

21世纪软件工程专业规划教材

软件测试

周元哲 编著

10010101000101

111010010010

10010101000101

111010010010

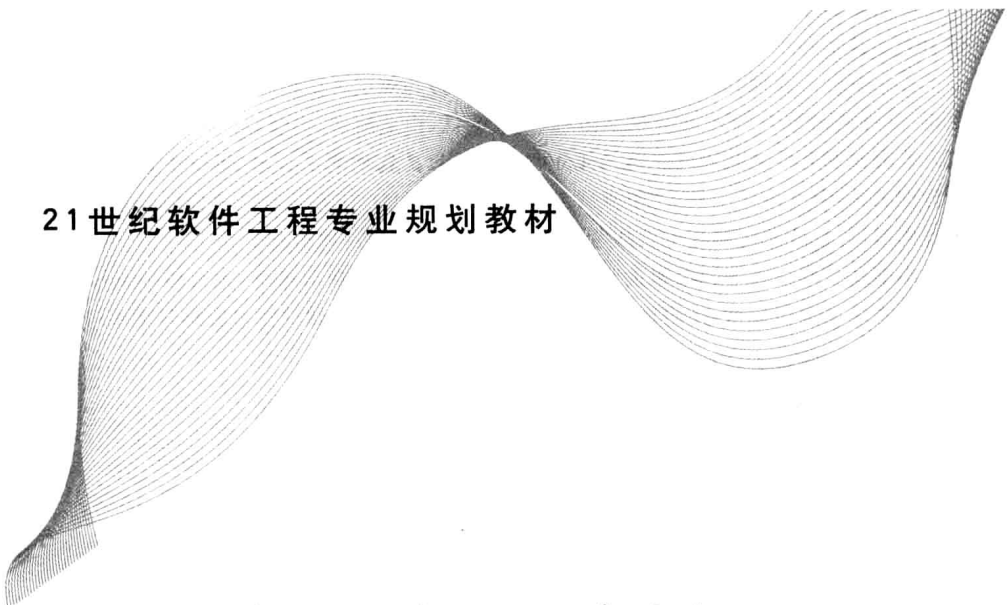
1000101

11101001

10010101000101

111010010010

清华大学出版社



21世纪软件工程专业规划教材

软件测试

周元哲 编著

清华大学出版社
北京

内 容 简 介

本书较为全面、系统地涵盖了当前业界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。

全书共分三大部分,分别是测试理论、测试实践和测试考试指导。第一部分内容主要包括软件测试概论、软件测试基本知识、软件测试过程、黑盒测试、白盒测试、自动测试技术、性能测试、面向对象测试、嵌入式测试和软件测试管理。第二部分内容主要包括软件测试工具、测试管理工具、性能测试工具、缺陷管理工具、单元测试工具、功能测试工具、嵌入式测试工具等。第三部分内容主要包括四级软件测试工程师考试、企业招聘测试工程师考试和微软公司测试的一些情况。

本书适合作为高等院校相关专业软件测试的教材或教学参考书,也可以供从事计算机应用开发的各类技术人员应用参考,或用作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。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

软件测试 / 周元哲编著. --北京:清华大学出版社, 2013

21世纪软件工程专业规划教材

ISBN 978-7-302-33192-6

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159103号

责任编辑:张 玥 顾 冰

封面设计:傅瑞学

责任校对:时翠兰

责任印制:刘海龙

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:010-62786544

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

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:17

字 数:387千字

版 次:2013年9月第1版

印 次:2013年9月第1次印刷

印 数:1~2000

定 价:33.00元

前言

P R E F A C E

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。软件测试作为成为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革新。

本书较为全面地涵盖了当前业界测试领域的专业知识,追溯了软件测试的发展史,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。全书共分三大部分,分别是测试理论、测试实践和测试考试指导。第一部分内容主要包括软件测试概论、软件测试基本知识、软件测试流程、黑盒测试、白盒测试、性能测试、面向对象测试、嵌入式测试和软件质量保证。第二部分内容主要包括软件测试自动化与测试工具、测试管理工具、性能测试工具、缺陷管理工具、单元测试工具、功能测试工具和嵌入式测试工具等。第三部分内容主要包括四级软件测试工程师考试、企业招聘测试工程师考试和微软公司软件测试的一些情况。

