

软件测试



原理及应用

崔梦天 张 波 郭雪峰 / 编著

普通高等教育“十三五”通用性基础规划教材
西南民族大学教材建设基金资助出版

软件测试



原理及应用

RUANJIAN CESH
YUANLI JI YINGYONG

崔梦天 张波 郭雪峰 / 编著

谈文蓉 / 主审

内容简介

本书较为全面系统地介绍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。全书共分15章,主要包括软件工程与软件测试、软件测试概论、软件测试基础知识、软件测试过程、黑盒测试、白盒测试、自动化测试技术及其工具、性能测试、面向对象测试、软件测试管理以及软件测试文档模板等相关内容。

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,也可作为从事计算机应用开发的软件项目经理和需要了解软件测试的各类管理人员的参考书。

图书在版编目(CIP)数据

软件测试原理及应用 / 崔梦天, 张波, 郭雪峰编著

—成都: 西南交通大学出版社, 2019.7

普通高等教育“十三五”通用性基础规划教材

ISBN 978-7-5643-6939-2

I. ①软… II. ①崔… ②张… ③郭… III. ①软件-测试-高等学校-教材 IV. ①TP311.55

中国版本图书馆CIP数据核字(2019)第125812号

普通高等教育“十三五”通用性基础规划教材

软件测试原理及应用

崔梦天 张波 郭雪峰 / 编著

责任编辑 / 穆丰

封面设计 / 墨创文化

西南交通大学出版社出版发行

(四川省成都市金牛区二环路北一段111号西南交通大学创新大厦21楼 610031)

发行部电话: 028-87600564 028-87600533

网址: <http://www.xnjdcbs.com>

印刷: 四川森林印务有限责任公司

成品尺寸 185 mm × 260 mm

印张 16.25 字数 406千

版次 2019年7月第1版 印次 2019年7月第1次

书号 ISBN 978-7-5643-6939-2

定价 45.00元

课件咨询电话: 028-87600533

图书如有印装质量问题 本社负责退换

版权所有 盗版必究 举报电话: 028-87600562

前 言

随着软件的规模和复杂性的大幅提升，如何保证软件质量的可靠性变得日益重要。软件测试是保证软件质量的关键技术之一，同时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更新。软件测试从软件工程中演化而来，并且还在不断地发展之中。在学习本书之前，需要一些先行知识（如掌握一门高级语言、数据库、数据结构以及软件工程的基本理论知识等）作为支撑。

本书较为全面地涵盖了当前测试领域的专业知识，共分为5个部分，第1部分介绍了软件测试的基本含义以及与软件测试相关的数学基础知识；第2部分介绍了进行软件测试常用的两种方法：白盒测试和黑盒测试；第3部分介绍了软件测试的实用技术：单元测试、集成测试、性能测试等；第4部分介绍了软件测试自动化技术，包括JUnit测试框架和自动化测试工具QuickTest Professional；第5部分则给出了进行软件测试所需的相关文档模板。本书综合性强，注重实践与理论相结合，用详细、完整的案例以及理论知识引领读者进入实际操作，主要面向初、中级读者，由“入门”起步，侧重“提高”。

本书的出版得到西南民族大学教材建设基金资助。与此同时，该教材也得到了四川省软件工程专业卓越工程师培养项目以及西南民族大学计算机学院实验基地建设项目的联合资助。

本书主要由西南民族大学崔梦天老师、中国人民武装警察部队学院张波老师、洛阳师范学院郭雪峰老师、山西师范大学孙清亮老师、赤峰学院贾丽美老师共同编写完成。其中第1章、第3章由崔梦天老师、郭雪峰老师完成；第2章、第5章、第6章、第7章由郭雪峰老师完成；第4章由孙清亮老师完成；第8章~第14章由张波老师完成；第15章由贾丽美老师完成。

本书在编写过程中得到了多方面的帮助、指导和支持。感谢西南民族大学计算机学院领导，他们为软件测试方向的课程规划与建设付出了大量的心血，特别感谢谈文蓉教授，花了大量的时间对全书做了审阅，并提出了宝贵的修改意见。在写此书过程中，作者参考和引用了大量的书籍和文献资料。在此，也向被引用文献的作者以及给予此书帮助的所有人士表示衷心的感谢。

