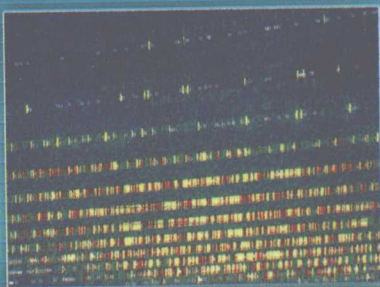




普通高等教育“十一五”规划教材  
高等院校计算机技术系列教材

# 软件质量 保证与测试

刘怀亮 相洪贵 编著



冶金工业出版社

普通高等教育“十一五”规划教材  
高等院校计算机技术系列教材

# 软件质量保证与测试

刘怀亮 相洪贵 编著

北 京

冶金工业出版社

## 内 容 简 介 “十一五” 普通高等教育

本书是根据普通高等教育“十一五”规划教材的指导精神而编写的。

本书介绍了软件质量保证和软件测试的一些基本概念、技术和方法，涵盖了业界出现的大多数测试领域的知识。主要内容有：软件质量保证概述，软件质量标准及软件质量保证的标准化，软件质量的成本，软件质量保证的实施，软件的质量设计，软件质量保证的度量，软件测试概述，单元测试，集成测试，系统测试，系统集成测试，回归测试，验收测试，测试度量与环境配置，测试工具和测试用例设计。

本书既可作为高等院校本、专科计算机及其相关专业的教材，也可作为从事计算机软件开发人员的参考书。

### 图书在版编目 (CIP) 数据

软件质量保证与测试 / 刘怀亮, 相洪贵编著. —北京:  
冶金工业出版社, 2007.4

普通高等教育“十一五”规划教材

ISBN 978-7-5024-4260-6

I. 软... II. ①刘...②相... III. 软件质量—高等学校—  
教材 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2007) 第 044731 号

出版人 曹胜利 (北京沙滩嵩祝院北巷 39 号, 邮编 100009)

责任编辑 肖放

ISBN 978-7-5024-4260-6

广州锦昌印务有限公司印刷; 冶金工业出版社发行; 各地新华书店经销

2007 年 4 月第 1 版第 1 次印刷

787mm×1092mm 1/16; 16 印张; 365 千字; 246 页

25.00 元

冶金工业出版社发行部 电话: (010) 64044283 传真: (010) 64027893

冶金书店 地址: 北京东四西大街 46 号 (100711) 电话: (010) 65289081

(本社图书如有印装质量问题, 本社发行部负责退换)

# 前 言

## 一、关于本书

本书是根据普通高等教育“十一五”规划教材的指导精神而编写的。计算机工业技术在发达国家曾经走了不少弯路，软件技术曾经饱受风霜，“软件危机”至今仍然困扰着 IT 工程师们。开发出健壮性强的优秀软件的速度远远落后于计算机硬件的发展和社会对大型软件的需求。

为了摆脱软件危机的困扰，以研究软件开发与维护的工程学科在 60 年代末开始迅速发展起来，这就是软件工程。软件工程已经成为计算机科学的一个重要的分支、一个异常活跃的研究领域，软件质量保证和测试技术更是其中的一个重要的环节。

软件质量保证与测试作为一门软件工程方面的学科，具有一定的程序设计经验和分析设计经验，掌握一门编程语言，如：C++/Java 语言，并且了解一定的软件开发知识对本书的理解是十分必要的。

本书的重点在于介绍软件质量保证和软件测试的一些基本概念、技术和方法，涵盖了业界出现的大多数测试领域内的知识。

## 二、本书结构

全书共 17 章，结构如下：

第 1 章：软件质量保证概述。介绍了软件质量保证的相关概念及其之间的联系与区别；软件质量保证的主要任务，软件质量保证的重要性等。

第 2 章：软件质量标准及软件质量保证的标准化。介绍了软件质量保证标准的分类和 MIL-STD-498、IEEE-STD-1074、ISO/IEC-12207 以及它们之间的对比。