本书在写作过程中参阅了大量中外文的专著、教材、论文、报告及网上的资料,由于篇幅所限,未能一一列出。在此,向各位作者表示敬意和衷心的感谢。

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,需要一些先行知识作为本书的支撑,如掌握一门高级语言,这包括 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。

西安邮电学院计算机科学与技术学院院长胡滨、潘晓英、刘海、刘有耀对本书的编写给予了大力的支持并提出了指导性意见。北京化工大学信息科学与技术学院计算机系主任赵瑞莲教授、清华大学出版社对本书的写作大纲、写作风格等提出了很多宝贵的意见。

本书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,内容精练、文字简洁、结构合理、综合性强,明确定位面向初、中级读者,由“入门”起步,侧重“提高”。特别适合作为高等院校相关专业软件测试的教材或教学参考书,也可以供从事计算机应用开发的各类技术人员应用参考,或用作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。

由于作者水平有限,时间紧迫,本书难免有不足之处,诚恳期待读者的批评指正,以使本书日臻完善。我们的电子信箱是 zhouyuanzhe@163.com。

编者
2013年6月

目 录

C O N T E N T S

第一部分 测试理论

第 1 章 软件测试概论	3
1.1 软件与软件项目	3
1.1.1 软件发展史	3
1.1.2 项目特性	4
1.1.3 软件项目	5
1.1.4 项目生命周期	6
1.2 软件缺陷	7
1.2.1 软件缺陷案例	7
1.2.2 软件缺陷	9
1.2.3 软件缺陷流程	12
1.3 习题	13
第 2 章 软件测试基本知识	15
2.1 测试发展历程	15
2.2 测试的几种观点	16
2.3 软件测试目的与原则	18
2.4 软件测试分类	20
2.4.1 按照测试阶段划分	20
2.4.2 按照执行主体划分	21
2.4.3 按照执行状态划分	21
2.4.4 按照测试技术划分	23
2.5 软件测试模型	26
2.5.1 V 模型	26
2.5.2 W 模型	26
2.5.3 H 模型	27
2.5.4 X 模型	28

2.5.5	前置模型	28
2.6	测试用例	29
2.7	习题	32
第3章	软件测试流程	33
3.1	测试流程概述	33
3.2	测试计划	34
3.3	测试设计	35
3.4	单元测试	35
3.4.1	概述	35
3.4.2	内容	36
3.4.3	步骤	38
3.5	集成测试	39
3.5.1	主要任务	39
3.5.2	集成测试方法	39
3.6	确认测试	42
3.7	系统测试	43
3.8	验收测试	44
3.8.1	α 测试和 β 测试	44
3.8.2	回归测试	44
3.9	评估测试	47
3.10	习题	48
第4章	黑盒测试	50
4.1	概述	50
4.2	等价类划分	51
4.2.1	划分原则	51
4.2.2	设计测试用例步骤	51
4.3	边界值分析法	53
4.3.1	设计原则	54
4.3.2	应用举例	54
4.4	决策表	56
4.4.1	应用举例	57
4.4.2	优点和缺点	59
4.5	因果图	59
4.5.1	基本术语	59
4.5.2	应用举例	61
4.6	场景法	62
4.6.1	基本流和备选流	62

