术（Software Testing Management Technology）3 大部分。软件验证技术和软件确认技术相辅相成，其中软件验证技术也称为验证测试，是指测试人员对在软件开发过程中产生的各类开发文档进行规范性、完整性、一致性和正确性的检查或评审的活动，其测试的对象主要是开发过程中形成的各类文档。软件确认技术也称为确认测试，是指测试人员在软件开发过程中或软件产品交付时，运用人工或自动化测试工具，评估软件部件或软件系统是否满足设计目标和顾客需求的活动，测试的对象主要是开发过程中形成的程序代码或产品交付时递交的软件产品。软件测试管理技术则架构于软件验证技术与软件确认技术之上，主要是从技术管理的角度去探讨如何确保软件测试技术在软件生命周期内得到高效实施，并产生预期的效果。软件验证技术、软件确认技术以及软件测试管理技术这 3 个领域既相互独立又相互融合，共同形成了软件测试学科的基本要义。

软件测试是帮助识别开发完成（中间或最终的版本）的计算机软件（整体或部分）的正确度、完全度和质量的过程，是软件质量保证的重要子域。

Grenford J.Myers 曾对软件测试的目的提出过以下观点：

1）测试是为了证明程序有错，而不是证明程序无错误。

2）一个好的测试用例在于它能发现至今未发现的错误。

3）一个成功的测试是发现了至今未发现的错误的测试。

这种观点指出测试是以查找错误为中心的，不是为了演示软件的正确功能。但只从字面意思去理解，往往会产生这样的误导，认为发现错误是软件测试的唯一目的，查找不出错误

的测试就是没有价值的测试。

软件测试的目的往往包含如下内容：

1）测试不仅是为了找出错误，通过分析错误产生的原因和发展趋势，帮助项目管理者发现当前软件开发过程中的缺陷，以便及时改进。

2）测试分析帮助测试人员设计出有针对性的测试方法，改善测试效率和有效性。

3）没有发现错误的测试也是有价值的，完整的测试是评定软件质量的一种方法。测试的目标就是以最少的时间和人力找出软件中潜在的各种错误和缺陷，证明软件的功能和性能与需求说明相符。此外，实施测试收集到的测试结果数据，能为可靠性分析提供依据。

1.3 软件测试的几种观点

下面给出几种关于软件测试的观点。

（1）软件测试的广义论与狭义论

传统的瀑布模型中，测试是指在编码阶段与维护阶段之间，通过运行程序发现错误。这种意义上的测试是不能在编写代码之前发现系统需求及软件设计中的问题的。如果将需求与设计上的问题遗留到后期，就会造成大量的返工，不但会增加软件开发的成本，而且延长了软件开发的周期等。

为了更早地发现问题，将测试延伸到需求评审、设计审查中，并要在软件生命周期的每一阶段中都应包含测试，尽早发现错误并加以修正。这种将软件测试和质量保证合并起来，延伸后的软件测试，被认为是一种软件测试的广义概念。

（2）软件测试的辨证论

验证软件是以正向思维，针对软件系统的所有功能点，逐个验证其正确性。其代表人物是软件测试领域的先驱 Dr. Bill Hetzel，他的代表论著为《The Complete Guide to Software Testing》。

反向思维方式的目的是发现系统中各种各样的问题。其代表人物是 G.J.Myers。他强调一个成功的测试必须是发现 Bug 的测试，不然就没有价值。

（3）软件测试的风险论

软件测试的风险论认为测试是对软件系统中潜在的各种风险进行评估的活动。对应这种观点，产生基于风险的测试策略，首先评估测试的风险，软件功能存在缺陷的概率有多大？哪些是用户最常用的 20%功能？如果某个功能出问题，其对用户的影响有多大？然后根据风险大小确定测试的优先级。优先级高的测试，优先得到执行。一般来讲，针对用户最常用的 20%功能的测试会得到完全执行，另外用户不经常用的 80%功能的测试可不做或少做。

（4）软件测试的经济论

“一个好的测试用例在于它能发现至今未发现的错误”，体现了软件测试的经济学观点。这是由于在需求阶段修正一个错误的代价是 1，而在设计阶段就是它的 3～6 倍，在编程阶段是它的 10 倍，在内部测试阶段是它的 20～40 倍，在外部测试阶段是它的 30～70 倍，而到了产品发布出去时，这个数字就是 40～1000 倍。修正错误的代价不是简单地随着时间线性增长，而几乎是呈指数级增长的。因此，应该尽快尽早地发现缺陷。

（5）软件测试的标准论

软件测试的标准论认为软件测试为“验证”和“确认”活动构成的整体，即软件测试 = V&V。验证是检验软件是否已正确地实现了产品规格书所定义的系统功能和特性。有效性确认是确认所开发的软件是否满足用户真正需求的活动。

