



普通高等教育

软件工程

“十二五”规划教材

12th Five-Year Plan Textbooks
of Software Engineering

软件测试

(慕课版)

郑炜 刘文兴 杨喜兵 王文鹏 吴潇雪 © 主编

*Software
Testing*



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

高等教育

工程

“十二五”规划教材

12th Five-Year Plan Textbooks
of Software Engineering

软件测试

(慕课版)

郑炜 刘文兴 杨喜兵 王文鹏 吴潇雪 © 主编

*Software
Testing*

人民邮电出版社

121111

图书在版编目(CIP)数据

软件测试：慕课版 / 郑炜等主编. -- 北京：人民邮电出版社，2017.8
普通高等教育软件工程“十二五”规划教材
ISBN 978-7-115-46442-2

I. ①软… II. ①郑… 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(2017)第168027号

内 容 提 要

本书系统介绍了软件测试的基本理论、工具的使用，以及前沿的软件测试理论和技术。

全书共分为11章，内容包括：软件测试的基础、软件测试的策略、黑盒测试与白盒测试、软件测试过程管理、软件测试的度量方法、系统测试技术、软件测试工具及其应用、第三方测试、公有云测试质量评估与退出方法、企业测试实践、CMMI与软件测试、基于搜索的软件测试。本书所有知识都以结合具体实例的形式进行介绍，力求详略得当，使读者可以快速理解软件测试的方法。章后附有习题，供读者实践练习。同时各章节主要内容配备了以二维码为载体的微课，并在学堂在线(<http://www.xuetangx.com>)平台上提供了在线慕课。

-
- ◆ 主 编 郑 炜 刘文兴 杨喜兵 王文鹏 吴潇雪
责任编辑 刘 博
责任印制 陈 彝
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京隆昌伟业印刷有限公司印刷
 - ◆ 开本：787×1092 1/16
印张：19 2017年8月第1版
字数：438千字 2017年8月北京第1次印刷

定价：49.80元

读者服务热线：(010)81055256 印装质量热线：(010)81055316

反盗版热线：(010)81055315

广告经营许可证：京东工商广登字20170147号

软件测试是保证软件质量的重要手段和方法，是软件工程化的重要环节，在整个软件的生命周期中占有非常重要的地位，它对于软件产品质量与生产率的提高起着举足轻重的作用。

软件测试是高等学校计算机科学与技术学科各专业的一门主干课程，与时俱进是软件测试专业教材不断发展的需要。近年来，随着云计算、大数据等技术不断发展，软件测试技术的发展也是日新月异，因此，教学内容也需要同步不断更新。与此同时，传统的教学模式也初现不足，而MOOC课程正在受到全世界广泛的关注。《软件测试（慕课版）》始于软件测试基础理论和核心技术，为慕课版教材，各章节主要内容配备了以二维码为载体的微课，并在学堂在线（<http://www.xuetangx.com>）平台上提供了在线慕课。

《软件测试（慕课版）》的结构安排上力求由浅入深、简明扼要，遵循由基础理论到发展前沿的原则。全书分为3个部分，第一部分（第1章至第6章）强调基础：讲述了与软件测试相关的各种知识，包括软件测试基础、软件测试策略、软件测试的目的和原则、软件测试的各种方法论、软件测试的具体过程，以及软件测试的度量方法。这部分内容是本课程的核心知识。当然，书中也提供了大量的实用经典的习题供大家课下练习。第二部分（第7章至第9章）注重实践：具体讲述业界常用的软件测试技术，以及流行的软件测试工具的使用。其中包括软件自动化测试、兼容性测试、Web测试、系统测试、第三方测试。测试工具包括LoadRunner、JUnit、UFT、Bugzilla，以及appium。

其中, appium项目为国家软件测试大赛中终端测试大赛的练习。第三部分(第10章至第11章)着眼提高:具体讲述软件测试前沿的理论和技術。主要包括云测试、CMMI和软件测试、公有云测试质量评估和可靠性指标,以及基于搜索的软件测试。其中的项目实例大多来源于华为、中兴的项目实践。

本书基本涵盖软件测试各个方面的知识,涉及从测试设计到测试用例、从测试执行到测试管理、从测试基本理论到测试的实用技术,以及测试工具的具体介绍和使用、各种常用测试用例的设计方法,以及测试工具在实践项目的使用。在平台上还提供了对应的习题答案和项目实践的资源 and 源码资源。

本书的完成得到多位工作人员的支持和帮助,刘文兴、杨喜兵、王文鹏、黄月明、冯晨、蔺军、曹石超、蒲敏超、王超、鲍建安、陈杰参与了编写和校稿工作,在此表示衷心感谢!另外,本书编写中借鉴了国内外一些学者的优秀研究成果,在此向他们表示诚挚的感谢!

