

内 容 简 介

本书作为全国计算机技术与软件专业技术资格（水平）考试用书，全面系统地涵盖了软件评测所需的专业知识。全书共19章，对软件评测的基本理论、测试技术、测试技术应用、新技术应用进行系统的讲解，并给出了相关的实践案例。

本书以软件评测相关的标准为基础，同时结合最新测试技术的发展，给出了软件评测的相关理论和实践。通过本书的学习，读者可以掌握软件评测最佳实践，提升软件评测能力。

本书既是软件评测师考试培训的必备学习教材，同时也适用于测试人员、测试经理和软件质量保证的技术人员使用。

本书扉页为防伪页，封面贴有清华大学出版社防伪标签，无标签者不得销售。
版权所有，侵权必究。举报：010-62782989，beiqinquan@tup.tsinghua.edu.cn。

图书在版编目（CIP）数据

软件评测师教程 / 张旻旻, 于秀明主编. —2 版. —北京: 清华大学出版社, 2021.5

全国计算机技术与软件专业技术资格（水平）考试指定用书

ISBN 978-7-302-58120-8

I. ①软… II. ①张… ②于… III. ①软件—测试—资格考试—自学参考资料 IV. ①TP311.55

中国版本图书馆 CIP 数据核字(2021)第 084264 号

责任编辑：杨如林

封面设计：杨玉兰

责任校对：徐俊伟

责任印制：杨 艳

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：010-83470235

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

印 装 者：三河市金元印装有限公司

经 销：全国新华书店

开 本：185mm×230mm 印 张：26.25 防伪页：1 字 数：564 千字

版 次：2005 年 3 月第 1 版 2021 年 6 月第 2 版 印 次：2021 年 6 月第 1 次印刷

定 价：99.00 元

产品编号：091903-01

第 2 版前言

2003 年 10 月 18 日，国家原人事部与信息产业部联合发文（国人部发〔2003〕39 号），在全国计算机技术与软件专业技术资格（水平）考试中增加“软件评测师”一项，为软件测评人员设立了独立的专业技术资格，促进软件测试人才培养，同时推动软件测试行业的发展。

十多年后的今天，涌现了大量新的开发语言、开发模式和应用类型，在任何软件项目的生存周期过程管理中，软件测试仍是保障软件质量的重要手段，面对大量新技术的发展，高素质的软件测试人才短缺的问题仍然存在。随着《新时期促进集成电路产业和软件产业高质量发展的若干政策》（国发〔2020〕8 号）和《特色化示范性软件学院建设指南（试行）》（教高厅函〔2020〕11 号）政策的更新和发布，软件产业的高质量发展和人才培养再一次引起国家的重视。在新时代新政策的指导下，《软件评测师教程》也迎来了第一次改版。

与前一版类似，《软件评测师教程》（第 2 版）仍以软件与系统工程领域的国际标准和国家标准为基础，在保证书籍内容的科学性、准确性、先进性和完整性的基础上，将代表着通用、成熟和最佳实践的标准化成果进行总结，同时结合最新测试技术的发展，详细解读了软件评测的基础理论知识、测试技术，填补相关人员在软件评测领域的知识空缺；本书同时是对软件评测的实践指导，将相关理论知识结合案例进行描述，使得本书成为实用的技术手册；结合新技术的发展，本书还结合新的领域进行了测试技术应用和测试技术提升的探讨。

本书的目的是引导读者通过对基础知识和必要测试技术的学习，结合相关的实践案例，成为一名优秀的软件测评工程师。本书主要包括以下内容。

第一篇 基础理论篇

主要介绍软件测试的基本概念和基础知识，包括软件测试概述、软件测试基础、软件测评相关标准，以及软件测试过程和管理。

第二篇 测试技术篇

主要介绍测试人员设计测试用例时使用的测试技术，包括基于规格说明的测试技术、基于结构的测试技术、基于经验的测试技术，详述了自动化测试中所涉及的测试技术，并从测试实践的角度给出了基于质量特性的测试与评价。

第三篇 测试技术应用篇

结合前两篇的技术内容，首先以基于风险的测试为例，给出了软件测试的设计过程，然后以软件架构的视角，结合案例给出软件测试技术的应用实践，包括分层架构软件测试、事件驱动架构软件测试、微内核架构软件测试、分布式架构软件测试。

第四篇 新技术应用篇

随着软件工程化和信息技术的不断进步，软件测试技术也有了较大的发展，本篇以介绍新技术新场景为目的，给出了移动应用软件、物联网软件系统和大数据系统的测试技术和实践；同时给出了可信软件验证和人工智能对于软件测试技术发展的思考。