第 3 章：软件质量的成本。介绍了软件质量的概念及输入、输出等，熟悉并掌握如何对软件质量成本的度量。

第 4 章：前期软件质量保证的技术——审查。介绍了审查的过程，掌握需求审查、规约审查、文档审查的步骤。

第 5 章：软件质量保证的实施。介绍了建立软件配置管理时要考虑的问题，Pareto 原理；能力成熟度模型及 SQA 在软件开发成熟度中的应用。

第 6 章：软件的质量设计。介绍了软件质量模型设计方法、软件计划和质量计划的规划，CASE 工具的使用，CASE 工具进行高质量的软件开发的方法。

第 7 章：软件质量保证的度量。介绍了软件质量保证的度量方法及其应用，了解软件质量保证的评价。

第 8 章：软件测试概述。介绍了软件测试的各种技术，如何管理和计划测试。

第 9 章：单元测试。介绍了掌握单元测试的方法，单元测试的数据要求，单元测试的测试技术，单元测试的输入和输出等。

第 10 章：集成测试。介绍了集成测试的方法，集成测试的技术，集成测试的数据要求，集成测试的输入和输出。

第 11 章：系统测试。介绍了系统测试的方法，系统测试的技术，系统测试的数据要求，系统测试的输入、输出。

第 12 章：系统集成测试。介绍了系统集成测试的方法，系统集成测试技术，系统集成测试的数据要求，系统集成测试的输入、输出。

第 13 章：回归测试。介绍了回归测试的方法，回归测试的数据要求，回归测试的技术，回归测试的输入、输出。

第 14 章：验收测试。介绍了用户验收测试的方法，用户验收测试的数据要求，用户验收测试的技术，操作验收测试的方法，操作验收测试的数据要求，操作验收测试的技术，验收测试的输入、输出。

第 15 章：测试度量与环境配置。介绍了衡量软件测试的方法，以及测试环境的配置，系统测试的方法和测试技术，通过使用好的测试技术来提高软件的质量。

第 16 章：测试工具。介绍了测试工具的使用方法，运用测试工具进行实际软件的测试。

第 17 章：测试用例设计。介绍了黑盒测试、白盒测试相关技术和测试用例。

### 三、本书特点

本书主要有如下特点：

(1) 内容的覆盖面广。综合考虑了软件质量保证和测试的方方面面。

(2) 理论与实践相结合。本书不是一本纯理论方面的书，书中很多涉及到理论的内容多是通过实践经验的方式来阐述的，这样使得读者更容易理解其理论知识和体会实践的过程。

(3) 强调了全面的质量保证。无论是单元测试、集成测试、还是系统测试，都是讲究如何进行测试数据分析、测试用例设计的。

(4) 将软件质量保证与软件测试紧密地结合在一起。

### 四、本书适用对象

本书既可作为高等院校本、专科计算机及其相关专业的教材，也可作为从事计算机软件开发人员的参考书。

在本书的编写过程中得到智能软件开发团队研发人员、工程研发中心多方面的帮助和支持，在此表示衷心感谢。

由于作者水平有限，编写时间仓促，书中的疏漏和不足之处在所难免，恳请读者批评指正。欢迎广大读者、专家和同仁对本书提出宝贵意见，联系方式如下：

电子邮箱：[service@cnbook.net](mailto:service@cnbook.net) 作者邮箱：[great\\_liu@126.com](mailto:great_liu@126.com)

网址：[www.cnbook.net](http://www.cnbook.net)

本书电子教案及习题参考答案可在该网站下载，此外，该网站还有一些其他相关书籍的介绍，可以方便读者选购参考。

编者

2007年3月

# 目 录

<b>第 1 章 软件质量保证概述</b> .....	1
1.1 软件质量保证.....	1
1.1.1 软件.....	1
1.1.2 软件错误.....	3
1.1.3 软件危机.....	5
1.1.4 质量.....	7
1.1.5 软件质量.....	7
1.1.6 软件质量保证.....	11
1.2 软件质量保证的主要任务.....	13
1.3 实例.....	14
小结.....	14
综合练习一.....	16
一、选择题.....	16
二、填空题.....	17
三、简答题.....	17
四、上机题.....	17
<b>第 2 章 软件质量标准及软件质量保证的标准化</b> .....	18
2.1 软件质量标准.....	18
2.2 软件质量保证的标准化.....	19
2.2.1 MIL-STD-498.....	21
2.2.2 IEEE-STD-1074.....	22
2.2.3 ISO/IEC-12207.....	23
2.2.4 498/CMM.....	27
2.2.5 498/12207.....	27
小结.....	28
综合练习二.....	30
一、选择题.....	30
二、填空题.....	31
三、简答题.....	31
四、上机题.....	31
<b>第 3 章 软件质量的成本</b> .....	32

