

高等学校“十三五”规划教材·计算机软件工程系列

Software Testing Technology and Tool Application

# 软件测试技术与工具应用

◎主 编 王 妍

◎副主编 刘 双 郑丽丽

紧扣基础知识 / 案例梯度递进 / 内容通俗易懂

项目驱动教学 / 实验全面覆盖 / 理论与实践结合



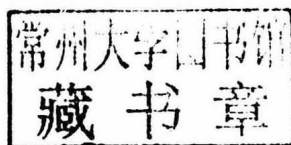
哈尔滨工业大学出版社  
HARBIN INSTITUTE OF TECHNOLOGY PRESS

高等学校“十三五”规划教材·计算机软件工程系列  
曲阜师范大学教材建设基金资助出版

# 软件测试技术与工具应用

主 编 王 妍

副主编 刘 双 郑丽丽



哈尔滨工业大学出版社

## 内 容 简 介

本书根据教学、科研和校企合作项目编写而成，共分 10 章：第 1 章讲述软件测试基本概念；第 2 章讲述软件测试基本流程；第 3 章讲述白盒测试的方法和技巧；第 4 章讲述黑盒测试的方法和技巧；第 5 章讲述软件缺陷及缺陷管理；第 6 章讲述测试过程度量及测试总结报告的撰写；第 7 章讲述开发者测试的方法及工具软件；第 8 章讲述功能测试的过程及工具；第 9 章讲述性能测试的过程及工具；第 10 章讲述 Web 应用测试的过程及工具。

本书采用了一种简单、易于接受的方式进行编写，即采用范例法教学，书中包含了大量案例和详解过程，且每个知识点都附有大量例题和习题。读者可以参照例题完成习题，以达到事半功倍、举一反三的效果。

本书可作为计算机及软件相关专业的本、专科生教材，也可作为软件测试人员的基本参考资料，还可作为各种与测试相关的技术资格水平考试的学习辅导用书。

### 图书在版编目 (CIP) 数据

软件测试技术与工具应用/王妍主编. —哈尔滨:  
哈尔滨工业大学出版社, 2019.7  
ISBN 978-7-5603-8421-4

I. ①软… II. ①王… III. ①软件—测试 IV.  
①TP311.55

中国版本图书馆 CIP 数据核字 (2019) 第 142616 号

策划编辑 王桂芝  
责任编辑 王桂芝 张 荣  
出版发行 哈尔滨工业大学出版社  
社 址 哈尔滨市南岗区复华四道街 10 号 邮编 150006  
传 真 0451-86414749  
网 址 <http://hitpress.hit.edu.cn>  
印 刷 黑龙江艺德印刷有限责任公司  
开 本 787mm×1092mm 1/16 印张 18.25 字数 450 千字  
版 次 2019 年 7 月第 1 版 2019 年 7 月第 1 次印刷  
书 号 ISBN 978-7-5603-8421-4  
定 价 49.80 元

---

(如因印装质量问题影响阅读，我社负责调换)

# 前 言

2014 年学院在校企合作中开拓了软件测试这个方向，本人从事“软件测试技术”这门课程的教学工作也已经 5 年了，在不断摸索前进的过程中，体会了教训也总结了经验，很想把教学过程中的这些体会记录下来，因此组织编写了本书。

本书前半部分以软件测试过程为引导，将软件测试基础、软件测试需求分析、软件测试计划的制订、软件测试用例的设计、软件测试执行及缺陷管理、软件测试度量及总结等相关过程的具体任务和技术方法穿插其中，辅以大量例题及详解过程，使读者可以通过自主学习融会贯通。本书后半部分从开发者测试、功能测试、性能测试和 Web 应用测试 4 个不同角度，分别介绍了测试内容、测试任务及相关的自动化测试工具，使读者可以学以致用，理论结合实践。

本书语言精练，通俗易懂，具有较强的条理性、系统性和逻辑性。全书共分 10 章，具体内容如下：

第 1 章 软件测试基础，主要讲述软件测试的定义，测试的目的、原则、分类、发展，软件质量及质量标准，测试人员应具备的基本素养等内容。

第 2 章 软件测试流程及流程管理，按照测试需求分析、测试计划制订、测试用例设计、测试执行、缺陷管理、测试总结这一过程，描述各个阶段的具体任务，并介绍开源的软件测试流程管理平台 TestLink 的具体应用方法。