本书由张旻旻、于秀明担任主编，丁晓明、王威、李文鹏、孙凤丽、陈耿、张敏、魏培阳担任副主编。全书共四篇十九章。第一篇由张旻旻、丁晓明、李文鹏、张文渊、王威编写，第二篇由李文鹏、孙凤丽、楼莉、杨桂枝、杨隼、魏培阳、陈耿、王威、郭栋编写，第三篇由于秀明、陈耿、魏培阳、刘魁、楼莉、郝琳编写，第四篇由张敏、刘增志、张艾森、苏婷、韩珂编写。中国电子技术标准化研究院的张旻旻、于秀明、李文鹏、李璐对全书进行了统稿、校对和修订工作。

本书在编写过程中，参考了许多相关的书籍、标准和文献资料，并得到了中国电子技术标准化研究院、工业和信息化部教育与考试中心、全国信息技术标准化技术委员会软件与系统工程分技术委员会的各位领导和专家的关怀和指导，国内高校软件学院、测评机构和企业的多位专家也对本书的编写给予了很多帮助，在此一并表示诚挚的感谢！

由于软件测评技术发展较快，日新月异，许多新技术的应用也还需进一步探讨，更离不开实践的反复验证，加之我们的水平有限，书中难免有疏漏欠妥之处，敬请广大读者不吝赐教。

编者
2021年5月

目 录

第一篇 基础理论篇

第 1 章 软件测试概述	2
1.1 软件测试的背景	2
1.2 软件错误相关的典型案例	4
1.3 国内外发展现状及趋势	7
第 2 章 软件测试基础	12
2.1 软件测试的基本概念	12
2.1.1 什么是软件测试	12
2.1.2 验证与确认	13
2.1.3 软件缺陷	14
2.1.4 测试与质量保证	17
2.1.5 测试用例	18
2.1.6 测试策略	19
2.2 软件测试的原则	20
2.3 软件测试模型	22
2.3.1 V 模型	22
2.3.2 W 模型	22
2.3.3 H 模型	23
2.3.4 敏捷测试模型	24
2.4 软件测试分类	24
2.4.1 按工程阶段划分的测试	25
2.4.2 按是否执行代码划分的测试	26
2.4.3 按测试实施主体划分的测试	27
2.4.4 按是否关联代码划分的测试	27
2.4.5 按软件质量特性划分的	

测试	29
2.4.6 按符合性评价要求划分的测试	30
2.4.7 回归测试	31
第 3 章 软件测评相关标准	33
3.1 标准化概述	33
3.1.1 标准化的意义	33
3.1.2 标准的分类	34
3.2 软件质量模型与评价标准	35
3.2.1 软件质量标准的发展	35
3.2.2 软件质量模型和测量	38
3.2.3 软件质量评价	43
3.2.4 就绪可用产品 (RUSP) 的质量要求和评价细则	44
3.3 软件测试标准	45
3.3.1 测试过程标准	46
3.3.2 测试文档标准	46
3.3.3 测试技术标准	49
3.4 软件测试工作量及成本估算相关标准	50
第 4 章 软件测试过程和管理	56
4.1 测试过程模型	56
4.2 组织级测试过程	57
4.2.1 目的	58
4.2.2 输入	59
4.2.3 活动和任务	59
4.2.4 结果	60
4.2.5 信息项	60

4.3	测试管理过程	60	第6章	基于结构的测试技术	115
4.3.1	测试策划过程	61	6.1	静态测试技术	115
4.3.2	测试设计和实现过程	66	6.1.1	代码检查	115
4.3.3	测试环境构建和维护 过程	69	6.1.2	编码规则检查	116
4.3.4	测试执行过程	71	6.1.3	静态分析	117
4.3.5	测试事件报告过程	73	6.2	动态测试技术	120
4.3.6	测试监测和控制过程	75	6.2.1	基于控制流设计用例	120
4.3.7	测试完成过程	78	6.2.2	基于数据流设计用例	129
4.4	静态测试过程	80	6.3	基于结构的测试辅助技术	136
4.4.1	目的	80	6.3.1	词法和语法分析	136
4.4.2	输入	80	6.3.2	程序插桩和驱动技术	137
4.4.3	活动和任务	80	6.4	基于结构测试的综合策略	138
4.4.4	结果	81	6.4.1	测试覆盖准则	139
4.4.5	信息项	81	6.4.2	最小测试用例数计算	140
第二篇 测试技术篇			第7章	自动化测试技术	144
第5章	基于规格说明的测试技术	83	7.1	自动化测试技术概述	144
5.1	测试用例设计方法	83	7.1.1	自动化测试技术的发展和 演进	144
5.1.1	等价类划分法	84	7.1.2	自动化测试的概念	145
5.1.2	分类树法	86	7.1.3	自动化测试的分类	146
5.1.3	边界值法	88	7.1.4	自动化测试的优缺点和 局限	147
5.1.4	语法测试	91	7.1.5	自动化测试系统的通用 架构	149
5.1.5	组合测试	92	7.2	自动化测试的实践策略	151
5.1.6	判定表测试	97	7.3	测试设计的自动化技术	153
5.1.7	因果图法	99	7.3.1	基于模型的测试技术	153
5.1.8	状态转移测试	102	7.3.2	基于搜索的测试技术	157
5.1.9	场景测试	105	7.4	测试执行的自动化技术	158
5.1.10	随机测试	107	7.4.1	测试工具的选择	158
5.2	测试设计方法选择策略	109	7.4.2	自动化测试语言的选择	159
5.3	测试用例的编写	109	7.4.3	测试输入的设计与实现	159
5.3.1	测试设计规格说明	110	7.4.4	测试输出结果的收集和 分析	164
5.3.2	测试用例规格说明	111			
5.3.3	测试规程规格说明	112			
5.3.4	测试用例编写的细节	113			