本书的编写过程中,还得到了中国兵器工业第203研究所蒲海峰高级工程师、中国华为西安研究所云计算测试专家委员会主任刘忠主任、西北工业大学计算机学院李宁副教授的大力支持,在此一并致以衷心的感谢。

由于编者水平有限,本书难免存在疏漏,敬请读者批评指正。

郑炜

2017年5月

1 第1章 软件测试基础

1

- 1.1 软件测试的基本概念····· 1
 - 1.1.1 软件测试是什么····· 1
 - 1.1.2 软件测试的目的····· 2
 - 1.1.3 软件测试与软件质量保证····· 2
- 1.2 软件测试的分类····· 4
- 1.3 软件缺陷管理····· 6
 - 1.3.1 软件缺陷的概念····· 6
 - 1.3.2 软件缺陷的属性····· 7
 - 1.3.3 缺陷生命周期····· 9
 - 1.3.4 常见的缺陷管理工具····· 10
- 1.4 软件质量与软件测试相关特性····· 11
 - 1.4.1 软件质量模型····· 11
 - 1.4.2 测试的复杂性和经济性····· 13
- 1.5 软件测试充分性和测试停止准则····· 16
 - 1.5.1 软件的测试充分性问题····· 16
 - 1.5.2 软件测试原则····· 17
 - 1.5.3 测试停止准则····· 20
- 1.6 小结····· 22
- 1.7 习题····· 23

2 第2章 软件测试策略

24

- 2.1 软件开发过程及模型····· 24
 - 2.1.1 软件开发过程····· 24
 - 2.1.2 软件开发过程模型····· 24
- 2.2 软件测试过程····· 28
 - 2.2.1 测试计划和控制····· 28
 - 2.2.2 测试分析和设计····· 30
 - 2.2.3 测试实现和执行····· 30
 - 2.2.4 测试出口准则的评估和报告····· 31
 - 2.2.5 测试活动结束····· 32
- 2.3 软件测试与软件开发的关系····· 32
 - 2.3.1 软件测试在软件开发中的作用····· 32
 - 2.3.2 软件测试与软件开发各阶段的关系····· 33
 - 2.3.3 常见软件测试模型····· 33
- 2.4 黑盒测试和白盒测试····· 37
 - 2.4.1 黑盒测试····· 37
 - 2.4.2 白盒测试····· 38
 - 2.4.3 黑盒测试与白盒测试的比较····· 39
- 2.5 小结····· 40
- 2.6 习题····· 41

- 3.1 测试用例综述 42
 - 3.1.1 测试用例设计原则 42
 - 3.1.2 测试用例设计步骤 44
- 3.2 等价类设计方法 45
 - 3.2.1 等价类划分 46
 - 3.2.2 等价类划分方法 47
 - 3.2.3 等价类划分的测试运用 50
- 3.3 边界值设计方法 55
 - 3.3.1 边界值分析法原理 55
 - 3.3.2 边界值分析原则 56
 - 3.3.3 健壮性分析 57
- 3.3.4 边界值分析法的测试运用 58
- 3.4 因果图设计法 60
 - 3.4.1 因果图原理 60
 - 3.4.2 因果图法应用 61
 - 3.4.3 决策表法 62
- 3.5 正交试验设计方法 65
 - 3.5.1 正交试验设计法原理 65
 - 3.5.2 利用正交试验法设计测试用例 68
- 3.6 小结 69
- 3.7 习题 69