综上所述，软件测试是贯穿整个软件开发生命周期、对软件产品（包括阶段性产品）进行验证和确认的活动过程，其目的是尽快尽早地发现在软件产品中所存在的各种问题。可以从下面几个角度来理解软件测试的定义。

1）从软件测试的目的来理解。测试的目的是发现软件中的错误，是为了证明软件有错，而不是证明软件无错，是在软件投入运行前，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。

2）从软件测试的性质来理解。在软件开发过程中，分析、设计与编码等工作都是“建设性的”，唯独测试是带有“破坏性的”。

3）从软件开发的角度来理解。软件测试以检查软件产品的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。

4）从软件工程的角度来理解。软件测试是软件工程的一部分，是软件工程过程中的重要阶段。

5）从软件质量保证的角度来理解。软件测试是软件质量保障的关键措施之一。

1.4 软件测试的原则

在软件测试过程中，应注意和遵循的原则，可以概括为如下几点：

1. 测试的标准都是建立在用户需求之上的

软件测试的目标就是验证产品的一致性和确认产品是否满足客户的需求，所以测试人员要始终站在用户的角度去看问题、去判断软件缺陷的影响，系统中最严重的错误是那些导致程序无法满足用户需求的缺陷。

2. 测试必须基于“质量第一”的思路去开展各项工作

当时间和质量冲突时，时间要服从质量。质量的理念和文化（如零缺陷的“第一次就把事情做对”）同样是软件测试工作的基础。

3. 事先定义好产品的质量标准

有了质量标准，才能依据测试的结果对产品的质量进行正确的分析和评估，例如进行性能测试前，应定义好与产品性能相关的各种指标。同样，测试用例应确定预期输出结果，如果无法确定测试结果，则无法进行校验。

4. 软件项目一启动，软件测试也就开始了

在代码完成之前，测试人员要参与需求分析、系统或程序设计的审查工作，而且要准备测试计划、测试用例、测试脚本和测试环境，测试计划可以在需求模型一完成就开始，详细的测试用例定义可以在设计模型被确定后开始。应当把“尽早和不断地测试”作为测试人员的座右铭。

5. 穷举测试是不可能的

即使一个特小的程序，其路径排列的数量也非常大，因此，在测试中不可能运行路径的每一种组合，然而充分覆盖程序逻辑，并确保程序设计中使用的所有条件是有可能的。

6. 第三方进行测试会更客观，更有效

程序员应避免测试自己的程序，为达到最佳的效果，应由第三方来进行测试。测试是带有“挑剔性”的行为，心理状态是测试自己程序的障碍。对于需求规格说明的理解产生的错误也很难在程序员本人测试时被发现。

7. 软件测试计划是做好软件测试工作的前提

在进行实际测试之前，应制订良好的、切实可行的测试计划并严格执行，特别要确定测试策略和测试目标。

8. 重视测试用例

根据测试的目的，采用相应的方法去设计测试用例，从而提高测试的效率，更多地发现错误，提高程序的可靠性。除了检查程序是否做了应该做的事，还要看程序是否做了不该做的事；不仅应选用合理的输入数据，对于非法的输入也要设计测试用例进行测试。特别是对作了修改之后的程序进行重新测试时，如不严格执行测试用例，将有可能忽略由于修改错误而引起的大量的新错误。所以，回归测试的关联性也应引起充分的注意，有相当一部分最终发现的错误是在早期测试结果中遗漏的。

9. 对于错误较多的程序段，应进行更深入的测试

一般来说，一段程序中已发现的错误数越多，其中存在的错误概率也就越大。错误集中发生的现象，可能和程序员的编程水平和习惯有很大的关系。

1.5 软件测试的分类

软件的测试的方法很多，不同的出发点会得到不同的测试方法。

1. 按软件开发的阶段来分类

软件测试贯穿软件开发的整个过程，软件测试分为单元测试、集成测试、确认测试、系统测试、验收测试等。

2. 按执行主体划分来分类

按照测试实施的组织来分类，软件测试分为开发方测试、用户测试、第三方测试。

（1）开发方测试

开发方测试通常也叫验收测试或α测试。在软件开发环境中，开发者检测与证实软件的实现是否满足软件设计说明或软件需求说明的要求。

（2）用户测试

用户测试是指在用户的应用环境下，用户检测与核实软件实现是否符合自己预期的要求。用户测试也称为 ß 测试，是把软件有计划地、免费地分发到目标市场，让用户大量使用、评价检查软件。通常情况下，用户测试不是指用户的验收测试，而是指用户的使用性测试，由用户找出软件在应用过程中发现的软件的缺陷与问题，并对使用质量进行评价。