3.1 软件质量的概念.....	32
3.2 输入.....	34
3.3 输出.....	35
3.4 质量的总成本.....	36
3.5 生产率.....	37
3.6 实例.....	38
小结.....	38
综合练习三.....	39
一、选择题.....	39
二、填空题.....	40
三、简答题.....	40
四、上机题.....	40
<b>第 4 章 前期软件质量保证的技术——审查</b> .....	41
4.1 审查过程.....	41
4.1.1 主要目标.....	41
4.1.2 辅助目标.....	41
4.1.3 审查阶段.....	42
4.1.4 审查的类型.....	44
4.1.5 审查缺陷类型与定义.....	46
4.1.6 审查初始化.....	46
4.1.7 审查前提.....	47
4.1.8 审查组.....	47
4.2 吸取的教训.....	50
4.2.1 心理学因素.....	50
4.2.2 审查要客观.....	51
4.2.3 审查者的限制.....	51
4.2.4 数据的收集要一致.....	51
4.2.5 实践.....	51
小结.....	53
综合练习四.....	53
一、选择题.....	53

二、填空题 .....	53	6.2.4 增量模型 (Incremental Model) ..	80
三、简答题 .....	54	6.2.5 螺旋模型 (Spiral Model) .....	81
四、上机题 .....	54	6.2.6 喷泉模型 (fountain model, (面向对象的生存期模型, OO 模型)) .....	82
<b>第 5 章 软件质量保证的实施 .....</b>	<b>55</b>	6.2.7 演化模型 (incremental model) ..	82
5.1 软件配置管理 .....	55	6.2.8 面向模型的变换型软件开发方法 (Model Oriented Specification and Transformation) .....	82
5.1.1 软件配置管理现实考虑 .....	55	6.3 开发计划和质量计划 .....	84
5.1.2 SCM 人员的组成 .....	56	6.3.1 开发计划的要素 .....	84
5.1.3 建立 CCB .....	58	6.3.2 质量计划编制 .....	86
5.2 Pareto 原理 .....	59	6.4 CASE 工具介绍 .....	87
5.2.1 Pareto 的历史背景 .....	59	6.4.1 什么是 CASE 工具 .....	88
5.2.2 经典实例 .....	60	6.4.2 与 CASE 相关的问题 .....	89
5.3 能力成熟度模型 .....	62	6.4.3 CASE 工具对软件产品质量 的作用 .....	90
5.3.1 SEI 过程能力成熟度模型 .....	62	6.4.4 在 SQA 中应用 CASE 工具 .....	91
5.3.2 SPR 评价 .....	65	小结 .....	94
5.4 SQA 在软件开发成熟度中的应用 .....	66	综合练习六 .....	94
5.4.1 过程的认识 .....	66	一、选择题 .....	94
5.4.2 生产线的隐喻 .....	67	二、填空题 .....	95
5.4.3 典型的 SQA 计划内容 .....	69	三、简答题 .....	96
5.5 实践 .....	72	四、上机题 .....	96
小结 .....	72	<b>第 7 章 软件质量保证的度量 .....</b>	<b>97</b>
综合练习五 .....	73	7.1 软件质量度量 .....	97
一、选择题 .....	73	7.1.1 软件过程质量度量 .....	98
二、填空题 .....	73	7.1.2 产品质量度量 .....	99
三、简答题 .....	73	7.1.3 软件维护的度量 .....	101
四、上机题 .....	74	7.1.4 软件质量度量的指标 .....	103
<b>第 6 章 软件的质量设计 .....</b>	<b>75</b>	7.2 维护过程 .....	103
6.1 经典软件开发方法学和其他软件 开发方法学 .....	75	7.2.1 维护组织 .....	103
6.1.1 经典的软件开发方法学 .....	75	7.2.2 维护报告 .....	104
6.1.2 其他软件开发方法学 .....	76	7.2.3 保存维护纪录 .....	104
6.2 软件开发模型 .....	77	7.2.4 评价维护活动 .....	105
6.2.1 边做边改模型 (Build-and-Fix Model) .....	77	7.3 可维护性 .....	105
6.2.2 瀑布模型 .....	78		
6.2.3 快速原型模型 (Rapid Prototype Model) .....	80		