- 4.1 程序控制流图 70
 - 4.1.1 基本块 70
 - 4.1.2 流图的定义与图形表示 71
- 4.2 逻辑覆盖测试 72
 - 4.2.1 测试覆盖率 72
 - 4.2.2 逻辑覆盖 72
 - 4.2.3 测试覆盖准则 74
- 4.3 路径分析与测试 75
- 4.4 数据流测试分析 77
 - 4.4.1 测试充分性基础 77
 - 4.4.2 测试充分性准则的度量 77
 - 4.4.3 测试集充分性的度量 79
- 4.4.4 数据流概念 79
- 4.4.5 基于数据流的测试充分性准则 82
- 4.5 变异测试 83
 - 4.5.1 变异和变体 83
 - 4.5.2 强变异和弱变异 84
 - 4.5.3 用变异技术进行测试评价 85
 - 4.5.4 变异算子 87
 - 4.5.5 变异算子的设计 88
 - 4.5.6 变异测试的基本原则 88
- 4.6 小结 89
- 4.7 习题 89

- | | | | |
|----------------------|-----|------------------------|-----|
| 5.1 软件测试的各个阶段 | 90 | 5.5 测试的执行 | 103 |
| 5.2 测试需求 | 91 | 5.5.1 测试用例的选择 | 103 |
| 5.2.1 测试需求的分类 | 91 | 5.5.2 测试人员分工 | 104 |
| 5.2.2 测试需求的收集 | 92 | 5.5.3 测试环境的搭建 | 104 |
| 5.2.3 测试需求的分析 | 93 | 5.5.4 BVT测试与冒烟测试 | 105 |
| 5.2.4 测试需求的评审 | 93 | 5.5.5 每日构建介绍 | 105 |
| 5.3 测试计划 | 94 | 5.6 软件缺陷分析 | 106 |
| 5.3.1 测试计划的目标 | 94 | 5.6.1 缺陷分析的作用 | 107 |
| 5.3.2 制定测试计划 | 95 | 5.6.2 软件缺陷的分类 | 107 |
| 5.3.3 划分测试优先级 | 96 | 5.6.3 软件缺陷分析方法 | 107 |
| 5.4 测试设计及测试用例 | 97 | 5.6.4 软件缺陷分析的流程 | 109 |
| 5.4.1 测试用例设计原则 | 97 | 5.6.5 缺陷报告 | 110 |
| 5.4.2 测试用例设计方法 | 98 | 5.7 小结 | 110 |
| 5.4.3 测试用例的粒度 | 102 | 5.8 习题 | 111 |
| 5.4.4 测试用例的评审 | 103 | | |

- | | | | |
|-------------------------------|-----|-----------------------|-----|
| 6.1 软件测试度量的目的 | 112 | 6.2.4 bug综合评价模型 | 125 |
| 6.1.1 度量的目的 | 112 | 6.2.5 测试覆盖率统计 | 126 |
| 6.1.2 测试度量的难度 | 114 | 6.3 软件测试常见的度量类型 | 129 |
| 6.1.3 软件测试人员工作质量的
衡量 | 115 | 6.3.1 手工测试度量 | 130 |
| 6.2 软件测试的度量及其应用 | 121 | 6.3.2 性能测试度量 | 135 |
| 6.2.1 度量bug的数量 | 121 | 6.3.3 自动化测试度量 | 138 |
| 6.2.2 软件测试的度量及其应用 | 121 | 6.3.4 通用度量 | 139 |
| 6.2.3 bug的定性评估 | 123 | 6.4 小结 | 140 |
| | | 6.5 习题 | 141 |

- 7.1 软件自动化测试 142
 - 7.1.1 自动化测试的概念 142
 - 7.1.2 自动化测试的优缺点 144
 - 7.1.3 自动化测试工具 145
- 7.2 兼容性测试 147
 - 7.2.1 兼容性测试的概念 147
 - 7.2.2 兼容性测试内容 148
 - 7.2.3 兼容性测试的标准和
规范 151
 - 7.2.4 兼容性测试工具 152
- 7.3 Web测试实践 152
 - 7.3.1 Web应用体系结构 152
 - 7.3.2 Web测试概述 153
 - 7.3.3 Web测试主要类型 153
- 7.4 移动终端测试实践 158
 - 7.4.1 移动终端测试背景 158
 - 7.4.2 移动终端测试要求 158
 - 7.4.3 移动终端测试实例 159
- 7.5 小结 173
- 7.6 习题 173