第 8 章 基于经验的测试技术 ····· 166	
8.1 错误猜测法 ····· 166	9.4.1 成熟性 ····· 194
8.1.1 基本概念 ····· 166	9.4.2 可用性 ····· 195
8.1.2 软件错误类型 ····· 166	9.4.3 容错性 ····· 195
8.1.3 估算错误数量的方法 ····· 167	9.4.4 易恢复性 ····· 196
8.2 探索性测试 ····· 168	9.5 信息安全性测试 ····· 197
8.2.1 基本概念 ····· 168	9.5.1 保密性 ····· 197
8.2.2 探索性测试风格 ····· 169	9.5.2 完整性 ····· 198
8.2.3 探索性测试的相关方法 ····· 170	9.5.3 抗抵赖性 ····· 198
8.2.4 探索性测试的优势与 局限 ····· 170	9.5.4 可核查性 ····· 198
8.3 基于检查表的测试 ····· 171	9.5.5 真实性 ····· 199
8.3.1 基本概念 ····· 171	9.6 维护性测试 ····· 199
8.3.2 基于代码检查表的测试 ····· 171	9.6.1 模块化 ····· 199
8.3.3 基于文档检查表的测试 ····· 172	9.6.2 可重用性 ····· 200
第 9 章 基于质量特性的测试与评价 174	9.6.3 易分析性 ····· 200
9.1 功能性测试 ····· 174	9.6.4 易修改性 ····· 201
9.1.1 完备性 ····· 174	9.6.5 易测试性 ····· 202
9.1.2 正确性 ····· 175	9.7 兼容性测试 ····· 202
9.1.3 适合性 ····· 175	9.7.1 共存性 ····· 203
9.1.4 功能性测试案例 ····· 176	9.7.2 互操作性 ····· 203
9.2 性能效率测试 ····· 183	9.8 可移植性测试 ····· 204
9.2.1 时间特性 ····· 183	9.8.1 适应性 ····· 204
9.2.2 资源利用性 ····· 183	9.8.2 易安装性 ····· 206
9.2.3 容量 ····· 184	9.8.3 易替换性 ····· 207
9.2.4 性能效率测试类型 ····· 185	9.9 依从性测试 ····· 207
9.2.5 性能效率测试案例 ····· 186	9.10 符合性测试 ····· 208
9.3 易用性测试 ····· 188	9.10.1 先决条件 ····· 208
9.3.1 可辨识性 ····· 188	9.10.2 评价活动内容 ····· 209
9.3.2 易学性 ····· 189	9.10.3 评价过程 ····· 209
9.3.3 易操作性 ····· 190	9.10.4 评价报告 ····· 209
9.3.4 用户差错防御性 ····· 191	9.10.5 后续的符合性评价 ····· 210
9.3.5 用户界面舒适性 ····· 192	
9.3.6 易访问性 ····· 193	
9.4 可靠性测试 ····· 194	
	第三篇 测试技术应用篇
	第 10 章 基于风险的测试 ····· 212
	10.1 基于风险的测试概述 ····· 212
	10.1.1 为何要基于风险 ····· 212

第 2 章 软件测试基础

在本章将给出与软件测试和软件质量相关的若干概念和定义，如软件测试、软件缺陷、测试用例、验证与确认、质量保证等等，同时结合国家标准介绍软件异常的分类及对应的缺陷描述，各种经典的软件测试模型，以及从不同的视角和维度讨论软件测试的划分。这些内容是后续章节的基础。

2.1 软件测试的基本概念

2.1.1 什么是软件测试

当今人类的经济活动中，所有的商品在进入流通环节之前，都有相应的产品检验或检测行为，并对检验合格的产品附加特别的标识（如合格证）。这已经成为各个国家的法律条文或贸易各方共同遵守的标准，产品检验是生产者的责任和义务，也是用户对产品建立信心的基础。在 1.1 节中已经指出，针对软件产品的检测，称为软件测试。不过这只能算是对软件测试的一个粗浅说明，下面给出在各个历史时期有代表性的软件测试定义，以确定软件测试的精准含义。

