



普通高等教育“十一五”国家级规划教材

软件测试与质量保证

袁玉宇 编著



北京邮电大学出版社
www.buptpress.com



普通高等教育“十一五”国家级规划教材

软件测试与质量保证

袁玉宇 编著

北京邮电大学出版社

• 北京 •

内 容 提 要

本书的主题是软件质量的改进,重点讨论的是如何提高软件质量的方法。本书提供了两种提高软件质量的技术,一是软件测试,二是软件质量保证。首先对软件质量属性进行了分析,全面论述了软件测试的基本原理和软件过程,讲解了软件测试用例设计方法,以及其在单元测试、集成测试、系统测试和验收测试中的应用。阐述了自动化测试的概念、技术和工具。本书还对软件质量保证从概念、过程和工具方面进行了详细介绍。

本书可以用做软件测试人员、软件质量保证人员、软件开发人员和需要了解软件质量知识的各级软件管理人员的工作参考书,也可以作为计算机专业高年级本科生和研究生的教学参考书。

图书在版编目(CIP)数据

软件测试与质量保证/袁玉宇编著. —北京:北京邮电大学出版社,2008.4

ISBN 978-7-5635-1596-7

I. 软… II. 袁… III. ①软件—测试 ②软件质量—质量管理 IV. TP311.5

中国版本图书馆 CIP 数据核字(2008)第 033915 号

书 名: 软件测试与质量保证

作 者: 袁玉宇

责任编辑: 张佳音

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号(邮编:100876)

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京源海印刷有限责任公司

开 本: 787 mm×960 mm 1/16

印 张: 17.75

字 数: 388 千字

印 数: 1—5 000 册

版 次: 2008 年 6 月第 1 版 2008 年 6 月第 1 次印刷

ISBN 978-7-5635-1596-7

定 价: 32.00 元

· 如有印装质量问题,请与北京邮电大学出版社发行部联系 ·

前 言

随着经济全球化和信息技术的快速发展,软件产业作为国民经济发展的支柱产业之一,在国民经济中发挥着越来越重要的作用。软件成为与人们日常生活息息相关的一部分,软件产品的质量是软件企业的生命,软件测试与质量保证工作伴随着软件的产生而产生。软件危机的频繁出现,使软件测试与质量保证的地位得到了前所未有的提高。软件测试与质量保证不仅仅局限于软件开发过程中的一个阶段,它已经开始贯穿于整个软件开发过程,成为软件产品质量控制和质量管理的重要手段。

编写本书的目的是引导读者通过基础知识和必要技能的学习,掌握改进软件质量的各种技术和方法。软件质量的提高,不是通常想象的那样,简单操作或使用一下软件就能发现软件中的缺陷,其实它包含了大量的科学和工程的技术和方法,也有很多的乐趣。

本书共分五篇。第一篇介绍了软件测试与质量保证是提高软件质量的重要手段,软件测试与质量保证的概念相对于软件质量而存在,所以在学习什么是软件测试与质量保证之前,首先要了解什么是软件质量;第二篇讲述了软件测试用例设计的基本方法,分别介绍了黑盒测试用例设计技术和白盒测试用例设计技术;第三篇将第二篇学到的技术应用到实际的软件测试工作中;第四篇是测试技术的提高部分,其中涉及测试工具的自动化测试的相关问题;第五篇介绍了软件质量保证部分。

非常感谢教育部高教司对此书的支持,感谢每一位参与编写工作的学生,他们是:张畅畅、郭新伟、胡宇、汪凡、庞浩、楼丽种憬、杨金翠、李英华、张丽华、屈萍萍、高微、李琳娜、巴迪曼;还要感谢7年来选修我开设的《软件测试技术》这门课程的同学,因为他们的鼓励和期待才使我有信心和决心完成这本书。

本书适合于以下类型的读者：软件工程研究生和本科高年级学生；在企业中从事软件质量管理、测试管理、测试用例设计和执行人员、测试结果分析和报告人员、质量保证人员；想对软件质量相关知识增进了解的程序员、软件项目经理、软件开发团队的其他人员。

由于时间仓促，书中难免存在一些错误和纰漏，望广大读者谅解并不吝予以指正。如果读者有问题和异议，请将您的意见告诉我，最好通过电子邮件 yuanyuyu@263.net、yuanyuyu@email.buptsse.cn，或者是登陆网站 www.sqaclub.com 发表意见。我不能承诺立即对所有邮件和帖子回复，但是我会尽我所能尽快回复。