由于编者水平有限，书中难免有不足之处，恳请广大读者批评指正，并提出宝贵意见。

编 者

2018年10月

目 录

第 1 部分 软件测试基础

第 1 章 软件测试概述	1
1.1 软件测试的意义	1
1.2 软件测试的发展史及现状	4
1.3 软件测试的特点和原则	7
1.4 软件测试的基本方法	9
1.5 软件测试中的认识误区	10
1.6 软件测试的心理学问题	10
1.7 测试人员的能力要求和职业前景	12
1.8 小 结	14
第 2 章 测试人员应掌握的离散数学知识	15
2.1 集合论	15
2.2 函 数	17
2.3 测试人员的图论	20
2.4 用于测试的图	25
第 3 章 软件测试过程	29
3.1 概 述	29
3.2 测试与软件开发各阶段的关系	29
3.3 软件测试的过程与策略	30
3.4 测试计划	40
3.5 小 结	48

第 2 部分 软件测试方法

第 4 章 白盒测试方法	50
4.1 白盒测试概述	50
4.2 典型的白盒测试方法	50
4.3 白盒测试的典型案例	59

4.4	白盒测试的工具	67
第 5 章	黑盒测试技术	70
5.1	黑盒测试概述	70
5.2	黑盒测试的测试用例设计方法	71

第 3 部分 软件测试的实用技术

第 6 章	单元测试	88
6.1	单元测试的内容	88
6.2	单元测试的测试方法	89
6.3	单元测试过程	89
6.4	单元测试活动	91
6.5	测试过程准则	93
6.6	单元测试用例设计	93
6.7	单元测试执行	94
6.8	单元测试报告	98
6.9	小 结	98
第 7 章	集成测试	99
7.1	集成测试概念	99
7.2	集成测试的内容和方法	99
7.3	集成测试原则	99
7.4	集成测试内容	100
7.5	集成测试方法	101
7.6	集成测试过程	102
7.7	集成测试举例	103
7.8	集成测试经验	105
第 8 章	性能测试技术	106
8.1	性能测试技术概述	106
8.2	性能测试的内容	109
8.3	性能测试的测试用例	111
8.4	性能测试的自动化工具和操作方法	114
8.5	性能测试案例	121
8.6	小 结	124
第 9 章	确认测试、系统测试和验收测试	125
9.1	确认测试的概念和活动	125
9.2	系统测试的概念和类型	126

9.3	回归测试	128
9.4	验收测试	130
9.5	小 结	131
第 10 章	软件的其他测试技术	132
10.1	可用性测试	132
10.2	容错性测试	132
10.3	易用性测试	133
10.4	安全性测试	134
10.5	可靠性测试	134
10.6	需求测试	136
10.7	小 结	137

第 4 部分 软件测试自动化技术

第 11 章	软件测试自动化	138
11.1	自动化测试概述	138
11.2	自动化测试的引入和实施	139
11.3	软件自动化测试的原理、方法和级别	143
11.4	软件自动化测试工具	145
11.5	小 结	150
第 12 章	JUnit 测试框架	151
12.1	初识 JUnit 测试框架	151
12.2	JUnit 框架分析	154
12.3	用 TestCase 来工作	156
12.4	创建单元测试方法	159
12.5	TestSuite	160
12.6	Fail 和 Error	161
12.7	创建 TestCalculator 全过程	164
12.8	深入 JUnit 4	169
第 13 章	JUnit 和类测试	180
13.1	类测试概念	180
13.2	确定类测试用例	183
13.3	类测试代码实例	188
13.4	JUnit 测试的延伸	191
第 14 章	自动化测试工具 QuickTest Professional	208
14.1	QuickTest Professional 的安装	208

14.2	测试流程	209
14.3	录制测试	214
14.4	运行并分析测试	216
14.5	创建检查点	218
14.6	参数化测试	223
14.7	创建输出值	225
14.8	使用正则表达式	228
14.9	将测试拆分为多操作	229
14.10	小 结	230

第 5 部分 软件测试的相关文档

第 15 章	软件测试文档模板	231
15.1	测试大纲写作模板	231
15.2	测试计划写作模板	232
15.3	测试用例写作模板	237
15.4	功能测试报告写作模板	241
15.5	性能测试报告写作模板	242
15.6	集成测试报告写作模板	244
15.7	系统测试报告写作模板	245
15.8	验收测试写作模板	247
15.9	测试分析报告模板	249
	参考文献	251