- 8.1 性能测试工具LoadRunner 175
 - 8.1.1 性能测试简介 175
 - 8.1.2 LoadRunner的主要
功能 177
 - 8.1.3 性能测试的主要术语 179
 - 8.1.4 LoadRunner的安装 180
 - 8.1.5 LoadRunner的脚本
录制 183
- 8.2 单元测试工具JUnit 188
 - 8.2.1 JUnit简介 188
 - 8.2.2 安装与使用 189
 - 8.2.3 JUnit使用原则 194
- 8.2.4 其他特性 195
- 8.3 功能测试工具C++test 197
 - 8.3.1 C++test的安装 199
 - 8.3.2 C++test静态测试 203
 - 8.3.3 RuleWizard 206
 - 8.3.4 C++test动态测试 212
- 8.4 开源缺陷管理工具Bugzilla 220
 - 8.4.1 Bugzilla简介 220
 - 8.4.2 Bugzilla安装说明 221
 - 8.4.3 Bugzilla使用说明 224
- 8.5 小结 226
- 8.6 习题 227

9 第9章 第三方测试

228

- 9.1 基本概念与测试过程 228
 - 9.1.1 第三方测试的应用现状 228
 - 9.1.2 第三方测试的意义和模式 229
 - 9.1.3 第三方测试的相关概念 229
 - 9.1.4 第三方测试的测试过程 230
- 9.2 测试实例实践 232
- 9.3 小结 235
- 9.4 习题 235

10 第10章 公有云测试质量评估与退出方法

236

- 10.1 云测试概念 236
 - 10.1.1 云计算 236
 - 10.1.2 云测试 237
- 10.2 云可靠性度量 238
 - 10.2.1 软件可靠性 239
 - 10.2.2 软件故障分析和诊断 249
- 10.3 安全测试及安全度量 250
 - 10.3.1 安全性测试方法 251
 - 10.3.2 安全测试方法举例（可用于云中） 253
- 10.4 小结 256
- 10.5 习题 256

11 第11章 软件测试的拓展与提高

257

- 11.1 企业测试实践 257
 - 11.1.1 测试计划 257
 - 11.1.2 测试管理 259
 - 11.1.3 企业的测试策略 261
 - 11.1.4 测试人员组织 262
 - 11.1.5 测试小组的职责 263
- 11.2 CMMI和软件测试 264
 - 11.2.1 CMMI简介 264
 - 11.2.2 基于CMMI的软件测试流程 265
- 11.3 基于搜索的软件测试 266
 - 11.3.1 智能搜索算法 266
 - 11.3.2 搜索技术在软件测试中应用 279
- 11.4 小结 294
- 11.5 习题 294

本章作为导引，目的在于让读者熟悉与软件测试相关的概念，对软件测试有整体的框架认识，为后面章节的学习奠定基础。学完这章后，读者就能够在软件测试和软件质量方面提出一些有意义的问题。

1.1 软件测试的基本概念

1.1.1 软件测试是什么

什么是软件测试，这个概念大家很熟悉，但是不少人确实不一定能够说清楚。许多人或许只能给出一个解释，那就是软件测试就是“发现缺陷/bug”。那么软件测试是否只局限于找 bug，这种解释是否过于简单呢？



软件测试绝非简单，不仅资深的测试工程师和测试初学者的理解不一样，而且连资深的测试工程师之间的理解也是不一样的，再甚者不同行业的测试人员也会给出不同的答案。

一般来说，软件测试是伴随着软件的产生而产生的。早期的软件开发中软件规模都很小、复杂程度低，软件开发的过程混乱无序、相当随意，测试的含义比较狭窄，开发人员将测试等同于“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。那时对测试的投入极少，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。到了 20 世纪 80 年代初期，软件和 IT 行业进入了大发展时期，软件趋向大型化、高复杂度，软件的质量越来越重要。这个时候，一些软件测试的基础理论和实用技术开始形成，并且人们开始为软件开发设计了各种流程和管理方法，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。人们还将“质量”的概念融入其中，软件测试定义发生了改变，测试不单纯是一个发现错误的过程，而且将测试作为软件质量保证的主要职能。

早在 1979 年，Glenford J. Myers 提出了他对软件测试的定义：“测试是为发现错误而执行的一个程序或系统的过程。”他认为软件测试的目的包括以下几点。

(1) 测试是程序的执行过程，目的在于发现错误。

(2) 测试是为了证明程序有错, 而不是证明程序无错误。

(3) 一个好的测试用例在于能发现至今未发现的错误。

(4) 一个成功的测试是发现了至今未发现的错误的测试。

1983年, Bill Hetzelt 在《软件测试完全指南》(Complete Guide of Software Testing)一书中指出:“测试是以评价一个程序或者系统属性为目标任何一种活动。测试是对软件质量的度量。”, 由此表明软件测试的目的不仅仅是为了发现软件缺陷与错误, 同时也对软件进行度量和评估, 提高软件的质量。