（3）第三方测试

第三方测试是指由第三方测试机构来进行的测试，也称为独立测试。由在技术、管理和财务上与开发方和用户方都相对独立的组织进行软件测试，一般在模拟用户真实应用的环境下，进行软件的确认测试。

3. 按执行状态来分类

按测试执行状态软件测试分为静态测试和动态测试。

（1）静态测试

静态测试也称为静态分析过程，是指计算机不真正运行被测试的程序，而是通过人工对程序和文档进行分析与检查，包括走查、符号执行、需求确认等。静态测试一方面利用计算机作为对被测程序进行特性分析的工具，与人工测试有着根本的区别；另一方面并不真正运行被测程序，这与动态测试方法也不相同。因此，静态测试常称为“分析”，是对被测程序进行特性分析方法的总称。

静态测试如图 1-1 所示。

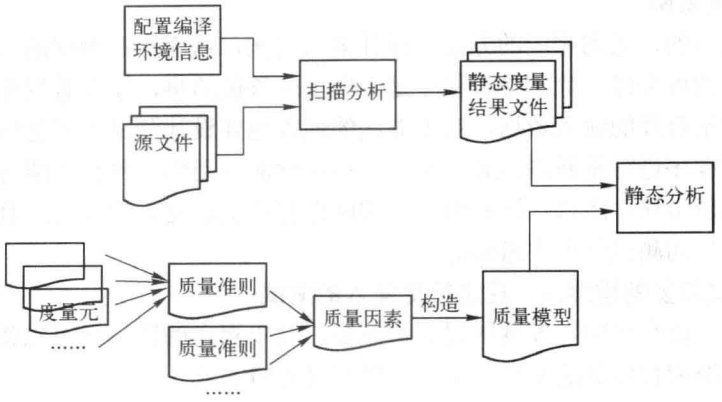

图 1-1　静态测试

其中，针对代码的静态测试包括代码检查、静态结构分析、代码质量度量等。静态测试将在本书第 4 章详细介绍。

（2）动态测试

动态测试是指通过运行被测程序，检查运行结果与预期结果的差异，并分析运行效率和健壮性等性能，这种方法由 3 部分组成：构造测试实例、执行程序、分析程序的输出结果。

4．按测试技术来分类

按照对被测的对象的了解划分，软件测试分为黑盒测试、白盒测试和灰盒测试。

（1）黑盒测试

黑盒测试也称为功能测试或数据驱动测试，它是通过测试来检测已知产品每个功能是否都能正常使用。在测试时，把程序看做一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，测试者在程序接口处进行测试，只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息，并且保持外部信息（如数据库或文件）的完整性。

黑盒测试可发现以下类型的错误：功能错误或遗漏、界面错误、数据结构或外部数据库访问错误、性能错误、初始化和终止错误等。

（2）白盒测试

白盒测试又称结构测试或逻辑驱动测试，与黑盒测试功能正好相反，它是知道产品内部工作过程，检测产品内部动作是否按照规格说明书的规定正常进行，按照程序内部的结构测试程序，检验程序中的每条通路是否都能按预定要求正确工作。白盒测试的主要方法有逻辑驱动、路径测试等，主要用于软件验证。

白盒测试是基于源代码下的测试，需要了解程序的构架、具体需求以及一些编写程序的技巧，能够检查一些程序规范、指针、变量、数组越界等问题。可发现以下类型的错误：变量没有声明、无效引用、数组越界、死循环、函数本身没有析构、参数类型不匹配、调用系统的函数没有考虑到系统的兼容性等。

（3）灰盒测试

灰盒测试介于黑盒测试和白盒测试之间，主要用于测试各个组件之间的逻辑关系是否正确，采用桩驱动，把各个函数按照一定的逻辑串起来，达到在产品还没有界面的情况下的输出结果。

灰盒测试相对白盒测试来说要求相对较低，对测试用例要求也相对较低，用于代码的逻辑测试，验证程序接收和处理参数。灰盒测试的重点在于测试程序的处理能力和健壮性，相对黑盒测试和白盒测试而言，投入的时间相对较少，维护量也较小。