第1章 软件测试概述

1.1 软件测试的意义

1.1.1 软件缺陷的典型例子

随着计算机技术的飞速发展，一方面使计算机软件产品能够应用到社会的各个领域，给人们的学习、生活带来了巨大的方便；另一方面，一些计算机软件产品的错误或缺陷也给人们带来了诸多不便甚至是灾难。下面是一些造成较大影响的软件缺陷的典型例子。

1. 美国迪斯尼公司的狮子王游戏软件缺陷（Bug）

1994年圣诞节前夕，迪斯尼公司发布了第一个面向儿童的多媒体光盘游戏 Lion King Animated Storybook（狮子王动画故事书）。由于迪斯尼公司的著名品牌影响和事先的大力宣传及良好的促销活动，市场销售情况非常不错，该游戏几乎成为父母为自己孩子过圣诞节的必买礼物。出人意料的是，12月26日，圣诞节后的第一天，迪斯尼公司客户支持部的电话开始响个不停，不断有人投诉，抱怨为什么游戏总是安装不成功，或没法正常使用，很快报纸和电视开始不断报道此事。后来证实，迪斯尼公司没有对当时市场上的各种PC（个人计算机）机型进行完整的系统兼容性测试，只是在几种PC机型上进行了相关测试。所以这个游戏软件只能在少数系统中正常运行，但在大众使用的其他常见系统中却不能正常安装和运行。

2. 美国航天局火星登陆事故

1999年12月3日，美国航天局的火星极地登陆飞船在试图登陆火星表面时失踪。错误修正委员会观测到了故障，并认定出现误动作的原因极可能是某一个数据位被意外修改。大家一致声讨，问题为什么没有在内部测试时解决。后来发现，登陆飞船之前通过了多个小组进行测试，其中一个小组测试飞船的脚落地位置，另一个小组测试此后的着陆过程。前一个小组没有注意着地数据位是否置位，而后一个小组却在开始测试之前直接重置计算机，清除了数据位，由于各小组缺乏合理的沟通，从而导致了事故的发生。

3. 跨世纪“千年虫”问题

20 世纪 70 年代的一位程序员，为了节省空间，在开发工资系统时将四位日期缩减为两位。当时计算机存储器的成本很高，这样的做法无可厚非。后来，随着计算机技术的迅速发展，存储器的价格不断降低，但在计算机系统中使用两位数字来表示年份的做法却由于思维上的惯性而被沿袭下来，年复一年，直到新世纪即将来临之际，大家才突然意识到用两位数字表示年份将无法正确辨识公元 2000 年及其以后的年份。1997 年，信息界开始拉起了“千年虫”警钟，并很快引起了全球关注。虽然经过多方努力，此问题基本得到解决，但更换或升级系统以解决 2000 年份问题的费用超过了数亿美元。

4. 金星探测飞行失败

1963 年，由于用 FORTRAN 程序设计语言编写的飞行控制软件中的循环语句 DO 5 I=1, 3 误写为 DO 5 I=1.3，将逗号误写为点号，结果导致美国首次金星探测飞行失败，造成价值 1000 多万美元的损失。

上述案例中的软件问题在软件工程或软件测试中都被称为软件缺陷或软件故障。

1.1.2 软件缺陷的产生原因

软件缺陷，常常又被叫作 Bug，即为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。

符合下面 5 个规则的任何一或数个，我们就称之为软件缺陷。

- (1) 软件未达到产品说明书中已经标明的功能；
- (2) 软件出现了产品说明书中指明不会出现的错误；
- (3) 软件未达到产品说明书中虽未指出但应当达到的目标；
- (4) 软件功能超出了产品说明书中指明的范围；
- (5) 软件测试人员认为软件难以理解、不易使用，或者最终用户认为该软件使用效果不良。

在软件开发的过程中，软件缺陷的产生是不可避免的。那么造成软件缺陷的主要原因有哪些？由于软件产品是整个软件开发活动的成果结晶，因此需要从软件开发的各阶段去寻找原因。