1973 年，Bill Hetzel 给出了软件测试的第一个定义：“软件测试就是为了程序能够按预期设想运行而建立足够的信心”。这个定义强调的是证实程序按预期运行，当软件测试这种技术手段发现程序能够如此时，建立信心的目的也就达到了，当然在测试发现程序不能按预期运行时，就意味着程序有错误，需要排除发现的错误再重新测试。这个定义后来被许多人认为是在为软件“证真”，意思是当测试通过时，“证明”了软件是“对的”。

上述定义受到一些人的质疑，这些人认为，测试本身有局限性，测试通过并不能证明软件是对的，而且测试的目的不应该是去证明软件正确。这种观点的代表人物 Glenford J. Myers 于 1979 年给出了软件测试的一个新定义：“测试是为了发现错误而执行一个程序或者系统的过程”。这个定义强调测试目的是发现错误，因此后来被一些人称为是对软件“证伪”。这种观点的支持者甚至认为不能发现软件错误的测试不是“好的”测试，因此软件测试应当竭尽所能去发现尽可能多的错误。

上面两种定义看似观点相反，实则是从不同角度看待软件测试。他们都希望软件能够正常按预期执行，而要做到这点一定是软件中的错误越少越好，只是那个年代关注的焦点是软件中的错误，没有现代的软件质量意识。实际上，这两个定义至今仍然有一定

2.3 软件测试模型

在软件工程的发展过程中，形成了许多开发过程模型，如瀑布模型、原型模型、基于构件的模型、快速应用开发（RAD）、敏捷过程模型等，软件测试的模型通常是对应着开发模型演变的。本节介绍部分经典的软件测试模型。

2.3.1 V 模型

软件测试的 V 模型对应于开发的瀑布模型。瀑布模型将软件的开发明确地划分为需求分析、概要设计、详细设计、编码和测试等阶段，需要在完成前一阶段的工作后才能进入下一阶段，因此测试成了一个阶段性的工作，是最为典型的 V&V 活动。测试 V 模型如图 2-1 所示。

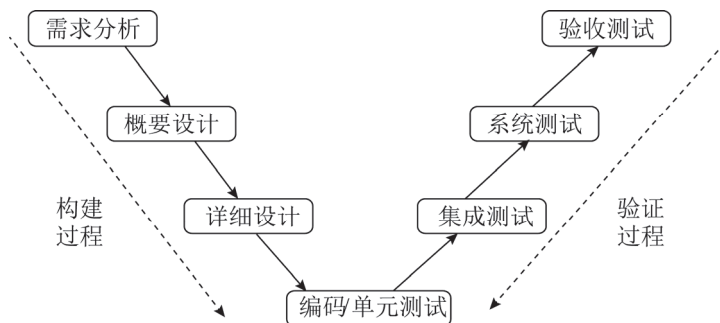


图 2-1 V 模型

在 V 模型中，测试活动对应于瀑布模型的每一个工程阶段，即单元测试对应编码、集成测试对应详细设计、系统测试对应概要设计、验收/交付测试对应需求分析。传统的测试划分就是因此而产生的，这是 V 模型的重要贡献。

2.3.2 W 模型

V 模型存在比较大的局限性。它把测试标定为软件工程的一个阶段性活动，而且是编码结束之后才开始的活动的，启动时间太晚，不符合尽早开始测试的原则。这个模型不仅会让人误解测试在软件工程活动中的作用，而且会造成软件缺陷发现的延迟，越是早期的活动引入的缺陷却越晚被发现，这将带来缺陷修复的巨大代价。