软件测试方法与软件开发过程相关联，单元测试一般应用白盒测试方法，集成测试应用灰盒测试方法，系统测试和确认测试应用黑盒测试方法。

黑盒测试和白盒测试的比较如表 1-1 所示。

表 1-1　黑盒测试和白盒测试比较

项　目	黑盒测试法	白盒测试法
规划方面	功能测试	结构测试
性　质	是一种确认技术，回答“我们在构造一个正确的系统吗？”（Verification）	一种验证技术，回答“我们在正确地构造一个系统吗？”（Validation）
优点	（1）确保从用户角度出发 （2）适用于各阶段测试 （3）从产品功能角度测试 （4）容易入手生成测试数据	（1）针对程序内部特定部分进行覆盖测试 （2）可构成测试数据使特定程序部分得到测试 （3）有一定的充分性度量手段 （4）可获得较多工具支持
缺点	（1）无法测试程序内部特定部分 （2）某些代码得不到测试 （3）如果规格说明有误，则无法发现 （4）不易进行充分性测试	（1）无法测试程序外部特性 （2）不易生成测试数据(通常) （3）无法对未实现规格说明的部分进行测试 （4）工作量大，通常只用于单元测试
应用范围	边界分析法、等价类划分法、决策表测试	语句覆盖、判定覆盖、条件覆盖、路径覆盖等

1.6　软件测试工具

为了提高测试效率，可用软件测试工具来代替一些人工输入，实现人工无法实现的测试功能，发现人工测试中很难发现的缺陷，减少测试执行时间，提高测试效率。但软件测试工也具有如下不足之处：

1）某些测试工具难于学习和使用，创建和修改测试脚本费时费力，相对人工测试而言，不一定节省时间。

2）测试工具只能解决某一方面的问题，应用范围较窄，故应根据测试实际需要确定是否选用和选用什么样的测试工具。

3）某些商业测试工具售价昂贵。

1.6.1　软件测试工具的分类

根据自动化测试工具的分类方法标准不同，软件测试工具有多种分类方法。

1. 按测试工具的市场占有率分类

下面是几大公司的软件测试产品。

（1）MI 公司产品

MI（Mercury Interactive），公司的软件测试工具在市场上占绝对的主导地位。MI 的 4 大产品 LoadRunner、WinRunner、TestDirector、QTP 在全球市场占有率高达 55%。

1）LoadRunner

LoadRunner 属于性能测试工具，用于 C/S 和 B/S 的 Web 系统测试，通过模拟虚拟并发用户数实施压力测试，预测系统行为和性能的负载，对整个软件架构测试分析。LoadRunner 可运行在 Windows、Linux 等多种操作系统，目前流行的版本是 LoadRunner 8.0。

2）WinRunner

Mercury Interactive 公司的 WinRunner 是一种企业级的功能测试工具，用于检测应用程序是否能够达到预期的功能及正常运行。通过自动录制、检测和回放用户的应用操作，WinRunner 能够有效地帮助测试人员对复杂的企业级应用的不同发布版进行测试，提高测试人员的工作效率和质量，确保跨平台的、复杂的企业级应用无故障发布及长期稳定运行。

3）QTP

QTP（Quicktest Professional）是一种自动测试工具，使用 QTP 的目的是用它来执行重复的手动测试，主要是用于回归测试和测试同一软件的新版本。因此，在测试前要考虑好如何对应用程序进行测试，例如要测试哪些功能、操作步骤、输入数据和期望的输出数据等。

4）TestDirector

TestDirector 是基于 Web 集成的测试管理工具，组织和管理整个测试过程。作为业界第一个 Web 的测试管理系统，可以在公司组织内进行全球范围内测试的协调。在一个整体的应用系统中集成了测试需求管理、测试计划、测试日程控制，以及测试执行和错误跟踪等功能，TestDirector 极大地加快了测试过程。

TestDirector 消除组织机构间、地域间的障碍，使得测试人员、开发人员或其他的 IT 人员通过中央数据仓库在不同位置能互通测试信息。TestDirector 将测试过程流水作业：从测试需求管理到测试计划、测试日程安排、测试执行以及出错后跟踪，仅在一个基于浏览器的应用中便可完成。

TestDirector 具有如下功能。

① 需求管理：通过提供一个直观机制将需求和测试用例，测试结果和报告错误联系起来，从而确保完全的测试覆盖率。

② 计划测试：Test Plan Manager 指导测试人员将应用需求转化为具体的测试计划，帮助定义测试应用程序，明确任务和责任。

③ 安排和执行测试：测试计划建立好后，TestDirector 的测试实验室管理为测试日程提供一个基于 Web 的框架，根据测试计划中创立的指标对测试的执行进行监控。

④ 出错管理：TestDirector 将出错管理直接作用于测试的全过程，提供管理系统终端与终端的出错跟踪，发现问题、修改错误，直到检验修改结果。

⑤ 图形化和报表输出：通过图表和报告在测试的任一环节对数据信息进行分析。

（2）IBM Rational 公司产品

从项目设计到实现，Rational 软件交付平台为软件和基于软件系统的开发提供了完整解