7.4 应用实施 .....	106	三、简答题 .....	122
7.4.1 建立有效的软件度量过程 .....	106	四、上机题 .....	122
7.4.2 弗罗哈克的实用软件度量原则 .....	108	<b>第9章 单元测试 .....</b>	<b>123</b>
小结 .....	108	9.1 单元测试的方法 .....	123
综合练习七 .....	108	9.1.1 概述 .....	123
一、选择题 .....	108	9.1.2 测试的内容 .....	124
二、填空题 .....	109	9.1.3 单元测试过程 .....	126
三、简答题 .....	109	9.2 单元测试的数据要求 .....	127
四、上机题 .....	109	9.3 单元测试的测试技术 .....	127
<b>第8章 软件测试概述 .....</b>	<b>110</b>	9.4 输入、输出 .....	128
8.1 测试技术介绍 .....	110	9.4.1 单元测试的输入 .....	128
8.1.1 软件测试的背景 .....	110	9.4.2 单元测试的输出 .....	128
8.1.2 软件可靠性问题 .....	111	9.5 设计单元测试说明 .....	128
8.1.3 软件缺陷与故障 .....	111	9.5.1 测试用例设计步骤 .....	129
8.1.4 测试的概念 .....	112	9.5.2 用例设计的一般原则 .....	130
8.1.5 软件测试的对象 .....	113	9.6 单元测试实践 .....	130
8.1.6 测试必要性 .....	114	小结 .....	131
8.2 软件测试基础理论 .....	114	综合练习九 .....	131
8.2.1 软件测试基本问题 .....	114	一、选择题 .....	131
8.2.2 软件测试的关键问题 .....	114	二、填空题 .....	132
8.2.3 软件测试的基本理论 .....	115	三、简答题 .....	132
8.3 管理和计划测试 .....	116	四、上机题 .....	132
8.3.1 测试与开发各个阶段的关系 .....	116	<b>第10章 集成测试 .....</b>	<b>133</b>
8.3.2 软件测试的管理和规划 .....	117	10.1 集成测试的概述 .....	133
8.4 软件测试过程 .....	117	10.2 集成测试的必要性及其过程 .....	133
8.4.1 软件开发过程及测试在开发 阶段的作用 .....	118	10.2.1 计划阶段 .....	133
8.4.2 制定测试计划 .....	118	10.2.2 设计阶段 .....	134
8.4.3 测试执行过程 .....	119	10.2.3 实现阶段 .....	134
8.4.4 执行阶段 .....	134	10.2.4 执行阶段 .....	134
8.5 实战——测试计划 .....	120	10.3 集成测试方法 .....	135
8.5.1 编写测试计划的目的 .....	120	10.4 集成测试分析 .....	139
8.5.2 计划的纲要 .....	120	10.4.1 模块分析 .....	139
小结 .....	121	10.4.2 体系结构分析 .....	140
综合练习八 .....	122	10.4.3 风险的分析 .....	140
一、选择题 .....	122	10.4.4 接口分析 .....	140
二、填空题 .....	122	10.4.5 集成测试策略分析 .....	142