W 模型是对 V 模型的一个重要改进，充分体现了尽早开展测试的原则，并将 V 模型中以发现缺陷为目标上升为保证软件质量为目标。W 模型如图 2-2 所示。

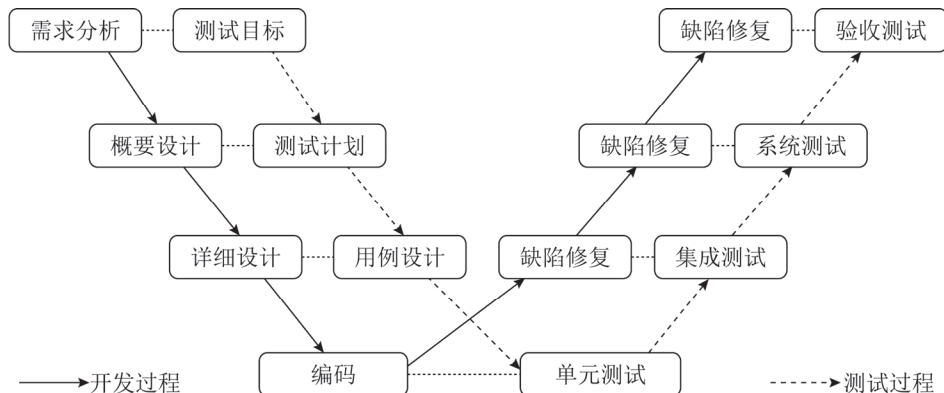


图 2-2 W 模型

W 模型实际上是两个 V 的叠加，一个 V 描述开发过程，另外一个 V 描述测试过程。开发过程的下降边依然是需求分析、概要设计、详细设计和编码，测试过程的上升边也依然是单元测试、集成测试、系统测试以及验收/交付测试，但测试的起始时机不再是编码结束之后，而是从需求分析时开始，且与开发的每一个阶段活动同步进行，通过适时的评审，可以尽早发现和及时处理软件过程中的缺陷，降低缺陷修复的代价，保障产品各生产阶段的质量，从而更充分地保证最终软件的质量。

显然 W 模型优于 V 模型，它体现了更多的软件测试原则。W 模型中测试分布于软件过程的每一个阶段，与开发的同步可以第一时间生成测试的各类文档，从而加快后期测试的进度。同时 W 模型也表明，测试的对象不仅仅只是程序，还包括各个阶段的文档和数据，因此对软件的验证和确认活动事实上也很早就开始了。

2.3.3 H 模型

虽然 W 模型比 V 模型完善了许多，但其局限性仍然存在，它们都高度依赖于开发的瀑布模型，活动具有明显的串行特征。事实上，即使是采用瀑布模型开发的软件，也不一定是如此清晰的串行化，而是存在大量的交叉活动，更不用说采用快速开发或敏捷开发方法的软件了，因此 W 模型不能适用于所有的软件项目。

H 模型进一步改善了 W 模型中的一些问题，其特征如图 2-3 所示。

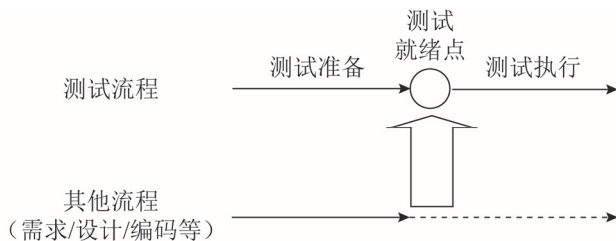


图 2-3 H 模型

H 模型把测试活动从软件开发过程中独立出来，在软件过程的任何一个时间点上，只要测试条件满足即开展测试。图 2-3 中的其他流程可以是软件过程中的各种活动的流程，如需求、设计、编码甚至就是测试流程本身，测试的流程与其他流程是并行的。H 模型也更充分地反映了每一个测试的完整活动，包括测试准备及测试执行。

H 模型比 W 模型更好的地方是能够兼顾测试的效率和灵活性，适合于各种规模及类型的软件项目。

2.3.4 敏捷测试模型

敏捷测试源于敏捷开发。当前敏捷开发是一种比较流行的方法，该方法以用户的需求进化为核心，以迭代、循序渐进的方式进行软件开发，主张简单、拥抱变化、递增、快速反馈等原则。敏捷测试是敏捷开发的组成部分，需要与开发流程良好融合，其特征如图 2-4 所示。

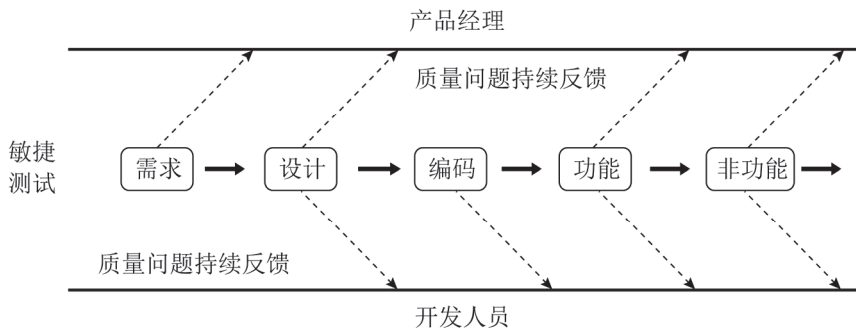


图 2-4 敏捷测试过程

敏捷测试在整个敏捷开发过程中，需要与项目的其他人员甚至用户保持紧密协作，时刻关注需求变化并实施测试，以体现测试的时效性和适应性，这对测试人员有比较高的能力要求。

上面这些是比较经典的软件测试模型。如果关注软件测试的过程改进，还有若干过程改进的模型，如 TMMi (Test Maturity Model integration) 测试成熟度模型集成、TPI (Test Process Improvement) 测试过程改进、CTP (Critical Test Process) 关键测试过程、STEP (Systematic Test & Evaluation Process) 系统化测试和评估过程等模型。限于篇幅，对此不再赘述，感兴趣者请参考其他文献。