第 3 章 白盒测试技术，主要讲述静态测试、逻辑覆盖测试、基本路径测试的原理、方法及应用。

第 4 章 黑盒测试技术，主要讲述等价类划分、边界值分析、决策表与决策树、因果图、场景法、正交实验法等黑盒测试技术的原理、方法和应用。

第 5 章 软件缺陷及缺陷管理，主要介绍缺陷的定义、属性、分类，缺陷报告的书写方法和原则，并介绍自动化缺陷管理工具 Mantis 的应用方法。

第 6 章 软件测试度量及测试报告，主要讲述测试过程度量的定义、方法、指标项及测试报告的具体内容。

第 7 章 开发者测试，主要介绍单元测试和集成测试的方法，并引入单元测试工具 Junit。

第 8 章 功能测试，主要讲述功能测试的方法、过程及自动化测试工具 UFT。

第 9 章 性能测试，主要讲述性能测试的方法、过程及自动化测试工具 LoadRunner。

第 10 章 Web 应用测试，主要讲述 Web 应用测试的方法、过程及自动化测试工具 Selenium 的环境搭建。

本书结合教学中校企合作的经验，以实验为驱动，注重学生动手能力的培养。本书可以作为计算机及软件相关专业的本、专科生教材，也可作为软件测试人员的基本参考资料，还可作为各种与测试相关的技术资格水平考试的学习辅导用书。

本书由王妍担任主编，刘双和郑丽丽担任副主编，具体编写分工如下：第1~6章、9~10章由王妍编写；第7章由刘双编写；第8章由郑丽丽编写。全书由王妍统稿，高峰同学进行校对。本书由曲阜师范大学教材建设基金资助出版，在此表示感谢；另外，高仲合老师、倪建成老师对全书提出了宝贵意见，在此一并表示感谢。由于作者水平有限，书中难免存在疏漏和不妥之处，敬请读者不吝指正，不胜感谢。

编者

2019年3月

于曲园

# 目 录

第 1 章 软件测试基础	1
1.1 软件测试基本概念	1
1.1.1 软件与软件测试	1
1.1.2 软件测试的定义	2
1.1.3 软件测试的目的	2
1.1.4 软件测试原则	3
1.2 软件测试的分类	4
1.3 软件质量与质量模型	5
1.3.1 软件质量定义	5
1.3.2 软件质量模型	5
1.4 软件测试发展史	9
1.5 测试工作者的必备能力	10
1.6 小结	11
课后习题	11
第 2 章 软件测试流程及流程管理	12
2.1 软件测试模型	12
2.1.1 V 模型	12
2.1.2 W 模型	13
2.1.3 H 模型	14
2.1.4 X 模型	14
2.1.5 前置测试模型	15
2.2 软件测试流程	16
2.3 软件测试需求	16
2.4 软件测试计划	20
2.4.1 为什么制订测试计划	20
2.4.2 如何制订测试计划	20
2.4.3 测试计划报告	21
2.5 测试用例的设计	24

2.5.1	测试用例概述	24
2.5.2	测试用例设计的原则	24
2.5.3	测试用例的构成	25
2.6	测试执行	28
2.7	测试总结	29
2.8	自动化测试管理工具——TestLink	30
2.8.1	设置用户和创建测试项目	30
2.8.2	创建测试需求	32
2.8.3	创建测试计划	34
2.8.4	创建测试用例	35
2.8.5	测试任务的分配	37
2.8.6	测试的执行	37
2.8.7	分析测试结果	38
2.9	小结	38
	课后习题	39
<b>第3章</b>	<b>白盒测试技术</b>	<b>40</b>
3.1	静态测试技术	40
3.1.1	代码检查法	40
3.1.2	静态结构分析法	41
3.2	逻辑覆盖法设计测试用例	41
3.2.1	语句覆盖	41
3.2.2	判定覆盖(又称分支覆盖)	43
3.2.3	条件覆盖	44
3.2.4	判定/条件覆盖	44
3.2.5	条件组合覆盖	45
3.3	基本路径测试法	48
3.3.1	基本路径测试的定义	48
3.3.2	基本路径法设计测试用例的步骤	53
3.4	白盒测试的综合应用	58
3.5	小结	58
	课后习题	58
<b>第4章</b>	<b>黑盒测试技术</b>	<b>60</b>
4.1	等价类划分	60
4.1.1	等价类定义	60
4.1.2	划分等价类的原则	61
4.1.3	等价类划分的形式	62
4.1.4	等价类划分设计测试用例的步骤	64