10.4.6	可测试性分析	142	12.4.5	编码过程	163
10.4.7	集成测试的数据要求	142	12.4.6	Bug 报告过程	165
10.5	集成用例设计	142	12.5	输入、输出	165
10.6	输入、输出	142	12.5.1	输入	166
10.6.1	集成测试的输入	142	12.5.2	输出	166
10.6.2	集成测试的输出	143	12.6	实践	166
10.7	实践	143	小结		167
小结		144	综合练习十二		167
综合练习十		144	一、选择题		167
一、选择题		144	二、填空题		167
二、填空题		144	三、简答题		167
三、简答题		144	四、上机题		168
四、上机题		145	<b>第 13 章 回归测试</b>		<b>169</b>
<b>第 11 章 系统测试</b>		<b>146</b>	13.1	概述	169
11.1	系统测试的内容	146	13.2	回归测试的方法	169
11.2	系统测试步骤	147	13.2.1	如何维护测试用例库	170
11.3	系统测试的方法和技术	148	13.2.2	如何选择回归测试包	170
11.3.1	进行系统测试应遵循的方法	148	13.2.3	回归测试的基本过程	171
11.3.2	系统测试过程	153	13.3	回归测试的数据要求	172
11.4	系统测试人员	155	13.4	回归测试的技术	172
11.5	输入、输出	155	13.5	回归测试的输入、输出	172
小结		155	13.6	面向对象的回归测试	173
综合练习十一		155	13.6.1	对象系统中的依赖	173
一、选择题		155	13.6.2	方法的测试依赖	174
二、填空题		156	13.6.3	面向对象软件回归测试	175
三、简答题		156	13.7	回归测试实践	176
四、上机题		156	小结		177
<b>第 12 章 系统集成测试</b>		<b>157</b>	综合练习十三		177
12.1	系统集成测试的方法	157	一、选择题		177
12.2	系统集成测试技术	157	二、填空题		177
12.3	系统集成测试的数据要求	157	三、简答题		178
12.4	系统集成测试的过程	158	四、上机题		178
12.4.1	构建的确认过程	158	<b>第 14 章 验收测试</b>		<b>179</b>
12.4.2	补丁过程	159	14.1	用户验收测试	179
12.4.3	系统集成测试组提交过程	160	14.1.1	用户验收测试的方法	179
12.4.4	测试用例设计过程	162	14.1.2	用户验收测试的技术	179

14.1.3 输入、输出.....	181	二、填空题.....	202
14.2 操作验收测试.....	181	三、简答题.....	202
14.2.1 操作验收测试的方法.....	181	四、上机题.....	202
14.2.2 操作验收测试的数据要求.....	181	<b>第 16 章 测试工具.....</b>	<b>203</b>
14.2.3 操作验收测试的技术.....	182	16.1 使用目的.....	203
14.2.4 输入、输出.....	182	16.2 种类 (Types of Testing Tool).....	203
14.3 自动化验收测试工具-Selenium.....	182	16.3 目前主流测试工具.....	204
14.3.1 Web 应用程序测试介绍.....	182	16.3.1 单元测试方面:JUnit 工具.....	204
14.3.2 Selenium 介绍.....	183	16.3.2 功能测试工具.....	205
14.3.3 Selenium 命令.....	183	16.3.3 性能/负载/压力测试.....	209
14.3.4 Selenium 模式.....	183	16.3.4 黑盒测试工具.....	212
14.4 实际使用.....	186	16.3.5 白盒测试工具.....	213
14.4.1 安装和启动.....	186	16.3.6 集成测试工具.....	213
14.4.2 用例.....	186	16.3.7 测试管理工具.....	214
14.4.3 软件验收测试工作程序.....	188	16.3.8 测试辅助工具.....	214
小结.....	189	16.4 其他测试工具.....	214
综合练习十四.....	190	16.5 自动化测试.....	219
一、选择题.....	190	16.6 测试工具.....	222
二、填空题.....	190	小结.....	224
三、简答题.....	190	综合练习十六.....	224
四、上机题.....	190	一、选择题.....	224
<b>第 15 章 测试度量与环境配置.....</b>	<b>191</b>	二、填空题.....	224
15.1 测试过程中常用的度量.....	191	三、简答题.....	224
15.2 建立和管理度量.....	192	四、上机题.....	225
15.3 测试环境的配置与管理.....	193	<b>第 17 章 测试用例设计.....</b>	<b>226</b>
15.3.1 配置测试环境.....	193	17.1 软件测试的目标.....	226
15.3.2 管理测试环境.....	196	17.2 黑盒测试与白盒测试.....	226
15.3.3 测试环境与外界真实环境.....	198	17.2.1 黑盒测试.....	226
15.4 软件测试的组织与管理.....	198	17.2.2 白盒测试.....	227
15.4.1 测试的过程及组织.....	198	17.3 测试用例的设计.....	228
15.4.2 测试方法的应用.....	199	17.3.1 逻辑覆盖.....	228
15.4.3 测试的人员组织.....	200	17.3.2 关于控制结构测试的一些 讨论.....	229
15.4.4 软件测试文件.....	200	17.3.3 基本路径测试.....	232
小结.....	201	17.3.4 等价类划分.....	234
综合练习十五.....	201		
一、选择题.....	201		