2.4 软件测试分类

从不同的视角和维度去看待软件测试，会有不同的分类。比如可以从测试的方法去分类，也可以从开发的阶段性去划分，也可以从测试的对象去划分，还可以从测试的目

第二篇 测试技术篇

测试技术是指导测试人员针对被测软件来设计测试用例所使用的技术。根据测试用例设计所依据的信息来源和测试方式，本篇分为基于规格说明的测试设计技术（第5章）、基于结构的测试技术（第6章）、自动化测试技术（第7章）、基于经验的测试技术（第8章）和基于质量特性的测试与评价（第9章）。

在基于规格说明的测试中，测试的依据包含需求、规格说明、模型或用户需求等，是设计测试用例的主要信息来源，基于规格说明的测试，通常也被称为“黑盒测试”，被测件的内部结构是不可见的，把被测软件视为黑盒子，按照测试依据设计用例输入，以及相应的输出，对软件实现的正确性进行确认。在基于结构的测试中，被测软件的结构（如源代码或模型结构）是设计测试用例的主要信息来源，基于结构的测试，通常也被称为“白盒测试”，对于白盒测试，被测软件的内部结构是透明可见的。自动化测试是把人为驱动的测试行为转化为机器执行的一种过程，即模拟手工测试步骤，通过执行程序语言编制的测试脚本，自动地完成软件的测试设计、单元测试、功能测试、性能测试等全部工作，包括测试过程的自动化和管理工具的自动化。在基于经验的测试中，测试人员的知识和经验是设计测试用例的主要信息来源，一般是测试人员基于以往的项目经验、特定的系统和软件知识或应用领域知识开展，能够发现运用系统化的测试方法不易发现的隐含特征的问题。基于质量特性的测试与评价的测试设计信息来源可以是与软件相关的规格说明、源代码、任务书、数据等，主要从软件质量特性的角度对软件产品进行测试与评价。

尽管每种测试技术都是独立于其他技术而定义的，实际工作中它们相互之间可以结合使用。综合运用这些技术会使测试更加有效。自动化测试是提高测试效率的有效途径和发展方向。

第5章 基于规格说明的测试技术

基于规格说明的测试的依据为软件需求规格说明，以及模型、用户需求等，把程序看作一个黑盒子，不考虑程序内部结构和内部特性，在程序接口进行测试，检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能有效接收输入数据而产生正确的输出信息。

本章介绍了 GB/T 38634.4—2020《系统与软件工程 软件测试 第4部分：测试技术》中提到的基于规格说明的一些常见测试技术，包括等价类划分、分类树、边界值、语法测试、组合测试、判定表测试、因果图、状态转移测试、场景测试和随机测试等，并介绍了多种测试设计方法间的选择策略，最后从测试设计规格说明、测试用例规格说明、测试规程规格说明三个方面介绍了测试用例设计相关文档的编写。

5.1 测试用例设计方法

基于规格说明的测试用于发现软件功能性、性能效率、易用性、可靠性、信息安全性、维护性、兼容性、可移植性等的错误。要保证测试的充分性，关键是测试用例的设计，测试用例设计方法可以用于回答以下问题：

- ①如何测试功能的有效性？
- ②何种类型的输入会产生好的测试用例？
- ③软件是否对特定的输入值尤其敏感？
- ④如何分隔数据类的边界？
- ⑤软件能够承受何种数据率和数据量？
- ⑥特定类型的数据组合会对软件产生何种影响？

相较于基于结构的测试来说，基于规格说明的测试更为注意软件的信息域。它注重于测试软件的功能性需求，使软件测试人员派生出执行程序所有功能需求的输入条件。

理论上说，从输入端要保证查出程序中所有的错误，只能采用穷举的方法，把所有可能的输入都作为测试情况考虑。但实际上测试情况有无穷多个，我们不仅要测试所有合法的输入，而且还要对那些不合法但可能的输入进行测试，所以穷举测试在很多时候都是不可行的。实际的测试工作需要采用各种技术来有针对性地进行测试，通过制定测试案例指导测试的实施，保证软件测试有组织、按步骤，以及有计划地进行。