而在通俗意义层面上讲, 狭义的软件测试仅仅指动态测试, 即测试是执行程序的过程, 通过运行程序来发现程序代码或软件系统中的错误。广义的软件测试不仅是指运行程序或系统而进行测试, 还包括需求/设计/代码等评审活动。

IEEE 给出了两个规范、约束的测试定义:

(1) 在特定的条件下运行系统或构件, 观察或记录结果, 对系统的某个方面做出评价;

(2) 分析某个软件项以发现和现存的, 以及要求的条件之差别 (即错误并评价此软件项的特性)。

有关软件测试的概念其实一直没有定论, 人们所看到的有关软件测试的概念多是从软件测试的目的、作用等方面进行客观的描述。对软件测试理解不同, 就会决定测试所采取的流程和方法, 以及如何开展测试活动。

1.1.2 软件测试的目的

现对软件测试的目的总结为以下 3 点。

(1) 最少的人力、物力、时间找出软件中潜在的各种错误和缺陷, 全面评估和提高软件质量, 及时揭示质量风险, 控制项目风险。

(2) 通过分析测试过程中发现的问题帮助发现开发工作所采用的软件过程的缺陷, 通过对缺陷分析, 获得缺陷模式, 有助于缺陷预防, 以便进行软件过程改进; 同时通过对测试结果的分析整理, 可修正软件开发规则, 并为软件可靠性分析提供相关的依据。

(3) 评价程序或系统的属性, 对软件质量进行度量和评估, 以验证软件的质量满足用户的需求, 为用户选择、接受软件提供有力的依据。

只有在设计阶段就开始测试工作, 坚持在各个环节进行技术评审和验证, 才能尽早发现错误, 以较低的代价修改错误。

测试用例多次重复使用后, 其发现缺陷的能力会逐渐降低。为了克服这种现象, 测试用例需要进行定期评审和修改, 同时需要不断增加新的、不同的测试用例来测试软件或者系统的不同部分, 从而发现更多的潜在错误。

1.1.3 软件测试与软件质量保证

软件质量保证是贯穿软件项目整个生命周期的有计划、有组织的系统活动, 经常针对整个项目质量计划执行情况进行评估、检查和改进, 确保项目质量与计划保持一致。

确保软件项目的过程遵循了对应的标准及规范要求, 且产生了合适的文档和精确反映项目情况的报告, 其目的是通过评价项目质量建立项目达到质量要求的信心。软件质量保证活动主要包括评审项

目过程、审计软件产品，就软件项目是否真正遵循已经制定的计划、标准和规程等，给管理者提供可视性项目和产品可视化的管理报告。

评价、度量和测试在技术内容上有着非常重要的关系。软件测试是获取度量值的一种重要手段。软件度量在 GJB 5236 中的主要规定是：软件质量模型和内部质量度量、外部质量度量，以及使用质量的度量，可用于在确定软件需求时规定软件质量需求或其他用途。

软件质量评价在 GJB 2434A 中针对开发者、需求方和评价者提出了 3 种不同的评价过程框架。在执行软件产品评价时，确立评价需求的质量模型就需要采用 GJB 5236 给出的内部度量、外部度量、使用质量的度量等。

这两个系列标准的关系如图 1-1 所示，从图中可以看出 GJB 2434A 和 GJB 5236 的联系是非常密切的，需要有机结合才能有效完成软件产品的度量和评价工作。其中，度量值的获取主要来自软件测试。可以说，评价依据度量，而度量也依据测试。也可以说，评价指导度量，度量也指导测试。