4.6.2 应用举例	62
4.7 综合策略	66
4.8 习题	67
第5章 白盒测试	69
5.1 概述	69
5.2 逻辑覆盖法	70
5.2.1 语句覆盖	70
5.2.2 判定覆盖	71
5.2.3 条件覆盖	71
5.2.4 条件判定覆盖	72
5.2.5 修正条件判定覆盖	73
5.2.6 条件组合覆盖	74
5.2.7 路径覆盖	75
5.2.8 综合举例	75
5.3 路径分析	78
5.3.1 控制流图	78
5.3.2 基路径测试	80
5.3.3 循环测试	82
5.3.4 逻辑覆盖法与路径测试比较	83
5.4 数据流测试	84
5.4.1 变量定义/引用分析	84
5.4.2 程序片	86
5.5 程序插桩	86
5.6 习题	88
第6章 性能测试	90
6.1 基本概念	90
6.2 性能测试分类	92
6.2.1 负载测试	92
6.2.2 压力测试	93
6.2.3 可靠性测试	95
6.2.4 数据库测试	95
6.2.5 安全性测试	96
6.2.6 文档测试	96
6.3 性能测试的步骤	97
6.4 网站测试	99
6.4.1 网站结构模型	99

6.4.2	网站测试内容	100
6.5	习题	100
第7章	面向对象测试	102
7.1	面向对象影响测试	102
7.2	面向对象测试模型	103
7.3	面向对象分析测试	104
7.3.1	对象测试	104
7.3.2	结构测试	105
7.3.3	主题测试	105
7.3.4	属性和实例关联测试	106
7.3.5	服务和消息关联测试	106
7.4	面向对象设计测试	107
7.5	面向对象单元测试	108
7.5.1	功能性和结构性测试	109
7.5.2	测试用例设计和选择	109
7.6	面向对象集成测试	110
7.7	面向对象系统测试	112
7.8	习题	112
第8章	嵌入式软件测试	113
8.1	嵌入式系统	113
8.1.1	基本概念	113
8.1.2	嵌入式系统软件架构	114
8.1.3	嵌入式系统开发方式	114
8.2	嵌入式软件测试	115
8.2.1	测试特点	115
8.2.2	测试策略	115
8.2.3	三种测试环境	117
8.2.4	测试流程	118
8.3	嵌入式软件测试工具	119
8.3.1	纯软件测试工具	119
8.3.2	纯硬件测试工具	120
8.3.3	软硬结合测试工具	120
8.4	习题	120
第9章	软件质量保证	121
9.1	软件测试管理	121

9.2	软件测试文档	121
9.2.1	测试文档的类型	122
9.2.2	测试文档的重要性	123
9.3	测试人员组织	124
9.3.1	测试团队架构	124
9.3.2	测试团队阶段性	125
9.4	软件缺陷管理	126
9.4.1	概述	126
9.4.2	缺陷跟踪流程	127
9.4.3	缺陷跟踪管理系统概述	127
9.5	软件质量	128
9.5.1	概述	128
9.5.2	ISO 9000 系列	129
9.5.3	CMM/CMMI	130
9.5.4	ISO 15504 过程评估	133
9.6	习题	133

第二部分 测试实践

第 10 章	软件测试自动化与测试工具	137
10.1	自动化测试	137
10.2	测试成熟度模型	138
10.3	测试工具原理	143
10.3.1	白盒测试工具	143
10.3.2	黑盒测试工具	144
10.3.3	测试设计和开发工具	146
10.3.4	测试执行和评估工具	147
10.3.5	测试管理工具	147
10.4	测试工具选择	148
10.5	习题	149
第 11 章	测试管理工具	151
11.1	概述	151
11.2	测试管理工具——TestDirector	152
11.2.1	TestDirector 简介	152
11.2.2	TestDirector 使用概述	153