17.3.5 边界值分析..... 235

17.3.6 因果图..... 235

17.3.7 错误推测法..... 236

17.3.8 测试用例的设计..... 236

17.3.9 测试方法选择的综合策略..... 236

小结..... 237

综合练习十七..... 237

一、选择题..... 237

二、填空题..... 238

三、简答题..... 238

四、上机题..... 239

**附录 软件测试中的常见术语  
中英文对照表..... 240**

18.1 单元测试..... 241

18.2 集成测试..... 242

18.3 系统测试..... 243

18.4 验收测试..... 244

18.5 回归测试..... 245

18.6 性能测试..... 246

18.7 压力测试..... 247

18.8 负载测试..... 248

18.9 可靠性测试..... 249

18.10 可用性测试..... 250

18.11 易用性测试..... 251

18.12 兼容性测试..... 252

18.13 国际化测试..... 253

18.14 安全测试..... 254

18.15 灾难恢复测试..... 255

18.16 灾难恢复计划..... 256

18.17 灾难恢复策略..... 257

18.18 灾难恢复团队..... 258

18.19 灾难恢复演练..... 259

18.20 灾难恢复文档..... 260

18.21 灾难恢复计划模板..... 261

18.22 灾难恢复计划模板..... 262

18.23 灾难恢复计划模板..... 263

18.24 灾难恢复计划模板..... 264

18.25 灾难恢复计划模板..... 265

18.26 灾难恢复计划模板..... 266

18.27 灾难恢复计划模板..... 267

18.28 灾难恢复计划模板..... 268

18.29 灾难恢复计划模板..... 269

18.30 灾难恢复计划模板..... 270

18.31 灾难恢复计划模板..... 271

18.32 灾难恢复计划模板..... 272

18.33 灾难恢复计划模板..... 273

18.34 灾难恢复计划模板..... 274

18.35 灾难恢复计划模板..... 275

18.36 灾难恢复计划模板..... 276

18.37 灾难恢复计划模板..... 277

18.38 灾难恢复计划模板..... 278

18.39 灾难恢复计划模板..... 279

18.40 灾难恢复计划模板..... 280

14.1 Selenium 简介..... 281

14.2 Selenium 命令..... 282

14.3 Selenium 脚本..... 283

14.4 Selenium 应用..... 284

14.5 Selenium 进阶..... 285

14.6 Selenium 进阶..... 286

14.7 Selenium 进阶..... 287

14.8 Selenium 进阶..... 288

14.9 Selenium 进阶..... 289

14.10 Selenium 进阶..... 290

14.11 Selenium 进阶..... 291

14.12 Selenium 进阶..... 292

14.13 Selenium 进阶..... 293

14.14 Selenium 进阶..... 294

14.15 Selenium 进阶..... 295

14.16 Selenium 进阶..... 296

14.17 Selenium 进阶..... 297

14.18 Selenium 进阶..... 298

14.19 Selenium 进阶..... 299

14.20 Selenium 进阶..... 300

14.21 Selenium 进阶..... 301

14.22 Selenium 进阶..... 302

14.23 Selenium 进阶..... 303

14.24 Selenium 进阶..... 304

14.25 Selenium 进阶..... 305

14.26 Selenium 进阶..... 306

14.27 Selenium 进阶..... 307

14.28 Selenium 进阶..... 308

14.29 Selenium 进阶..... 309

14.30 Selenium 进阶..... 310

14.31 Selenium 进阶..... 311