按照软件工程的观点，软件开发的主要阶段依次为：需求分析、软件设计、软件编码、软件测试、软件运行和维护。

通过对大量软件及软件缺陷的研究，人们发现导致软件缺陷的最大的原因是软件需求（产品说明书），排在第二位的是软件设计，而人们通常认为的最容易导致缺陷的软件编码仅仅排在第三位。软件缺陷产生原因分布如图 1-1 所示。

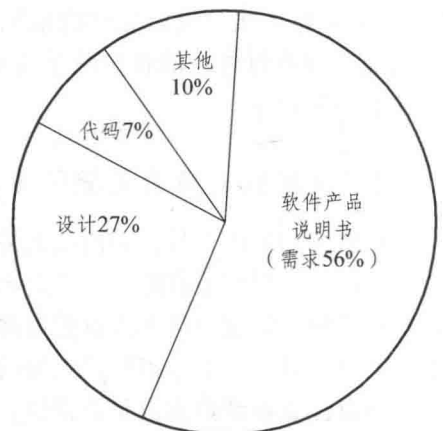


图 1-1 软件缺陷产生的原因分布

产品说明书是造成软件缺陷的罪魁祸首有不少原因。我们通常把规格说明书扩展为需求功能说明、功能规格说明、操作规格说明、使用规格说明等软件文档。首先，软件的用户大多是非计算机专业人士，软件开发人员和用户对要开发的软件功能理解可能不一致，从而引起沟通障碍，致使产品的功能设计不能完全满足用户需要。其次，用户的需要也在不断变化，这些变化没有及时在说明书中得到体现。最后，整个开发团队之间没有进行充分的沟通。这些都导致了软件缺陷的产生。

设计方案是软件缺陷产生的第二大来源。原因与产品说明书类似，如方案设计得不够全面，容易变化，开发小组成员之间沟通不足等。

软件编码排在了导致软件缺陷出现的第三位。如现在的软件系统越来越复杂，开发人员不太可能精通所有的技术；说明文档质量普遍比较糟糕，或者文档本身就有错误；开发人员常处于进度的压力之下，容易产生错误等。

另外，还有其他一些原因，由于每种原因占的比例都很小，可将其归为一类，主要的原因还是前三类。

从上面分析可以看出，软件在各个阶段都有可能产生缺陷。测试人员应从需求分析就介入进去，尽早发现问题，及时修复缺陷。缺陷发现或解决得越迟，更正缺陷或修复问题的代价就越高。图 1-2 表明了同一缺陷在不同阶段发现时的修复费用对比。

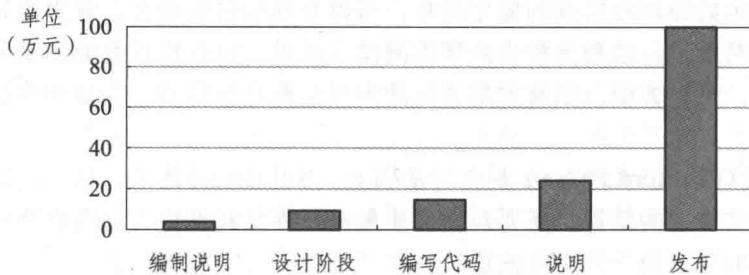


图 1-2 软件缺陷在不同阶段发现时修复的费用示意图

统计表明，在典型的软件开发项目中，软件测试工作量往往占软件开发总工作量的 40% 以上。而在软件开发的总成本中，用在测试上的开销要占 30%~50%。如果把维护阶段也考虑在内，讨论整个软件生存期时，测试的成本比例也许会有所降低，但实际上维护工作相当于二次开发，乃至多次开发，其中必定还包含有许多测试工作。因此，测试对于软件生产具有重要的意义。

1.1.3 软件测试的目标

对于软件测试的目标，基于不同的立场，存在着两种完全不同的观点。从用户的角度出发，普遍希望通过软件测试暴露软件中隐藏的错误和缺陷，而考虑是否可以接受该产品。而软件开发者则希望通过测试验证软件产品中不存在错误，证明该软件已正确地实现了用户的要求，确立人们对软件质量的信心。