第 12 章 性能测试工具	165
12.1 综述	165
12.2 LoadRunner 测试流程	166
12.3 项目实践	166
12.3.1 使用 VuGen 创建脚本	167
12.3.2 使用 Controller 设计场景	172
12.3.3 使用 Controller 运行场景	174
12.3.4 分析场景结果	175
第 13 章 缺陷管理工具	178
13.1 Bugzilla	178
13.2 JIRA	182
13.2.1 跟踪操作	183
13.2.2 查询操作	185
13.2.3 生成报表	186
第 14 章 单元测试工具	188
14.1 JUnit 特点	188
14.2 JUnit 在 eclipse 中的使用	188
第 15 章 功能测试工具	194
15.1 WinRunner	194
15.1.1 WinRunner 测试模式	194
15.1.2 WinRunner 测试流程	194
15.1.3 WinRunner 测试举例	197
15.2 QuickTest Professional 简介	202
15.2.1 QuickTest Professional 测试过程	202
15.2.2 使用 Mercury Tours 范例网站	202
15.2.3 QTP 测试范例	203
第 16 章 嵌入式软件测试工具	213
16.1 Logiscope 简介	213
16.2 Logiscope 三大功能	214
16.2.1 使用 Audit	215
16.2.2 使用 RuleChecker	224
16.2.3 使用 TestChecker	228

第三部分 测试考试指导

第 17 章 全国计算机等级考试四级软件测试工程师	241
17.1 内容介绍	241
17.1.1 考试说明	242
17.1.2 考试大纲及考试重点	242
17.2 相关资料	247
第 18 章 软件测试行业	249
18.1 测试行业现状	249
18.2 软件测试职位	250
18.3 软件测试思维方式	251
18.4 常用软件测试工程师笔试题	252
第 19 章 微软公司软件测试	256
19.1 微软测试策略	256
19.2 一道微软测试题目	258
参考文献	259

第  部分

测试理论

软件测试概论

本章介绍软件的发展历史、软件项目的特性和生命周期等内容,并对软件缺陷的内容以及缺陷的后果做了介绍,引出了软件测试的内容,为学习本书后续内容作必要准备。

1.1 软件与软件项目

软件是一系列按照特定顺序组织的计算机数据和指令的集合。一般来讲软件被划分为系统软件、应用软件和介于这两者之间的中间件。其中系统软件为计算机使用提供最基本的功能,但是并不针对某一特定应用领域。而应用软件则恰好相反,不同的应用软件根据用户和所服务的领域提供不同的功能。

一般认为,软件包括如下内容:

- (1) 运行时,能够提供所要求功能和性能的指令或计算机程序集合。
- (2) 程序能够满意地处理信息的数据结构。
- (3) 描述程序功能需求以及程序如何操作和使用所要求的文档。

1.1.1 软件发展史

20世纪50年代初期至60年代中期是软件发展的第一阶段(又称为程序设计阶段)。此时硬件已经通用化,而软件的生产却是个体化。软件产品为专用软件,规模较小,功能单一,开发者即使用者,软件只有程序,无文档。软件设计在人们的头脑中完成,形成了“软件等于程序”的错误观念。

第二阶段从20世纪60年代中期至70年代末期是程序系统阶段。此时多道程序设计技术、多用户系统、人机交互式技术、实时系统和第一代数据库管理系统的出现,出现了专门从事软件开发的“软件作坊”,软件广泛应用,但软件技术和管理水平相对落后,导致“软件危机”出现。软件危机主要表现在以下几个方面:

- (1) 软件项目无法按期完成,超出经费预算,软件质量难以控制;
- (2) 开发人员和开发过程之间管理不规范,约定不严密,文档书写不完整,使得软件维护费用高,某些系统甚至无法进行修改;
- (3) 缺乏严密有效的质量检测手段,交付给用户的软件质量差,在运行中出现许多问题,甚至带来严重的后果;
- (4) 系统更新换代难度大。

第三阶段称为软件工程阶段,从 20 世纪 70 年代中期至 80 年代中期,由于微处理器的出现、分布式系统广泛应用,以软件的产品化、系列化、工程化和标准化为特征的软件产业发展起来,软件开发有了可以遵循的软件工程化的设计准则、方法和标准。1968 年,北大西洋公约组织的计算机科学家在联邦德国召开国际会议,讨论软件危机问题,正式提出并使用“软件工程”概念,标志软件工程诞生。软件工程学涉及与生产软件相关的所有活动,包括计算机科学、管理学、经济学、心理学等,其研究的主要内容是:如何应用科学的理论和工程上的技术来指导软件的开发,从而达到以较少的投资获得高质量软件的最终目标。