4.2	边界值分析	67
4.2.1	边界值分析法定义	67
4.2.2	边界值分析法的形式	68
4.2.3	边界值分析法的原则	71
4.3	决策表与决策树	73
4.3.1	决策表法定义	73
4.3.2	决策表的组成	74
4.3.3	决策表设计测试用例	76
4.3.4	决策树	82
4.4	因果图	85
4.4.1	因果图定义	85
4.4.2	因果图法设计测试用例	87
4.5	场景法	91
4.5.1	场景法定义	91
4.5.2	场景法设计测试用例	92
4.6	正交试验法	93
4.6.1	正交试验法简介	93
4.6.2	用正交试验法设计测试用例	95
4.7	黑盒测试的使用策略	101
4.8	小结	102
	课后习题	102
第5章	软件缺陷及缺陷管理	103
5.1	软件缺陷概述	103
5.1.1	缺陷的定义	103
5.1.2	软件缺陷分类及属性	104
5.2	软件缺陷的状态转换	106
5.3	缺陷报告	108
5.3.1	缺陷报告原则	108
5.3.2	缺陷报告内容	108
5.4	软件缺陷管理	113
5.4.1	缺陷管理系统	113
5.4.2	自动化缺陷管理平台 Mantis	113
5.5	小结	119
	课后习题	120
第6章	软件测试度量及测试报告	121
6.1	软件测试度量的定义	121
6.2	软件测试度量指标	122

6.3	软件测试度量难度	125
6.4	测试总结报告	127
6.5	小结	128
	课后习题	128
<b>第7章</b>	<b>开发者测试</b>	<b>129</b>
7.1	单元测试	129
7.1.1	单元测试定义	129
7.1.2	单元测试的目标	130
7.1.3	单元测试中常见的问题	130
7.2	Junit 在单元测试中的应用	131
7.2.1	Junit 简介	131
7.2.2	Eclipse 中使用 Junit 进行单元测试	131
7.3	集成测试	135
7.3.1	集成测试的定义	136
7.3.2	集成测试的目标	136
7.3.3	集成测试策略	136
7.4	Junit 在集成测试中的应用	139
7.5	小结	140
	课后习题	140
<b>第8章</b>	<b>功能测试</b>	<b>141</b>
8.1	功能测试的概述	141
8.1.1	功能测试的定义	141
8.1.2	功能测试类型	141
8.2	功能测试过程	143
8.2.1	功能测试需求分析	144
8.2.2	功能测试计划制订	145
8.2.3	功能测试设计与开发	147
8.2.4	测试执行与缺陷跟踪	148
8.2.5	功能测试报告	149
8.3	功能测试工具及 UFT 的应用	151
8.3.1	功能自动化测试概述	151
8.3.2	功能测试工具的选择	152
8.3.3	自动化测试项目流程	152
8.3.4	自动化功能测试工具 UFT 的应用	153
8.4	小结	167
	课后习题	167

第 9 章 性能测试	168
9.1 性能测试的概念	168
9.1.1 性能问题典型案例	168
9.1.2 性能测试的定义	169
9.1.3 功能测试与性能测试的区别	169
9.1.4 性能测试相关术语	170
9.2 性能测试过程	172
9.2.1 性能需求分析	173
9.2.2 性能测试计划	174
9.3 性能测试工具 LoadRunner 应用	175
9.3.1 性能测试脚本的生成	176
9.3.2 场景设计及管理	178
9.3.3 测试结果分析	183
9.4 小结	185
课后习题	185
第 10 章 Web 应用测试	186
10.1 Web 应用系统概述	186
10.2 Web 应用测试的类型	187
10.3 Selenium 自动化 Web 应用测试	191
10.3.1 Selenium 简介	191
10.3.2 基于 python 的 Selenium 环境搭建	192
10.3.3 基于 Java 的 Selenium 环境配置	197
10.4 小结	203
课后习题	203
附录 软件测试实验	204
实验 1 TestLink 的安装与配置	204
实验 2 TestLink 应用实例 (1)	213
实验 3 TestLink 应用实例 (2)	220
实验 4 白盒测试——逻辑覆盖法	227
实验 5 白盒测试——基本路径测试法	230
实验 6 等价类划分法设计测试用例	232
实验 7 边界值分析法设计测试用例	236
实验 8 黑盒测试——判定表	237
实验 9 黑盒测试——因果图分析法	241
实验 10 缺陷管理工具 Mantis 的安装与配置	246
实验 11 缺陷管理工具 Mantis 应用练习	250