14.32 Selenium 进阶..... 312

14.33 Selenium 进阶..... 313

14.34 Selenium 进阶..... 314

14.35 Selenium 进阶..... 315

14.36 Selenium 进阶..... 316

14.37 Selenium 进阶..... 317

14.38 Selenium 进阶..... 318

14.39 Selenium 进阶..... 319

14.40 Selenium 进阶..... 320

14.41 Selenium 进阶..... 321

14.42 Selenium 进阶..... 322

14.43 Selenium 进阶..... 323

14.44 Selenium 进阶..... 324

14.45 Selenium 进阶..... 325

14.46 Selenium 进阶..... 326

14.47 Selenium 进阶..... 327

14.48 Selenium 进阶..... 328

14.49 Selenium 进阶..... 329

14.50 Selenium 进阶..... 330

14.51 Selenium 进阶..... 331

14.52 Selenium 进阶..... 332

14.53 Selenium 进阶..... 333

14.54 Selenium 进阶..... 334

14.55 Selenium 进阶..... 335

14.56 Selenium 进阶..... 336

14.57 Selenium 进阶..... 337

14.58 Selenium 进阶..... 338

14.59 Selenium 进阶..... 339

14.60 Selenium 进阶..... 340

## 第1章 软件质量保证概述

软件质量保证是进行软件质量评估和度量的一个功能单位，应贯穿于整个开发过程。它是一种系统化的、有计划的行动模式，是为项目或者产品符合已有的技术需求提供充分信任所必需的。它与质量控制有区别，是计划用来评价开发或者制造产品的过程的一组活动。

本章教学目标：

- (1) 了解软件质量保证的相关概念。
- (2) 掌握软件质量、软件质量保证的概念及其之间的联系与区别。
- (3) 掌握软件质量保证的主要任务，了解软件质量保证的重要性。

### 1.1 软件质量保证

如果要提高软件项目的质量，首先就需要明确软件和定义质量等基本概念。由于这些概念在不同的领域甚至在同一领域中有着模棱两可的含义，所以在正式开展工作之前，理清它们的含义是非常有必要的。

#### 1.1.1 软件

当人们谈到软件的时候，通常会和“代码”这个词联系在一起，或者认为软件就是编程语言指令和语句或开发工具指令等组成的程序。显然，这样的理解是相当片面的，它忽略了软件的其他要素。

那么究竟应该怎样正确理解软件的含义呢？IEEE《Software Engineering Terminology Standard》和 ISO 是这样定义软件的：软件是计算机程序、规程以及可能的相关文档和运行计算机系统需要的数据。从这个定义中可以看出软件的4个组成部分，如图1-1所示。

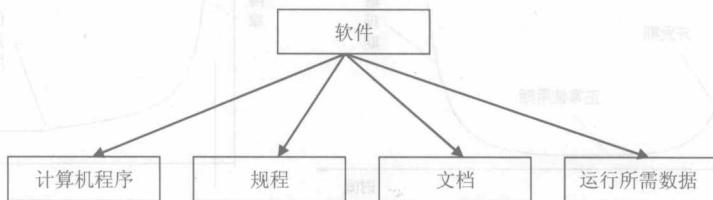


图 1-1 软件的4个组成部分

正是由于这4部分要素的组成，才能保证软件在开发过程和未来的维护服务中的质量：

(1) 计算机程序：也就是常说的代码，它是激活计算机执行所需的应用；是计算机设备可以接受的一系列指令和说明；是用来让计算机执行一个运算或操作。因此，可以看出代码并不等同于软件，它只是软件的一个重要的组成部分。

(2) 规程是软件的一个重要组成部分，它确定了程序的执行流程、进度、方法应用并且负责为应用软件完成所需活动的人。

(3) 文档的重要性是不言而喻的，它对于开发者、用户和维修人员来说是必备的。没有文档的程序就不能称为软件。文档可以规范工作流程、减少沟通成本，从而使得工作

易于检查。各种类型的文档又分为开发文档（需求报告、设计报告、程序描述等）、用户文档（用户手册）和维护文档（程序员手册）。需要注意的是文档永远只是工具而已，要善用文档而不要以为文档就是一切，文档是为了更好地促进软件开发，如果违背了这一原则，仅仅是为了文档而文档，那么文档最终将流于形式，毫无用处。

(4) 软件中所需的数据包括参数、代码和标准测试数据等，是事实、概念或指令的一种结构化表示，并且能够被计算机设备所接受、理解或处理。它使得软件适应特定用户的需要，并且其存在形式可以是外部的（通常是以计算机的可读的形式存在）或内置在计算机设备中。它同样也是软件的不可缺少的重要组成部分。