测试的目标决定了测试方案的设计，如果是为了表明程序是正确的而进行测试，就会设计一些不易暴露错误的设计方案；相反，如果测试是为了发现程序中的错误，就会设计出力

求发现错误的设计方案。因此，若由软件开发者自己测试或站在开发者的立场去测试，则会选择那些导致程序失效概率小的测试用例，回避那些易于暴露程序错误的测试用例。同时，也不会刻意去检测、排除程序中可能包含的副作用。所以，这样的测试对完善和提高软件质量毫无价值。因为在程序中往往存在着许多预料不到的问题，可能会被疏漏，许多隐藏的错误只有在特定的环境下才可能暴露出来。如果不把着眼点放在尽可能查找错误这样一个基础上，这些隐藏的错误和缺陷就会查不出来，会遗留到运行阶段中去。如果站在用户的角度，替他们设想，就应当把测试活动的目标对准揭露程序中存在的错误。在选取测试用例时，考虑那些易于发现程序错误的测试数据，这样就利于提高软件质量。

有鉴于此，Glendford.J.Myers在《the art of software testing》一书中提出了这样的观点：

- (1) 软件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。

测试的目标应该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。如果成功地实施了测试，就能够发现软件中的错误。但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也是错误的认识。一方面，测试并不仅仅是为了找出错误，通过分析错误产生的原因和错误的发生趋势，可以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。这种分析也能帮助测试人员设计出有针对性的测试方法，改善测试的效率和有效性。另一方面，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。

1983年，在Glendford.J.Myers观点的基础上，Bill Hetzel指出，软件测试的目标不仅是尽可能多地发现软件中的错误，还要对软件质量进行度量和评估，以提高软件质量。这一论断将对软件测试的认识提升到更高的层次。

而几乎同时，IEEE提出的软件工程术语中也对软件测试进行了定义，指出软件测试是使用人工或自动手段来运行或测试某个系统的过程，其目的在于检验它是否满足规定的需求或弄清楚预期结果与实际结果之间的差别。这个定义明确指出：软件测试的目的是为了检验软件系统是否满足需求，它也不再仅是一个一次性的、且只是开发后期的活动，而是与整个开发流程融成一体。

1.2 软件测试的发展史及现状

1.2.1 软件测试的发展史

软件测试是伴随着软件的产生而产生的。早在20世纪50年代，英国著名的计算机科学家图灵就给出了软件测试的原始定义。他认为，测试是程序正确性证明的一种极端实验形式。

但当时相对于软件开发而言，软件测试还处于次要位置。软件测试的含义比较狭隘，开

发人员将测试等同于“调试”，主要针对机器语言和汇编语言，设计特定的测试用例，运行被测程序，将所得结果与预期结果进行比较，从而判断程序的正确性，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的工作。测试用例一般在随机选取的基础上，吸取测试者的经验或是凭直觉判断出某些重点测试区域。但测试在软件开发中的作用并没有受到应有的重视。测试方法和理论研究进展缓慢，除去一些非常关键的程序系统外，一般程序的测试大都是不完备的。在开发工作结束后，投入运行的程序仍然残存着一些缺陷。这些隐藏的缺陷一旦暴露出来就会给用户和维护者带来不同程度的后果。

直到 20 世纪 70 年代，随着软件技术的发展，软件的复杂度越来越高，人们才逐渐认识到软件测试的重要意义。1972 年，在北卡罗莱纳大学举行了首届软件测试正式会议。1975 年，John Good Enough 和 Susan Gerhart 发表了名为《测试数据选择的原理》的文章，使软件测试得到了许多研究者的重视。1979 年，Glendford J. Myers 出版了软件测试领域非常重要的一本专著《软件测试的艺术》(The Art of Software Testing)，该书中对软件测试的定义、目标等进行了描述，认为测试是为发现错误而执行程序的过程，测试的目的在于尽可能多地发现软件中的错误。

到了 20 世纪 80 年代，人们的质量意识越来越高。Bill Hetzel 于 1983 年出版了《软件测试完全指南》(Complete Guide of Software Testing) 一书，该书指出：测试是以评价一个程序或系统属性为目标任何一种活动，是对软件质量的度量。同年，IEEE 提出的软件工程术语中给软件测试也下了定义，指出软件测试是使用人工或自动手段来运行或测试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楚预期结果与实际结果之间的差别。这个定义明确指出：软件测试的目的是为了检验软件系统是否满足需求。