---

实验 12 TestLink 与 Mantis 的集成 .....	255
实验 13 单元&集成测试 .....	257
实验 14 UFT 初体验 .....	259
实验 15 UFT 检查点练习 .....	266
实验 16 LoadRunner 性能测试初体验 .....	273
参考文献 .....	280

# 第 1 章 软件测试基础

随着软件在各个行业的普及应用，软件测试成为大多数行业提高软件产品质量的保障手段，软件测试越来越受到人们的重视。本章作为引导，对软件测试的基本概念、软件质量及质量模型、软件发展历程做了相关介绍，目的是让读者对软件测试的框架有一个直观的了解。

## 1.1 软件测试基本概念

### 1.1.1 软件与软件测试

软件一般定义为与计算机系统操作有关的计算机程序、规程、规则，以及可能有的文件、文档及数据。简单讲，软件=程序+数据+文档。程序是能够完成预定功能和性能的可执行的指令序列；数据是使程序能够适当地处理信息的数据结构；文档是开发、使用和维护程序所需要的图文资料。软件的构成如图 1.1 所示。

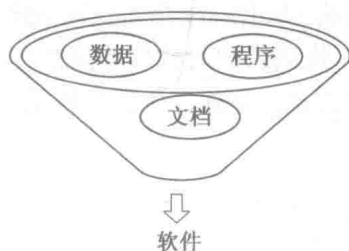


图 1.1 软件基本构成图

软件开发是一种直接将人类的脑力劳动转换为产品的行为，因此软件从性质上来说归根结底是一种商品。既然是商品就要有质量管理和质量保障，软件测试是软件质量保证的有效手段。

什么是软件测试？软件测试 = 发现 Bug？这个问题恐怕不能简单地回答对或者不对，因为不同的时间段，软件开发的方式和特点不同，因此测试的方式和特点也不同。软件开发早期，软件的规模较小，复杂程度低，软件开发的过程跟软件测试的过程没有明显的区分，开发者在调试程序的过程中发现并修改了软件中存在的 Bug，测试的工作往往由开发者完成，这一时期的测试等同于寻找 Bug。随着软件开发方法的工程化，软件规模越来越大，复杂程度越来越高，软件测试不再单纯是为了发现程序中的 Bug，对于构成软件的另外两个要素——数据和文档，也列入了测试的范畴。软件测试成为软件工程化过程的一部分，独立于软件开

发成为软件全生命周期中不可或缺的一部分，是确保软件质量的重要手段。

### 1.1.2 软件测试的定义

到目前为止，还没有任何一个非常官方的软件测试的定义，大多数定义都是从软件测试的目的方面考虑给出的。按照时间推移，对软件测试的定义大致有以下几种。

#### 1. 定义1 (Hetzel, 1973)

1973年，Bill Hetzel 博士给软件测试定义为：测试是对程序能够按预期运行建立起一种信心 (Establish confidence that a program does what it is supposed to do)。后来，1983年他又将定义修改为：测试是以评价一个程序和系统的特性或能力，并确定它是否达到预期结果的任何行为 (Any activities aimed at evaluating an attribute or capability of a program or system)。不管是前者还是后者，其核心思想是“测试是为了证明正确性”。

#### 2. 定义2 (Myers, 1979)

1979年，Glenford J. Myers 在其代表作 *The Art of Software Testing* 中对软件测试定义为：测试是为发现错误而执行程序的过程 (The process of executing a program or system with the intent of finding errors)。其核心思想是“测试是为了证明错误”。

#### 3. 定义3 (IEEE 定义)

1983年，IEEE 提出的软件工程术语中，给软件测试下的定义是：使用人工或自动的手段来运行或测量软件系统的过程，以检验软件系统是否满足规定的要求，并找出与预期结果之间的差异。

综上理解，个人认为软件测试 (Software Testing)，通俗地来说，就是软件在正式投入运行前，为了保证软件的正常运行，提高用户对软件的满意度而对软件开发过程中的需求分析、设计和编码过程进行的最终复审活动，是软件生存周期的一个重要的组成阶段，是软件质量保证的关键步骤。