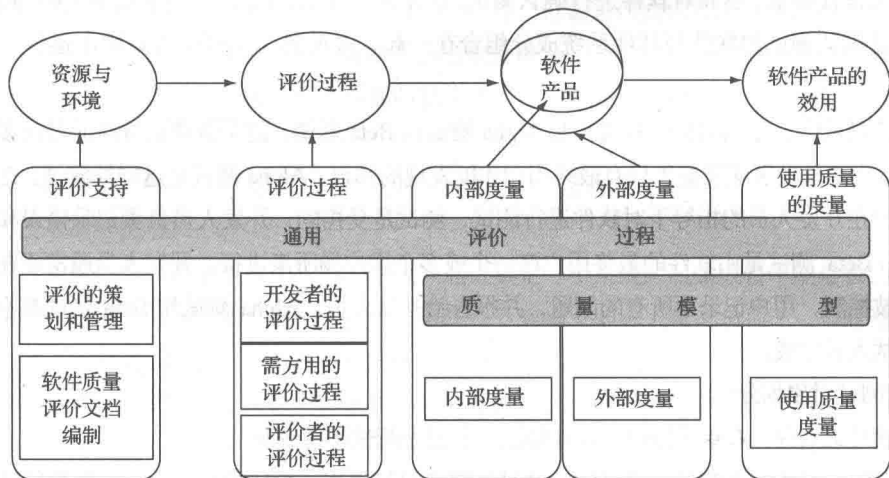


图1-1 评价、度量和测试的关系

软件质量保证与软件测试是否是一回事？有人认为，软件测试就是软件质量保证，也有人认为软件测试只是软件质量保证的一部分。这两种说法其实并不全面。软件质量保证与软件测试二者之间既存在包含又存有交叉的关系。

软件测试能够找出软件缺陷，确保软件产品满足需求。但是测试不是质量保证，二者并不等同。测试可以查找错误并进行修改，从而提高软件产品的质量。软件质量保证测试避免错误以求高质量，并且还有其他方面的措施以保证质量问题。

从共同点的角度看，软件测试和软件质量保证的目的都是尽力确保软件产品满足需求，从而开发出高质量的软件产品。两个流程都是贯穿在整个软件开发生命周期中。正规的软件测试系统主要包括：制定测试计划、测试设计、实施测试、建立和更新测试文档。而软件质量保证的工作主要为：制定软件质量要求、组织正式度量、软件测试管理、对软件的变更进行控制、对软件质量进行度量、对软件质量情况及时记录和报告。软件质量保证的职能是向管理层提供正确的可行信息，从而促进和辅助设计流程的改进。软件质量保证的职能还包括监督测试流程，这样测试工作就可以被客观地审查和评估，

同时也有助于测试流程的改进。

1.2 软件测试的分类

目前，软件测试领域有许多测试名称，这些名称来自不同的分类原则，以下是常见测试名称的分类。

1. 按测试阶段或测试步骤划分

按测试阶段或测试步骤来分，有单元测试、集成测试、确认测试和系统测试。这种划分来源于软件的开发过程，目的是验证软件开发过程各阶段的工作是否符合需求和设计要求。在软件单元完成编码后，首先进行单元测试，验证软件单元是否正确实现了规定的功能和接口等要求；在确认没有问题后，将软件单元组装在一起进行集成测试，验证软件是否满足设计要求；然后对软件进行确认测试，验证软件是否满足软件需求规格说明的各项需求；最后使通过确认测试的软件与其他系统成分组合在一起，并使其在实际运行环境中运行，进行系统测试。

在确认测试中，按照测试的方式又有 Alpha 测试和 Beta 测试。这两种测试针对的是由多用户使用的软件，由用户来发现那些似乎只有最终用户才能发现的错误。Alpha 测试是这样的测试，在开发方的场所，用户在开发人员的指导下对软件进行测试，测试是受控的，开发人员负责记录错误和使用中出现的问题；Beta 测试是由软件的最终用户在一个或多个用户场所来进行，开发人员通常不在现场，整个测试不被控制，用户记录下所有的问题，并报告给开发人员。Alpha 测试和 Beta 测试都不能由开发人员或测试人员完成。

2. 按测试对象划分

按测试对象划分，有单元测试、部件测试、配置项测试和系统测试。

软件配置项是为配置管理而设计的、并且能够满足最终用户功能的软件，在配置管理过程中作为单个实体对待。

软件部件是构成软件配置项或系统的部分之一，在功能上或逻辑上具有一定的独立性，且可以进一步划分为其他部件。

3. 按使用的测试技术划分

按使用的测试技术来分，有静态测试和动态测试，这两种测试分别代表了程序不同的运行状态。动态测试又分为白盒测试和黑盒测试，白盒测试包括逻辑覆盖测试、域测试、程序变异测试、路径测试、符号测试等，黑盒测试包括功能测试、强度测试、边界值测试、随机测试等。