到了 20 世纪 90 年代，软件行业开始迅猛发展，软件的规模变得非常大，而当时软件测试的手段几乎完全都是靠人工测试，测试的效率非常低，测试活动需要花费大量的时间和成本；并且随着软件复杂度的提高，在一些大型软件开发过程中，出现了很多通过手工方式无法完成测试的情况。于是，很多测试工作实践者开始尝试开发商业的测试工具来支持测试，辅助测试人员完成某一类型或某一领域内的测试工作，这样使得测试工具逐渐盛行起来。通过运用测试工具，可以达到提高测试效率的目的。测试工具的发展，大大提高了软件测试的自动化程度，让测试人员从烦琐和重复的测试活动中解脱出来，专心从事有意义的测试设计等活动，而测试工具的选择和推广也越来越受到重视。同时，越来越多的测试模型被提出，测试的成熟度问题也得到了众多专家的关注，并提出了若干测试的成熟度模型，如测试能力成熟度 TCMM (Testing Capability Maturity Model)、测试支持度 TSM 等。

到了 2002 年，Rick 和 Stefan 在《系统的软件测试》一书中对软件测试做了进一步定义：测试是为了度量和提高被测软件的质量，对测试软件进行工程设计、实施和维护的整个生命周期过程。

1.2.2 软件测试的国内外现状

在国外，尤其是软件业发达的国家，软件测试已得到了软件从业人员的普遍重视，软件测试不仅早已成为软件开发的一个有机组成部分，而且在整个软件开发的系统工程中占据着相当大的比重。以美国的软件开发和生产的平均资金投入为例，通常是：“需求分析”和“规

划确定”各占3%，“设计”占5%，“编程”占7%，“测试”占15%，“投产和维护”占67%。测试在软件开发中的地位，由此可见一斑。

我国的软件测试与软件业发达的国家相比仍存在不少差距，不过，随着软件的复杂程度的提高，国内IT企业也逐步开始重视软件测试团队的建设，在IT企业中，一些知名企业已经将软件测试作为企业未来发展的一个板块。在国内软件测试行业中各种软件测试的方法、技术和标准都还在探索阶段。

为了较全面地了解我国软件测试发展现状，作为国内最大的软件测试门户网站，51Testing软件测试网于2008年策划组织了中国首届企业软件测试现状调查活动，调查问卷所有项目均针对各企业软件测试部门而设立，目的是为企业和测试人员提供测试领域最全面、真实、有效的各项数据和分析，帮助企业和个人了解软件测试发展现状，有针对性地提高自身的软件测试技术水平和管理水平。

1.2.3 软件测试的发展趋势

1. 建立大的独立测试中心是软件测试发展的一个趋势

随着软件行业的快速发展，系统越来越精密，软件也越来越复杂，影响的范围也不断扩大。因此，这个时期开发的软件就必须进行十分严格的测试。而IT管理开发与运行维护分离的趋势，也推动着独立测试中心的诞生和发展。目前国内虽已涌现出很多的第三方测试机构，但其中很多仍然处于发展的初级阶段，行业自律也还不足。而金融、电信等很多行业对软件稳定性又非常依赖，因此，建立大的独立测试中心尤为必要。

2. 自动化测试工具的应用将会更普遍、更充分

软件测试很大程度上是一种重复性工作，这种重复性表现在同样的一个功能点或是业务流程需要借助不同类型的数据驱动运行很多遍。同时，由于某一个功能模块的修改而有可能影响其他模块而需要进行回归测试，也需要测试人员重复执行以前用过的测试用例，正是基于以上原因，人们才想出了自动化测试方法，同时计算机技术的发展也为自动化测试的实现提供了必要条件。就自动化技术的使用情况来看，大多数公司是使用负载测试工具进行性能测试，大多数软件往往不具备自动化功能工具应用的条件，自动化测试工具大规模的应用还需要一定的时间。但随着软件业的不断发展，自动化测试工具的应用必将会更普遍、更充分。