### 1.1.3 软件测试的目的

软件测试的目的是为了保证软件产品的最终质量。Glenford J. Myers 在《*The Art of Software Testing*》一书中提到测试是一个程序的执行过程，其目的在于发现错误。一个好的测试用例很可能会发现至今尚未察觉的错误，一个成功的测试是发现了至今尚未察觉的错误的测试。综合分析，软件测试大概有下面3个目的。

#### (1) 发现错误。

软件测试是为了发现错误而执行程序的过程，测试是为了证明程序有错，而不是证明程序无错。对软件进行的测试越多、越充分，人们对使用该软件的信心就越强；但不能因为在测试活动中没有发现错误就保证软件是完全正确没有潜在缺陷的。

#### (2) 确认与验证。

确认软件的质量，其一方面是确认软件做了你所期望做的事情，另一方面是确认软件以正确的方式做了这个事情。

验证软件的质量是否满足用户的需求，评价程序或系统的属性，对软件质量进行度量和

评估, 为用户选择、接受软件提供有力的依据。

### (3) 持续改进开发及测试过程。

通过分析测试过程中发现的问题, 可以帮助改进开发工作采用的软件过程; 同时通过缺陷分析, 可以找到缺陷关联, 提高再测试的缺陷发现率。

总之, 软件测试是以发现错误为目标的活动。这一过程应该尽量用最少的人力、物力和时间找出软件中的显性及隐性的错误, 以验证软件满足用户的需求程度, 提高软件的最终质量, 回避软件发布后由于潜在缺陷造成的隐患所带来的商业风险。

## 1.1.4 软件测试原则

为了使测试过程更加完善, 测试效果更加有效, 测试人员在执行测试活动的过程中应该遵循如下原则。

### (1) 尽早测试。

在软件或系统开发生命周期中, 测试活动应该尽可能早地介入, 并且应该将关注点放在已经定义的测试目标上。一般来说, 当软件工程进行到需求分析阶段时, 就可以开展测试活动了。另一方面早期发现错误进行修正的成本要少于晚期发现错误进行修正的成本。比如, 同一个错误在需求分析阶段被发现, 要比在发布阶段被发现而进行修正花费的成本少。

### (2) 全面测试。

全面测试应该从两个方面来理解, 一方面测试的对象要全面, 不仅是程序代码, 还有数据和相关文档。另一方面是参与测试的人员要全面, 开发者、测试者及用户都应该参与到测试工作中。

### (3) 全过程测试。

测试人员不能仅仅把精力放在测试的过程中, 还应该关注整个开发过程。只有对开发过程有相当的了解, 才能制订合理的测试方案及测试用例。另外, 测试人员还应该对测试的全过程进行跟踪管理, 尤其对自己发现和提交的缺陷, 应该对其全生命周期进行跟踪监控。

### (4) 穷尽测试是不可能的。

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。假如测试人员想对测试对象进行完全的检查 and 覆盖, 那基本上是不可能的。因此, 需要选择合适的测试技术来设计测试用例, 用尽可能少的测试用例发现尽可能多的缺陷。

### (5) Pareto 原则。

版本发布前进行的测试所发现的大部分缺陷和软件运行失效是由于少数软件模块引起的, 即测试发现 80% 的错误很可能起源于 20% 的模块中。

### (6) 避免自己测试自己的程序。

由于心理因素的影响或者程序员本身错误地理解了需求或者规范, 导致程序中存在错误, 应避免程序员或者编写软件的组织测试自己的软件。一般要求由专门的测试人员进行测试, 并且还要求用户参与, 特别是验收测试阶段, 用户是主要的参与者。必要时寻找第三方测试公司进行专业的测试是更合理的选择。

### (7) 严格按照测试计划实施测试过程。

制订严格的测试计划, 并把测试时间安排得尽量宽松, 不要希望在极短的时间内完成一个高水平的测试。测试计划是整个测试过程的统领, 测试需要的人力、物力、财力资源在测

试计划中都要有统筹预算，因此在测试过程中应该严格按照测试计划开展测试活动。

(8) 明确软件的质量标准。

只有建立了质量标准，才能根据测试的结果，对产品的质量进行分析和评估。同样，测试用例应该确定期望输出结果。如果无法确定测试期望结果，则无法进行检验。必须用预先精确对应的输入数据和输出结果来对照检查当前的输出结果是否正确，做到有的放矢。