4. 按软件质量特性划分

按软件质量特性划分，有功能性测试、可靠性测试、易用性测试、效率测试、可移植性测试和维护性测试。

功能性测试又包含适合性测试、准确性测试、互操作性测试、安全保密性测试、功能性依从性测试；可靠性测试包含容错性测试、成熟性测试、易回复性测试、可靠依从性测试；易用性测试包括易理解性测试、易学习测试、易操作性测试、吸引力测试、易用性依从性测试；效率测试包括时间特性测试、资源利用率测试、效率依从性测试；维护性测试包括改变性测试、稳定性测试、易测试性测试、



软件测试的分类

易分析性测试、易替换性测试、共从性测试和可移植性依从性测试。

5. 按照测试项目划分

按照测试项目划分,有功能测试、性能测试、接口测试、余量测试、强度测试、可靠性测试、安全性测试、人机界面测试、健壮性测试、恢复测试、安装测试、压力测试、负载测试等。

(1) 功能测试:主要针对软件/产品需求规格说明的测试,验证功能是否符合需求,包括原定功能的检验、是否有冗余的功能。

(2) 健壮性测试:侧重于软件容错能力的测试,主要是验证软件对各种异常情况(如数据边界、非法数据、异常中断等)是否进行正确处理。

(3) 恢复测试:对每一类导致恢复或重构的情况进行测试,验证软件自身运行的回复或重构,软件控制的系统的回复或重构,以及系统控制的软件的回复或重构。

(4) 人机界面测试:对人机界面提供的操作进行测试,测试人机界面的有效性、便捷性、直观性等,如用户界面是否友好、是否方便易用、设计是否合理、位置是否准确等。

(5) 接口测试:测试被测对象与其他软件(包括软件单元、部件、配置项)或硬件的接口。

(6) 强度测试:使软件在其设计能力的极限状态下,以及超过此极限下运行,检验软件对异常情况的抵抗能力。

(7) 可用性测试:对“用户友好性”的测试。受主观因素影响,且取决于最终用户。用户面谈、调查和其他技术都可使用。

(8) 压力测试:对系统不断施加压力的测试,通过确定一个系统的瓶颈或者不能接收的性能点,获得系统能提供的最大服务级别的测试。例如,测试一个 Web 站点在大量的负荷下,何时系统的响应会退化或失败。压力测试注重的是外界不断施压。

(9) 性能测试:测试软件是否达到需求规格说明中规定的各类性能指标,并满足相关的约束和限制条件。

(10) 兼容测试:测试软件在一个特定的硬件/软件/操作系统/网络等环境下的性能如何。

(11) 用户界面测试:对系统的界面进行测试,测试用户界面是否友好、是否方便易用、设计是否合理、位置是否正确等一系列界面问题。

(12) 安全性测试:测试软件在没有授权的内部或者外部的用户的攻击或者恶意破坏时如何处理,是否能保证软件和数据的安全。

(13) 可靠性测试:这里是比较狭义的可靠性测试,它主要是对系统能否稳定运行进行估计。

(14) 安装测试:安装测试主要检验软件是否可以正确安装,安装文件的各项设置是否有效,安装后是否影响原系统,卸载后是否删除干净,是否影响原系统。

(15) 文档测试:测试开发过程中生成的文档,以需求规格说明、软件设计、用户手册、安装手册等为主,检验文档是否和实际存在差别。文档测试不需要编写测试用例。

软件测试的这些类别之间有密切的关系,体现在以下方面。

(1) 在软件开发过程中,不同阶段的测试对应了对不同软件对象的测试,图 1-2 所示为这种对应关系。例如,集成测试对应了部件测试、确认测试对应了配置项测试。可以把按阶段/步骤分类的软件测试对等于按测试对象分类的软件测试。