作者于北京

目 录

第一篇 基础篇

第 1 章 软件质量的概念	3
1.1 软件质量的概念	3
1.1.1 质量的概念	3
1.1.2 软件质量的内涵	4
1.2 软件质量模型	5
1.2.1 McCall 模型	5
1.2.2 Boehm 质量模型	5
1.2.3 ISO/IEC9126 质量模型	7
1.3 软件缺陷	8
1.3.1 软件缺陷的定义	8
1.3.2 软件缺陷产生原因	9
1.3.3 软件缺陷的分类	11
第 2 章 软件测试的基本概念	13
2.1 软件测试的概念	13
2.1.1 软件测试的定义	13
2.1.2 软件测试的目的	14
2.2 软件测试的分类	14
2.2.1 按测试技术分类	14
2.2.2 按测试方式分类	15
2.2.3 按测试阶段分类	15
2.2.4 按测试实施组织分类	16
2.3 软件测试的最佳实践	16

2.3.1	尽量由独立的测试人员进行测试	17
2.3.2	关键是注重测试用例的设计	17
2.3.3	测试中的集群现象应当被充分的重视	17
2.3.4	完全的测试是不可能的	17
2.3.5	修复缺陷后,一定要进行回归测试	17
第3章	软件测试风险管理	19
3.1	测试风险的基本概念	19
3.1.1	技术风险	20
3.1.2	管理风险	20
3.2	测试风险识别技术	20
3.2.1	头脑风暴法	21
3.2.2	访谈	21
3.2.3	风险检查表	22
3.3	测试风险分析	22
3.3.1	确定测试范围的功能点和性能属性	23
3.3.2	确定测试风险发生的可能性	23
3.3.3	确定测试风险发生后产生的影响程度	24
3.3.4	计算测试风险优先级	25
3.3.5	确定测试风险优先级	25
3.4	测试计划风险的应对措施	27
第4章	测试过程概述	29
4.1	常见测试过程模型	29
4.1.1	V测试过程模型	29
4.1.2	W模型	30
4.1.3	H模型	31
4.2	软件测试过程的体系结构	32
4.2.1	软件测试人员	32
4.2.2	测试过程的活动分解	33
4.2.3	测试过程的工作产品	34

4.3	测试计划	36
4.4	测试用例的获取	39
4.4.1	测试设计规格说明	40
4.4.2	测试用例规格说明	41
4.4.3	测试规程规格说明	42
4.5	执行测试	45
4.5.1	测试日志	46
4.5.2	测试意外事件报告	46
4.5.3	测试总结报告	48
4.6	测试有效性的度量	50
4.6.1	客户满意度度量	50
4.6.2	缺陷度量	51
4.6.3	覆盖度量	54

第二篇 技术篇

第5章	黑盒测试用例设计技术	59
5.1	等价类划分法	59
5.2	边界条件法	61
5.3	决策表法	70
5.4	因果图法	74
5.5	正交表测试法	77
5.6	黑盒测试的其他方法	84
第6章	白盒测试用例设计技术	88
6.1	静态白盒法	88
6.2	侵入式法	90
6.3	路径覆盖法	92
6.4	基本路径法	98
6.5	定义/使用法	102
6.6	程序片法	106