第四阶段是从 20 世纪 80 年代中期至今,客户端/服务器(C/S)体系结构、特别是 Web 技术和网络分布式对象技术飞速发展,导致软件体系结构向更加灵活的多层分布式结构演变,CORBA、EJB、COM/DCOM 等三大分布式的对象模型技术相继出现。2006 年,面向服务架构(Service-Oriented Architecture, SOA)作为下一代软件架构,作为“抽象、松散耦合和粗粒度”的软件架构,根据需求通过网络对松散耦合的粗粒度应用组件进行分布式部署、组合和使用,主要用于解决传统对象模型中无法解决的异构和耦合问题。

至此,软件发展经历了 Mainframe 结构、Client/Server 结构、B/S 多层分布式结构、SOA 的演变过程,整个软件系统变得越来越分散、越来越开放、越来越强调互操作性。

1.1.2 项目特性

项目是为完成某项独特的产品、服务或成果所做的临时性努力。项目具有以下特征:

(1) 项目具有明确的目标。项目的目标就是完成某一产品、服务或预期成果,而且在定义项目目标时通常带有进度和成本的限制。例如,某一项目的目标是:“在 6 个月内,以 2 万元的成本开发完成学校网络教学平台”。

(2) 项目具有临时性。临时性是指每一个项目都有开始和结束时间。当项目的目标已经达到,或由于各种原因项目不需要再持续下去时,项目即达到了它的终点,项目团队也会解散。任何项目的期限都是有限的,项目不是持续不断的努力。

(3) 项目具有独特性,也称一次性。不同于那种重复性的日常工作,项目创造独特的产品、服务或成果。例如,设计和建造“国家歌剧院”是一个项目,而每天的卫生保洁工作不是项目。

(4) 项目是逐步完善(渐进明细)的。逐步完善意味着分步、连续的积累和逐步的细化。在项目初期,对项目范围、规模、成本、进度的估计和计划都是粗粒度的,随着项目的进展,对这些因素的理解会逐渐地深入和细化。

(5) 项目使用的资源是受到限制的。资源包括人员、设备、材料等,可供一个项目使用的这些资源是有限的。

(6) 项目具有一定程度的不确定性。在一个项目开始时,通常要对项目的进度、成本等进行估计,并据此提出项目目标,制定项目计划。但在项目执行过程中,人员、资金、技术、市场等因素在不断变化,项目可能会遇到各种各样的风险,这会给项目带来一定程度的不确定性,使项目不能完全按照原有计划执行,项目目标也不可能完全达到。

项目作为一种特殊的活动,有效地利用各种资源,通过执行一系列相互联系的任务而

达到一个独特的目标。项目普遍存在于人类社会中,以下是项目的一些例子:

- 开发一个新的产品。
- 设计和实现一个新版的计算机应用系统。
- 一个工厂的现代化改造。
- 建造一座建筑。
- 自然灾害后一个城市的重建。
- 某软件企业的 CMMI3 级认证。
- 举行一次学术研讨会。
- 举办一个一百周年庆典。

1.1.3 软件项目

软件项目是一种特殊的项目,其特殊性表现在它的目标是生产软件产品。软件产品与其他类型的项目产品有很大的差异,Fred Brooks 在他的文章《没有银弹》中,总结了软件的以下特点。

(1) 复杂性。软件实体可能比任何人类创造的其他实体都复杂。软件系统有数量极大的状态,这使得设计、描述和测试软件系统都非常困难;软件中没有任何两个部分是完全相同的,软件系统的扩展也不是相同元素的重复添加,而是不同元素实体的添加,大多数情况下,这些元素之间的交互途径以非线性递增的方式增长,因此整个软件系统的复杂度以更大的非线性级数增长。

(2) 可变性。由于软件是纯粹的逻辑思维的产物,它可以很容易地被改变,可以无限地扩展。而实际上软件也总是处于持续的变更之中,用户需求的改变,运行环境和硬件平台的改变都会强迫软件随之变化。