运用基于规格说明的测试技术，可以导出满足以下标准的测试用例集：

- ①所设计的测试用例能够减少达到合理测试所需的附加测试用例数。

造成一棵树，输入域作为树的根节点，分类作为分支节点，类或者子类作为叶节点。这就是“分类树”这个名字的由来。

一般地，每个分类应该是一个测试条件。通过分解分类所得到的“类”可能会进一步分为“子类”。根据要求的测试覆盖程度，导出的划分和类可能同时包括有效和无效的输入数据。

例如，某组织为其员工到国内主要城市出差而记录了他们的旅行选项，分为目的地、舱位、座位、食物偏好四个分类的单选项，这些单选项的内容如表 5-4 所示。

表 5-4 旅行选项案例说明

单选项	可选值
目的地	北京、上海、广州、深圳、武汉、西安、成都、重庆
舱位	头等舱、公务舱、经济舱
座位	靠过道、靠窗、中间
食物偏好	糖尿病餐、无麸质、蛋奶素食、低脂、低糖、严格素食、标准

从每个分类中选择一个类的任意组合都会出现消息“成功预订”，但其他输入都会出现一个错误的信息“无效输入”。员工不能选择不进餐，因此在本例中没有这个选项。

那么，对于分类树测试而言，测试条件是通过生成每个输入参数的划分和类来确定的。本例构造的分类树可用图 5-1 表示，其中顶层表示测试对象的整个输入域，矩形框表示可以继续划分的分类，叶子结点的文字表示分解到的原子类。

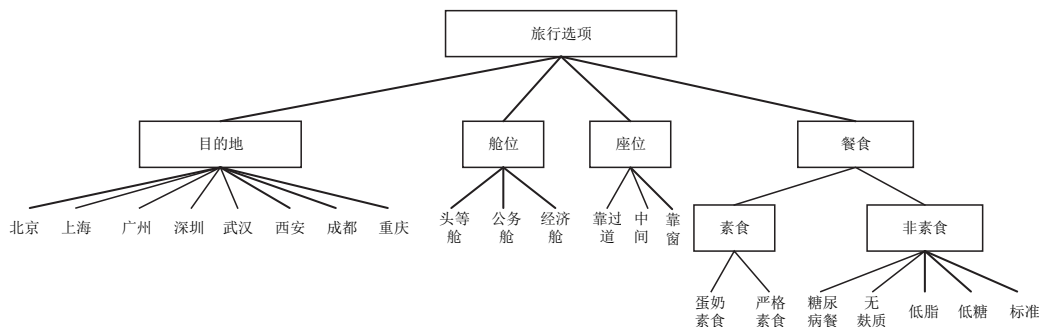


图 5-1 分类树示意图

分类树划分完毕之后，可以通过结合表的方式将各个叶子节点的内容结合而成测试用例，如图 5-2 所示。

分类树的划分过程和等价类划分有点类似，但两者的区别在于，分类树方法中所划分出的类是完全不相交的，而在等价类划分中，它们某些时候也可能会重叠。

考虑到分类树算法在机器学习和数据挖掘中的广泛影响，该测试技术在研究测试用例自动生成时也有广泛应用。

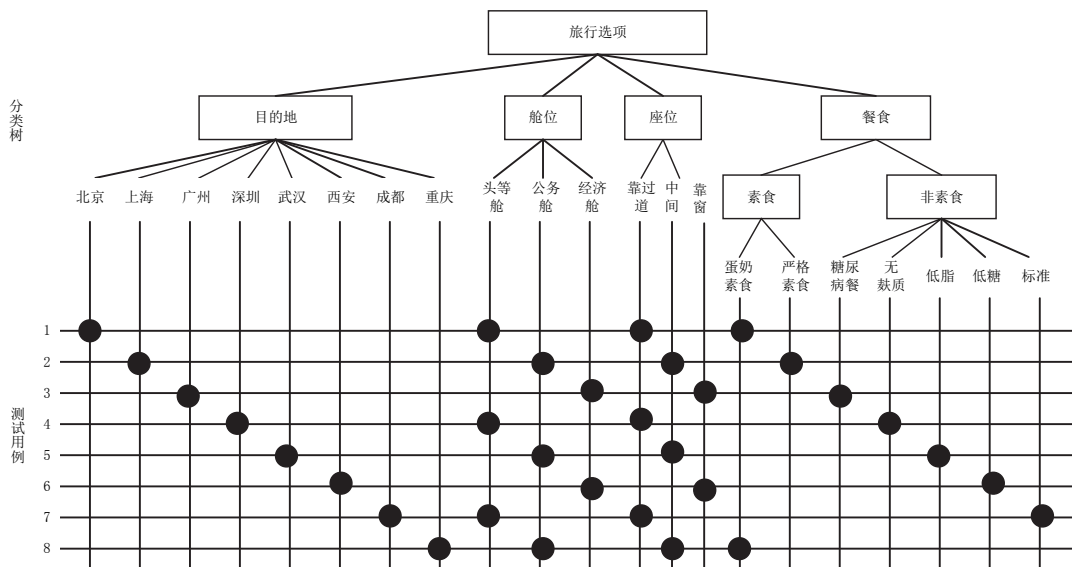


图 5-2 测试用例的确定

5.1.3 边界值法

边界值测试是源于人们长期以来的测试工作经验所提出的一个关键假设：错误更容易发生在输入域的边界或者说极值附近，而非输入域的中间部分。

边界值的选择可以分为二值边界测试和三值边界测试。对于二值边界测试，应为每个边界选择两个输入，这些输入对应于边界上的值和等价划分边界外的增量距离；对于三值边界测试，应为每个边界选择三个输入，这些输入对应于边界上的值和等价划分边界的每一侧的增量距离。增量距离应定义为对应的数据类型的最小有效值。