6.7 白盒测试的其他方法 109

第三篇 应用篇

第7章 单元测试 117

7.1 单元测试概念及目的 117

 7.1.1 单元测试定义 117

 7.1.2 单元测试目的 117

7.2 单元测试内容 119

 7.2.1 接口 119

 7.2.2 局部数据结构 119

 7.2.3 边界条件 119

 7.2.4 独立路径 120

 7.2.5 错误处理路径 120

7.3 单元测试策略 120

 7.3.1 由顶向下的单元测试策略 121

 7.3.2 由底向上的单元测试策略 121

 7.3.3 孤立的单元测试策略 121

7.4 单元测试关键实践 122

 7.4.1 单元测试人员 122

 7.4.2 测试时间 123

 7.4.3 测试过程 123

第8章 集成测试 125

8.1 集成测试概念 125

 8.1.1 集成测试的定义 125

 8.1.2 集成测试的必要性 126

 8.1.3 常见的集成测试故障 126

8.2 集成测试策略 127

 8.2.1 大爆炸集成 127

 8.2.2 自顶向下集成 129

 8.2.3 自底向上集成 130

 8.2.4 三明治集成 131

 8.2.5 协作集成 132

8.2.6 高频集成	132
第9章 系统测试	135
9.1 系统测试概念	135
9.1.1 系统测试的定义	135
9.1.2 系统测试与单元测试、集成测试的区别	135
9.2 系统数据测试	136
9.3 系统端口事件测试	138
9.4 系统测试类型	140
第10章 验收测试	143
10.1 验收测试概念	143
10.2 Alpha 测试	144
10.3 Beta 测试	145
第四篇 测试自动化	
第11章 测试自动化的基本概念	153
11.1 测试自动化的定义	153
11.1.1 概述	153
11.1.2 自动化测试的优点	154
11.1.3 自动化测试的局限性	155
11.2 测试自动化的适用范围	156
11.2.1 不适合自动化测试情况	156
11.2.2 合适自动化测试的情况	157
第12章 测试自动化的技术	159
12.1 录制/回放技术	159
12.2 脚本技术	160
12.2.1 线性脚本	161
12.2.2 结构化脚本	162
12.2.3 共享脚本	163
12.2.4 数据驱动脚本	166
12.2.5 关键字驱动脚本	169
12.3 自动比较	170

12.3.1	自动比较的定义	170
12.3.2	自动比较的原因和内容	171
12.3.3	动态比较	172
12.3.4	执行后比较	173
12.3.5	比较准则	177
12.4	自动化前后处理	179
12.4.1	前处理和后处理	179
12.4.2	不同阶段的前后处理	182
12.4.3	执行中的问题	184
第 13 章	测试自动化工具	187
13.1	测试工具类型	187
13.2	基本测试工具	188
13.2.1	配置管理工具	189
13.2.2	缺陷跟踪工具	191
13.2.3	监控工具	192
13.2.4	功能测试工具	193
13.2.5	性能测试工具	194
13.3	测试自动化工具产品简介	197
13.3.1	IBM Rational	197
13.3.2	HP Mercury Interactive	199
13.3.3	Compuware	201
13.3.4	Borland Segue	202
13.3.5	其他	202

第五篇 质量保证篇

第 14 章	软件质量保证概要	207
14.1	软件质量保证相关概念	207
14.1.1	质量保证和软件质量保证	207
14.1.2	质量控制和软件质量控制	209
14.1.3	全面质量管理	210
14.2	软件质量保证的目标和任务	211
14.2.1	软件质量保证的目标	211

14.2.2 软件质量保证的任务	211
14.3 软件质量保证活动及实施	212
14.3.1 软件质量保证活动	212
14.3.2 软件质量保证的实施	213
14.4 软件质量保证体系架构	214
第 15 章 软件质量保证过程	216
15.1 计划阶段	216
15.2 需求分析阶段	219
15.3 设计阶段	222
15.4 编码阶段	225
15.5 测试阶段	230
15.6 系统交付和安装阶段	233
第 16 章 软件质量保证工具	235
16.1 规程与工作条例	235
16.1.1 规程与工作条例的概念	235
16.1.2 规程与工作条例的作用	236
16.1.3 规程与规程手册	236
16.1.4 工作条例与工作条例手册	237
16.1.5 规程与工作条例的编制、执行与更新	238
16.2 模板	239
16.2.1 模板的概念	239
16.2.2 模板的作用	240
16.2.3 模板的编制	241
16.2.4 模板样例	242
16.3 检查表	244
16.3.1 检查表的概念	244
16.3.2 检查表的作用	247
16.3.3 编制、执行和更新检查表的组织框架	248
16.3.4 设计与使用检查表的注意事项	249
16.4 配置管理	250
16.4.1 管理者的承诺	252
16.4.2 SCM 人员组成	253
16.4.3 建立 CCB	255

16.4.4	验收测试中的 SCM	257
16.4.5	SCM 的必要性	257
16.4.6	测试周期中的 CCB 的角色	257
16.4.7	审计的理由和实践	261
16.4.8	避免大量的“文书工作”	263
16.4.9	在 SCM 活动间分配资源	265
16.5	受控文档与质量记录	267
16.5.1	受控文档和质量记录的定义和目标	267
16.5.2	文档编制控制规程	269
16.5.3	受控文档清单	270
16.5.4	受控文档的编制	270
16.5.5	受控文档的批准问题	271
16.5.6	受控文档的存储与检索问题	271
16.5.7	记录控制	272