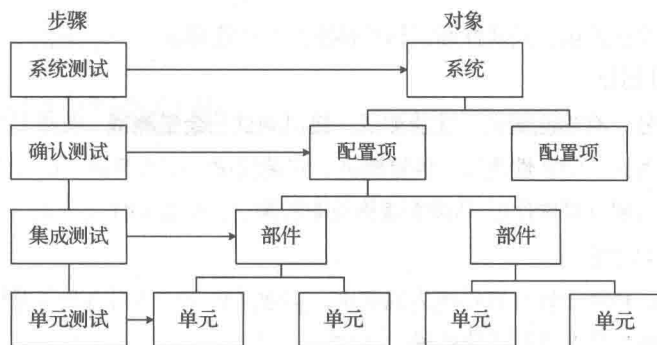


图1-2 软件测试的对象和步骤

(2) 在不同的测试阶段,由于测试目标、对象、要求的不同而采用不同的测试技术。表 1-1 所列为一般情况下不同测试对象中采用的测试技术。

表 1-1 测试对象和测试技术

测试对象	测试技术
单元测试	黑盒测试、白盒测试、静态测试
部件测试	黑盒测试、白盒测试、静态测试
配置项测试	黑盒测试、白盒测试
系统测试	黑盒测试

(3) 在不同的阶段对不同对象的测试包含不同的测试项目。例如,确认测试可包含功能测试、性能测试、人机界面测试;组合测试可包括接口测试;系统测试可包括可靠性测试、强度测试等。同时,对各阶段和对象的测试完整性要求也不同。

1.3 软件缺陷管理

1.3.1 软件缺陷的概念

软件缺陷常常又被叫作 bug。所谓软件缺陷,即计算机软件或程序中存在的某种破坏正常运行能力的问题、错误,或者隐藏的功能缺陷。缺陷的存在会导致软件产品在某种程度上不能满足用户的需要。IEEE729-1983 对缺陷有一个标准的定义:从产品内部看,缺陷是软件产品开发或维护过程中存在的错误、毛病等各种问题;从产品外部看,缺陷是系统所需要实现的某种功能的失效或违背。

一般看来,满足以下的任何一种情况都可以称为软件缺陷。

- (1) 软件未达到产品说明书中标明的功能。
- (2) 软件出现了产品说明书中指明的不会出现的功能。
- (3) 软件功能超出了产品说明书中指明的范围。
- (4) 软件未达到产品说明书中指明应达到的目。
- (5) 软件难以理解和使用、运行速度慢,或最终用户认为不好。



软件缺陷管理

以计算器开发为例。计算器的产品规格说明书中描述：计算器应能准确无误地进行加、减、乘、除运算。如果按下加法键，没什么反应，就是第1种类型的缺陷；若计算结果出错，也是第1种类型的缺陷。

产品规格说明书还可能规定计算器不会死机，或者停止反应。如果随意敲键盘导致计算器停止接受输入，这就是第2种类型的缺陷。

如果使用计算器进行测试，发现除了加、减、乘、除之外还可以求平方根，但是产品规格说明书没有提及这一功能模块。这是第3种类型的缺陷——软件实现了产品规格说明书中未提及到的功能模块。

在测试计算器时若发现电池没电会导致计算不正确，而产品规格说明书是假定电池一直都有电的，从而发现第4种类型的错误。

软件测试人员如果发现某些地方不对，比如测试人员觉得按键太小、“=”键布置的位置不好按、在亮光下看不清显示屏等，无论什么原因，都要认定为缺陷。而这正是第5种类型的缺陷。

1.3.2 软件缺陷的属性

发现缺陷后，需要提交缺陷单，通常情况下，缺陷单需要包含以下的内容。

(1) 缺陷标识 (Identifier)：缺陷标识是标记某个缺陷的一组符号。每个缺陷必须有一个唯一的标识。

(2) 缺陷类型 (Type)：缺陷类型是根据缺陷的自然属性划分的缺陷种类。类别通常可以分为以下几种情况，如表 1-2 所示。

表 1-2 缺陷类型

序号	类别	描述
1	界面	界面错误，如界面显示不符合需求、提示信息不合规等
2	功能	系统功能无效、不响应、不符合需求
3	性能	系统响应过慢、无法承受预期负荷等
4	安全性	存在安全隐患的缺陷
5	数据	数据导入或设置不正确

(3) 缺陷严重程度 (Severity)：缺陷严重程度是指因缺陷引起的故障对软件产品的影响程度，如表 1-3 所示。

表 1-3 缺陷严重程度

序号	缺陷严重程度	描述
1	严重缺陷	不能执行正常工作功能或重要功能。或者危及人身安全、系统安全
2	较大缺陷	严重地影响系统要求或基本功能的实现，且没有办法更正（重新安装或重新启动该软件不属于更正办法）
3	较小缺陷	影响系统要求或基本功能的实现，但存在合理的更正办法（重新安装或重新启动该软件不属于更正办法）