(3) 不可见性。软件是逻辑实体,不具有空间的形体特征,因此是不可见的和无法可视化的。用图形描述软件会受到很大限制,一种图形只能描述软件某一部分或某一方面的属性,而不能全面形象地描述软件。这种不可见性不仅给软件设计带来困难,也严重阻碍了人员之间的交流。

正由于软件具有以上特点,软件产品的生产比一般产品的生产更难于控制。因此软件项目虽然具有项目的一般特性,但它是一个新的领域,具有以下特点。

(1) 知识密集型,技术含量高。软件项目是知识密集型项目,技术性很强,需要大量高强度的脑力劳动。项目工作十分细致、复杂和容易出错。软件项目不需要使用大量的物质资源,而主要是使用人力资源,因此人员的因素极为重要,项目团队成员的结构、技能、责任心和团队精神对软件项目的成功与否有着决定性的影响。

(2) 涉及多个专业领域,多种技术综合应用。软件项目属于典型的跨学科合作项目,例如开发大型管理信息系统就需要项目成员具有行业的业务知识、数据库技术、程序设计技术和信息安全技术等多专业领域知识。

(3) 项目范围和目标的灵活性。随着项目的进展,客户需求可能会发生变化,从而导致项目范围和目标的变化。软件开发不像其他产品的生产,有着非常具体的标准和检验方法,软件的标准柔性很大,衡量软件是否成功的重要标准就是用户满意度,但用户满意

度这个标准在软件开发前很难精确地、完整地表达出来。

(4) 风险大,收益大。由于技术的高度复杂性和需求等因素的不确定性,软件项目风险控制难度较大,项目的成功率较低,但是一旦某个软件产品获得成功,将会带来相对高额的回报。

(5) 客户化程度高。项目的独特性在软件领域表现得更为突出,不同的软件项目之间差别较大。软件开发商往往要根据客户的具体要求提供独特的解决方案,即使有现成的解决方案,通常也需要进行一定的客户化工作。

(6) 过程管理的重要性。软件项目需要对整个项目过程进行严格和科学的管理,尤其是对大型、复杂的软件项目。“质量产生于过程”,必须监控软件开发的过程和中间结果。没有严格的过程管理,开发人员的个人能力再强也没有用。

目前,软件项目的开发和运作远远没有其他领域的项目规范,很多的理论还不能适应所有的软件项目,经验在软件项目中仍起很大的作用。

1.1.4 项目生命周期

在实施项目管理的过程中,将项目划分为便于进行管理的一系列阶段,并在不同的阶段执行一系列具体的项目管理活动。项目生命周期的长度根据项目的不同有很大差别,可能只有几个星期,也可能长达数年。在项目生命周期的各个阶段,人力、资源和费用的投入是不平均的,项目开始时投入的资源等比较低,然后逐渐升高,在项目的实施阶段,达到最高峰,此后逐渐下降,直到项目的终止。

软件项目的生命周期通常可划分为4个阶段:识别需求、方案设计、项目执行和项目收尾,如图1.1所示。

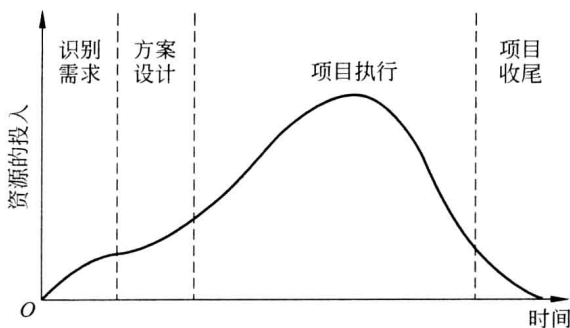


图 1.1 项目的生命周期

1. “识别需求”阶段

由客户提出需求,项目组与客户商议后确定需解决的问题。这里的“客户”是指为完成项目而提供资金的人或组织。这些被客户识别出的需求或问题通常被记录下来,提交给相关的人员或组织去解决,客户在提出需求时往往会附加时间和成本限制。