第一篇

基础篇

对软件质量和软件测试概念的深入了解,是做好软件测试的根本。只有基本概念清楚了,才能对此概念相关的问题有一个正确的理解和分析,最终解决所面临的问题。

本篇将分为4章介绍软件测试相关概念:第1章主要介绍软件质量的相关概念;第2章主要介绍软件测试的相关概念;第3章讲述软件测试风险管理;第4章描述软件测试过程。

第 1 章 软件质量的概念

软件测试是当前软件工程学科的重要组成部分。在实际的软件开发过程中,软件测试的重要性已经被广泛地认同。软件测试是提高软件质量的重要手段,软件测试的概念相对于软件质量而存在,所以在学习什么是软件测试之前,首先要了解什么是软件质量。

作为本书的开篇,本章重点对软件质量的基本概念、软件质量模型以及软件缺陷进行介绍。

1.1 软件质量的概念

1.1.1 质量的概念

“质量(Quality)”这个词,如果单从汉语文字来看,是由“质”和“量”两个词构成的,字面上理解就是在质和量上的程度。然而要想给质量下个明确的定义却并不容易。先来看看一些权威机构对质量做出的解释。

在《辞海》中,对质量的解释是产品或工作的优劣程度。

1986年ISO8492给出的质量的定义是:质量是产品或服务满足明示或暗示需求能力的特性和特征的集合。

IEEE在“Standard Glossary of Software Engineering Terminology”中给出的质量定义是被普遍接受的概念,即质量是系统、部件或过程满足明确需求。

世界著名的质量管理专家朱兰对“质量”给出的含义:满足使用要求的基础是质量特征,产品的任何特性(性质、属性等)、材料或满足使用要求的过程都是质量特征。

从众多的定义中,我们可以看到,质量是一个复杂的多层面概念,如果站在不同的观点上从不同的层面或角度对质量就有着不同的理解。

- 先验证观点:质量是产品的一种可以认识但不可定义的性质。
- 用户观点:质量是产品满足使用目的的程度。
- 制造者观点:质量是产品性能符合规格要求的程度。
- 产品观点:质量是联结产品固有性质的纽带。
- 基于价值观点:质量依赖于顾客愿意付给产品报酬的数量。

因此,有一个很重要的概念和质量息息相关,这个概念就是“客户”,不同的客户对待质量的看法是不同的,质量和客户两者相对而存在。

客户的定义至少存在两个范畴——内部的和外部的:

外部客户是产品的实际使用者或服务的对象,是传统意义上大家所认可的客户。

内部客户是更为广泛意义上的客户,客户可以被理解为下一道工序的接受者。在软件生产的环节中有关的人员都可被定义为这一类型的客户,软件的设计者是需求分析人员的客户,编程人员是设计者的客户,软件测试是编程人员的客户。

从质量的定义和不同的理解中均可以看到,质量是满足客户需求的特征这个核心含义,这样对质量的解释和说明就存在困难,传统的理性观点把世界分为主观和客观两部分,但是质量似乎被排除在这种区分之外,既不是客观的,也不是主观的。质量似乎不是客观的,因为没有什么科学仪器可以直接测出质量来;质量似乎也不是主观的,因为它不仅存在于人们的脑海中。其实,质量应该是客观存在的,但是测度它的方法却是主观的。

1.1.2 软件质量的内涵

关于软件质量有许多好的定义。通过审视每个定义,可以正确理解什么是软件质量。以下从一个较为抽象的定义逐步转向更具体的定义,这有助于对该问题的理解。

- Fisher 和 Light 在《Definitions in Software Quality Management》中的质量定义:(表征)计算机系统卓越程度的所有属性的集合。“所有属性的集合”包括可靠性、可维护性、可用性等。“卓越”则属于软件质量的定义范畴。

- 在 Donald Reifer 的《State of the Art in Software Quality Management》一书,有如下定义:软件产品满足明示需求程度的一组属性的集合。这个定义中继续沿用“属性集合”的说法,但增加了满足明示需求的成分。

- 在《Software Quality Assurance and Measurement; a Worldwide Perspective》中除了关注“明示需求”之外,还扩展到了“暗示”需求:软件产品满足明示或暗示需求能力的特性和特征的集合。

- Stephen Kan 在《Metrics and Models in Software Quality Engineering》中对“需求”这个层面更加明确:在质量定义中客户的角色必须明确指出,即满足客户的需求。

这一定义与 Philip B. Crosby《Quality Is Free》中“满足需求”的定义非常接近,只不过这里是“满足软件需求”。这个定义面临这样一个问题:什么是软件需求?难道软件需求仅是决定软件做什么的技术需求?还是也包括软件质量的需求?