二值边界测试在大多数情况下是充分的；但是，在某些情况下可能需要进行三值边界测试（例如，测试人员和开发人员在确定被测软件中变量的边界没有发生错误时的严格测试）。

1. 二值基本边界值分析

边界值测试的另一个关键假设是认为：失效极少是由两个（或多个）缺陷的同时发生引起的，在可靠性理论上叫作“单缺陷”假设，这种依据“单缺陷”假设的边界值测试称为基本边界值分析。

在边界值测试时，我们通常使用二值边界，再辅助以一个正常值来设计输入变量的值。

如图 5-3 所示，对于只有 x 和 y 两个输入变量的软件，其输入域在二维坐标系中就是阴影所标示出来的部分。采用基本边界值分析得到的测试用例就是黑点所在位置，一共九个测试用例。

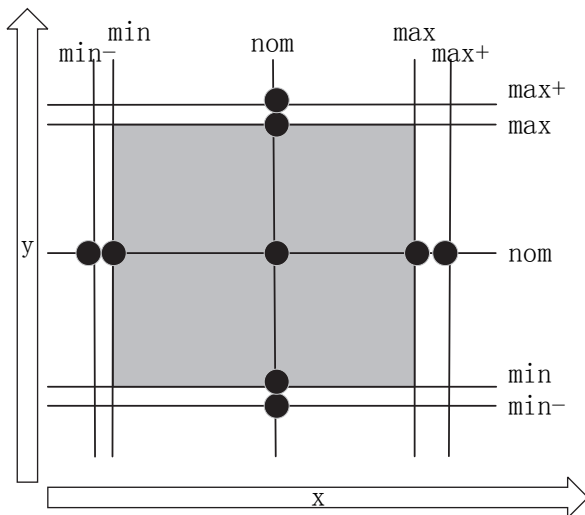


图 5-3 两输入变量的基本边界值分析

如果有一个 n 变量的软件输入域，使其中一个变量取略小于最小值、最小值、正常值、最大值、略大于最大值这样五种选择，其余的所有变量取正常值。如此对每个变量都重复进行之后，该 n 变量软件输入域的边界值分析会产生 $4n+1$ 个测试用例。

2. 三值基本边界值分析

如图 5-4 左图所示，对于只有 x 和 y 两个输入变量的软件，使每个变量取略小于最小值、最小值、略大于最小值、正常值、略小于最大值、最大值、略大于最大值这样七种选择，其余的所有变量取正常值。按照三值基本边界值分析所得到的测试用例就是黑点所在位置。

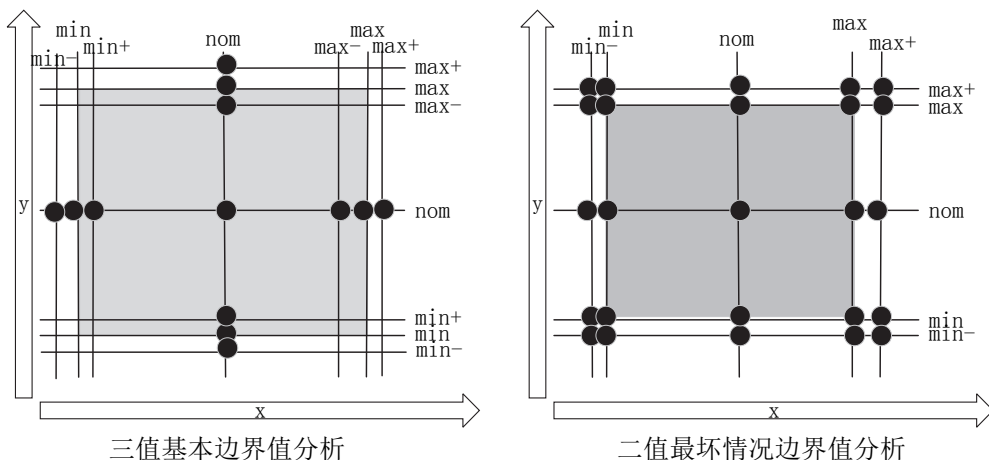


图 5-4 两输入变量的两种不同边界值分析

对于一个 n 变量输入域，三值基本边界值分析将会产生 $6n+1$ 个测试用例。