3. 测试技术将越来越细化，越来越专业

软件测试技术的发展正在经历不断细分的发展过程。最初大家只是关注于功能测试，于是引进了功能技术理论，近年来随着人们质量意识的提高，开始看重软件产品或系统的性能，经过测试工作者及测试厂商的努力，性能测试工具得到了较为广泛的应用，并总结出了在性能测试方面相关的一些理论与方法。除此之外，根据软件应用领域及软件类型的不同，出现了一些更加专业的测试技术类型，如手机软件测试和嵌入式软件测试等。

手机软件测试这个研究分支的出现，主要是因为手机上的软件种类越来越丰富，出现了很多专门从事手机软件的开发公司，于是自然而然出现一批手机软件测试的工程师。同时由

于手机软件的特殊性，如使用一些专门的操作系统，加上手机内存及 CPU（中央处理器）相对较小等特点，手机软件的测试有其特殊的技术方法，2009 年电子工业出版社出版了《手机软件测试最佳实践》一书，感兴趣的读者可以参考一下。

由于嵌入式系统的自身特点，如实时性（Real-timing），内存不丰富，I/O 通道少，开发工具昂贵，与硬件紧密相关，CPU 种类繁多等。嵌入式软件的开发和测试也就与一般商用软件的开发和测试策略有了很大的不同，相对来说操作起来困难会更大一些。

1.3 软件测试的特点和原则

1.3.1 软件测试的特点

在软件开发活动中，软件测试阶段具有不同于分析、设计、编码阶段的特点，表现在：

1. 复杂性

要做好软件测试，不仅需要站在客户的角度思考问题，真正理解客户的需求，具有良好的分析能力和创造性的思维能力，完成功能和用户界面的测试，而且还能理解软件系统的实现机理和各种使用场景，具有扎实的技术功底，通过测试工具完成相应的性能测试、安全性测试、兼容性测试和可靠性测试等更具挑战性的任务。从这些角度看，测试需要测试人员的业务分析能力、对客户需求的理解能力和团队沟通协作的能力。

2. 挑剔性

软件测试是为了让开发出的软件满足用户需求，而对软件质量进行监督和保证。其目的不是为了证明程序中没有错误，而是为了尽早发现程序中的错误，以减少修复错误的成本。因此，软件测试是一种“挑剔性”的行为。

3. 不彻底性

所谓彻底测试，就是让被测程序在一切可能的输入情况下全部执行一遍，通常也称这种测试为“穷举测试”。然而在许多实际的测试中，由于测试用例数量巨大，穷举测试往往是无法实现的。

4. 经济性

软件工程的总目标是充分利用有限的人力和物力资源，高效率、高质量地完成测试。而软件测试的不彻底性，要求我们在软件测试中要对测试用例进行选择，要选用典型的、有代表性的测试用例进行有限的测试。为了降低测试成本，选择测试用例时应注意遵守“经济性”的原则。第一，要根据程序的重要性和一旦发生故障将造成的损失来确定测试用例的等级；第二，要认真研究测试策略，以便能使用尽可能少的测试用例，发现尽可能多的程序错误。

1.3.2 软件测试的基本原则

除了掌握测试的技术外，还必须遵循一些基本原则。

1. 应尽早地和不断地进行软件测试

软件开发者应牢记“尽早地和不断地进行测试”。IBM360 操作系统的历史教训表明，缺陷修复成本随着各个阶段的靠后而上升，软件缺陷的积累与放大效应是导致软件危机的最主要原因。图 1-3 所示为缺陷放大模型大致状况。另外，不应该把软件测试仅仅看作是软件开发的一个独立阶段，而应当把它贯穿于整个软件的生命周期中。

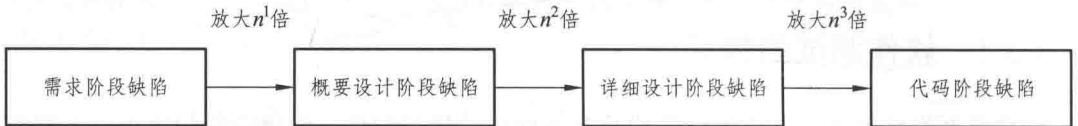


图 1-3 缺陷放大模型

2. 不可能进行完全的测试

对一个程序进行完全测试就意味着在测试结束之后，再也不会发现其他的软件错误了，但这是不可能的。即使是规模很小的软件或者软件产品，其逻辑路径和输入数据的组合也几乎是无穷的，测试人员基本不可能实现对测试对象进行全面的检查和覆盖。