(9) 注意回归测试的关联性。

修改一个错误有可能引入更多错误，因此回归测试非常必要，但要注意在回归测试中更新测试用例，避免出现“杀虫剂悖论”。如果采用同样的测试用例多次重复进行测试，则测试用例发现缺陷的能力将大大下降。

(10) 避免“同化效应”。

测试人员与开发人员在一个项目中待的时间久了，容易受开发人员观点的影响；测试人员对软件的熟悉程度越高，越容易忽略一些细微错误。测试过程中应避免这种同化效应造成的 Bug 免疫。

## 1.2 软件测试的分类

目前软件测试的名称有很多，按照不同的测试方法有不同的分类，从不同的角度理解可大体进行如下分类。

(1) 按测试的方式分类，软件测试可以分为静态测试和动态测试。

静态测试指不运行被测程序本身，仅通过分析或检查源程序的语法、结构、过程、接口等来检查程序的正确性。对需求规格说明书、软件设计说明书、源程序做结构分析、流程图分析、符号执行来找出欠缺和可疑之处。静态测试结果可用于进一步的查错，并为测试用例选取提供指导。

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、正确性和健壮性等性能问题。这种方法一般由构造测试用例、执行程序、分析程序的输出结果 3 部分组成。

(2) 按测试的技术方法分类，软件测试可以分为黑盒测试、白盒测试和灰盒测试。

黑盒测试又被称为功能测试、数据驱动测试或基于规格说明的测试，是通过使用整个软件或某种软件功能来严格地测试，而并没有检查程序的源代码或者很清楚地了解该软件的源代码程序的内部逻辑结构。

白盒测试又被称为结构测试、逻辑驱动测试或基于代码的测试，是通过执行测试用例对程序的源代码进行测试的方法。这种类型的测试需要从代码句法发现内部代码在算法、溢出、路径、条件中的缺点或者错误，进而加以修正。

灰盒测试是介于白盒测试与黑盒测试之间的一种测试，灰盒测试多用于集成测试阶段，不仅关注输出、输入的正确性，同时也关注程序内部的情况。灰盒测试不像白盒测试那样详细、完整，但又比黑盒测试更关注程序的内部逻辑，常常是通过一些表征性的现象、事件、标志来判断内部的运行状态。

(3) 按测试阶段或测试步骤划分，软件测试可以分为单元测试、集成测试、系统测试和验收测试。

单元测试是指对软件中的最小可测试单元进行检查和验证的活动。

集成测试是指在单元测试的基础上,将所有模块组装成为子系统或系统进行测试的活动。

系统测试是在集成测试之后,将硬件、软件、操作人员看作一个整体,检验它是否有不符合系统需求说明书规定的测试活动。

验收测试是软件产品完成了单元测试、集成测试和系统测试之后,产品发布之前由用户所进行的软件测试活动。

(4) 按测试目的划分,软件测试可分为功能测试、性能测试、界面测试、安全测试、可靠性测试、文档测试、兼容性测试、恢复性测试、安装/反安装测试等约 32 种测试类型。这里只介绍几种常用的测试类型,其他类型将在系统测试中具体介绍。

① 功能测试:对产品的各功能进行验证,根据功能测试用例逐项测试,检查产品是否达到用户需求的功能。

② 性能测试:通过自动化的测试工具模拟多种正常、峰值及异常负载条件来对系统的各项性能指标进行测试的活动。

③ 界面测试:测试用户界面的功能模块布局是否合理、整体风格是否一致、各个控件的放置位置是否符合客户使用习惯,从美学的角度衡量界面美观性的活动。

④ 安全测试:对产品进行检验以验证产品是否符合安全需求定义和产品质量标准的过程。

⑤ 可靠性测试:通过长时间运行程序以检验软件出现故障的相隔时间是否满足用户要求的测试活动。

⑥ 文档测试:检验软件在开发过程中产生的文档是否具有完整性、正确性、一致性、易理解性、易浏览性等特性的活动。

⑦ 兼容性测试:检查软件与软件之间、软件与硬件之间能否正确地进行交互和信息共享。

⑧ 恢复性测试:软件在发生故障后能否恢复到运行状态的测试活动。

## 1.3 软件质量与质量模型

### 1.3.1 软件质量定义

软件质量是软件符合明确叙述的功能和性能需求、文档中明确描述的开发标准,以及所有专业开发的软件都应具有的显式和隐含的特征相一致的程度。一般判断软件质量优劣的标准为:

(1) 软件需求是度量软件质量的基础,与需求不一致就是质量不高。

(2) 指定的标准定义了一组指导软件开发的准则,如果没有遵守这些准则,肯定会导致质量不高。

(3) 通常,有一组没有显式描述的隐含需求(如期望软件是容易维护的)。如果软件满足明确描述的需求,却又不满足隐含的需求,那么软件的质量仍然是值得怀疑的。

### 1.3.2 软件质量模型

软件质量是软件的生命,它直接影响软件的使用与维护,以及满足顾客的满意度。评价软件的质量因素很多,如正确性、精确性、可靠性、容错性、性能、效率、易用性、可理解

性、可维护性、可复用性、可扩充性、可移植性等。面对众多的质量因素如何取折衷，这实际上就是区分质量因素对软件质量影响程度轻重的问题，为解决这个问题专家们提出了很多软件质量模型。常见的质量模型有 McCall 模型、Boehm 模型、ISO/IEC 9126 模型、ISO/IEC 25010 模型。

### 1. McCall 模型

McCall 软件质量模型（1977 年）中的质量概念基于 11 个特性之上，这 11 个特性分别面向软件产品的运行、修正和转移，如图 1.2 所示。

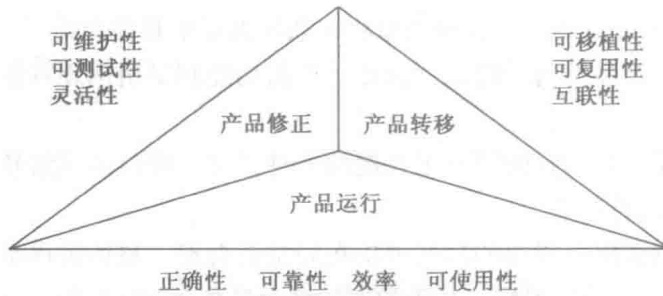


图 1.2 McCall 模型

从图 1.2 中可以看出，在所有质量指标中运行指标是基础，决定其他质量是否具有可测性，只有在产品运行没有问题的基础上才会考虑对产品修正和转移方面的测试。

### 2. Boehm 模型

Boehm 软件质量模型（1978 年）试图通过一系列的属性指标来量化软件质量。该质量模型包含了 McCall 模型中没有的硬件属性。Boehm 质量模型采用层级的质量模型结构，它由质量特性、质量子特性、度量 3 个层次构成，如图 1.3 所示。

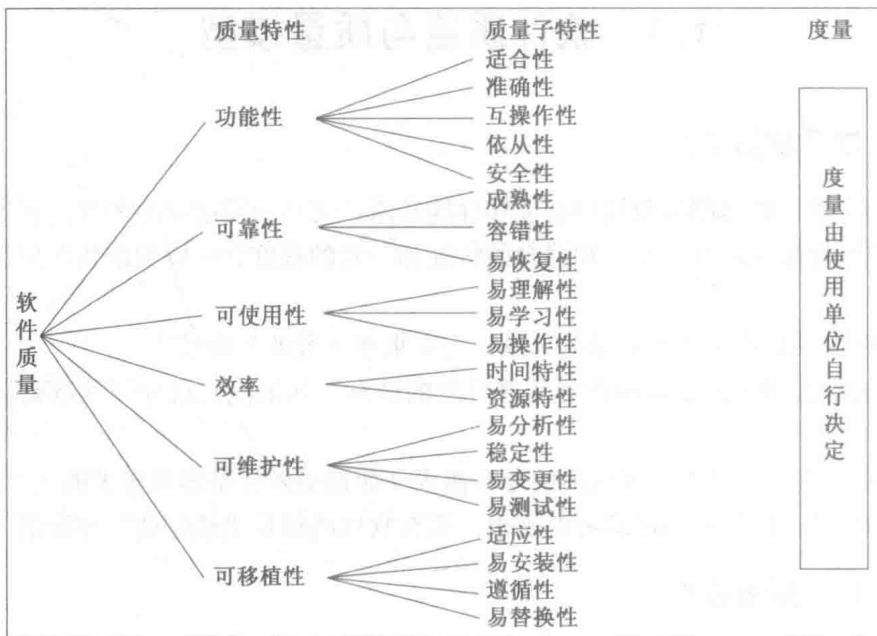


图 1.3 Boehm 模型