3. 增量测试，由小到大

无论是传统的软件测试还是面向对象的软件测试都要遵循“增量测试、由小到大”的原则。这里的由小到大，指的是软件测试的粒度。

4. 软件开发者应避免测试自己的程序

为达到较好的测试效果，软件开发者应避免测试自己的代码，而由独立的专业测试机构来完成。与软件开发者相比，作为第三方的软件测试机构，具有客观性、专业性和权威性更高的优点，因此，测试由独立的专业测试机构来完成，对提高软件测试的有效性具有重要意义。

5. 设计周密的测试用例

测试用例是测试工作的核心，软件测试的本质就是针对要测试的内容确定一组测试用例。好的测试用例是软件测试成功的前提。因此，测试用例应该尽量设计的周密细致，这样才能更好地保证测试工作的质量。

6. 注意错误集中的现象

在软件测试中，一定要注意测试中的错误集中发生现象，即通常情况下，哪个模块已发现的缺陷越多，其残存的未被发现的错误也越多，在该模块就需要投入更多的人力精力去测试。

7. 合理安排测试计划

合理的测试计划有助于测试工作顺利有序地进行，因此要求在对软件进行测试之前所做的测试计划中，应该结合了多种针对性强的测试方法，列出所有可使用资源，建立一个正确的测试目标；要本着严谨、准确的原则，周到细致地做好测试前期的准备工作，避免测试的随意性。

8. 确认 Bug 的有效性

在测试过程中，有时候测试人员提交的 Bug 并不是真正的 Bug。因此我们必须对 Bug 进行确认，保证其有效性。图 1-4 具体地描述了无效 Bug 的来源。一般由 A 测试人员发现的 Bug，一定要由另外一个 B 测试人员来进行确认，如果发现严重的 Bug 可以召开评审会进行讨论和分析。

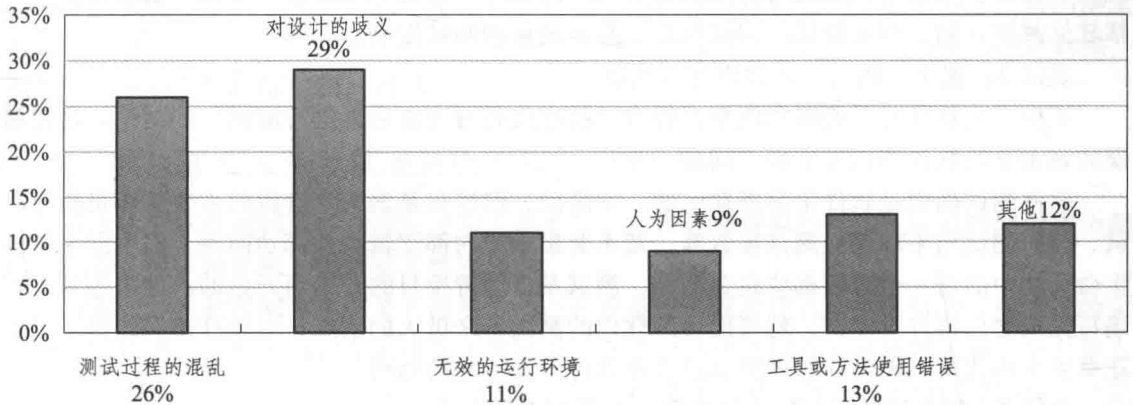


图 1-4 无效 Bug 来源构成图

9. 要进行回归测试

回归测试是指在软件发生修改之后重新进行先前的测试以保证修改的正确性。在软件生命周期中，只要软件发生了改变，就可能给该软件带来问题。所以在软件测试中，一定要进行回归测试，以便确定修改是否达到了预期的目的，检查修改是否破坏原有的正常功能。

10. 应保留一切测试用例，并对测试结果进行统计和分析

测试完成后，应保留一切测试用例，并对测试结果进行统计和分析。这样有利于正确地鉴别测试后输出的数据，给出清晰的错误原因分析报告。

1.4 软件测试的基本方法

软件测试的方法和技术是多种多样的。对于软件测试技术，可以从不同的角度加以分类，本书后面的章节将做具体介绍。