综上所述，软件指的是计算机指令或数据，可以进行电子存储（来自 Webopedia 对软件的定义），这正是计算机科学家基于软件的本质给出的一种定义。

了解软件的特征，有助于深入理解软件的概念，并最终能真正理解软件的开发过程及其特殊性，有助于开发高质量的符合用户需求的应用软件系统。

软件是一种逻辑性系统元素，不是物理性系统元素，它具有“非实物”属性，软件和硬件不同，设计软件使用的逻辑量要比同样规模的硬件产品多 10~100 倍，为了完成一个复杂的软件系统，需要建立庞大的逻辑体系，其最终的形态表现为计算机程序及其各种文档，融入其中的人们的创造性工作是无形的，只能通过执行计算机程序完成人们预期的功能，达到预期的性能并体现出来。

软件只能设计或实现出来，不是传统意义上的进行制造生产，软件和硬件之间有着根本性的区别。首先，根据系统设计生产出来的硬件产品，其质量由于生产过程的变异而不同，但是软件一经设计和实现后，其不同的拷贝的质量是相同的；其次，硬件产品由于质量问题可能报废，而软件产品的质量问题可以通过将开发过程中的错误予以纠正来解决。

硬件会磨损，软件不会“磨损”，但却会退化。如图 1-2 至图 1-4 所示分别描述了硬件和软件的故障曲线。

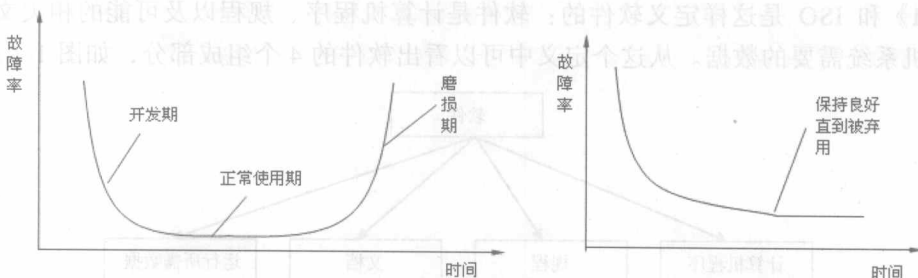


图 1-2 硬件的故障率曲线

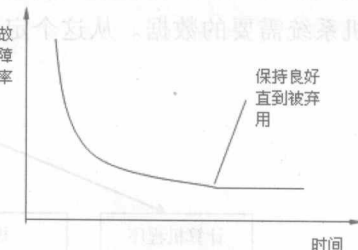


图 1-3 理想的软件故障曲线

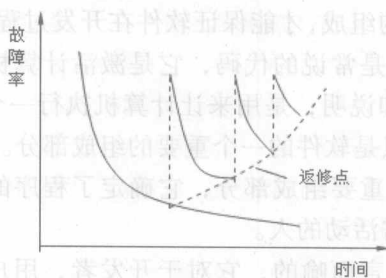


图 1-4 实际的软件故障曲线

软件分为系统软件和应用软件。

系统软件是给其他程序提供服务的程序集合（例如各种操作系统、编译程序、DBMS等）。系统软件的主要特征是：与计算机硬件系统有很强的交互性，并且要对共享资源进行调度管理，因而具有复杂的过程管理程序以解决并发性操作处理中存在的协调问题。而且系统软件中的数据结构复杂，外部接口多样化，以便于许多用户反复使用。

而应用软件又分为：

#### (1) 管理软件。

管理软件是指对管理信息进行处理的应用软件。管理软件往往集成各种分离的应用软件，例如将生产管理、计划管理、工资管理、财务管理、物料管理等分离的应用软件集成起来以支持企业的各项管理工作，管理软件应用一个或多个大型数据库管理系统来存放有关管理信息，并对数据库中存放的数据进行适当的处理，以辅助经营管理和决策。

#### (2) 实时软件。

对现实世界中随机发生的事件进行监测、分析和控制的软件，称为实时软件。实时软件一般有用于收集数据的基本模块，分析模块和监控模块。监控模块根据分析模块提供的信息对外部环境作出正确、实时的响应（一般要求的响应时间范围是1ms到1min之间）

#### (3) 工程分析和科学计算软件。

工程分析和科学计算软件是以对数据进行按部就班的处理为特征的软件，其应用范围非常广泛。例如，天气预报、自动应力分析、空间运行轨道动力学分析、分子生物学研究等。

#### (4) 嵌入式软件。

现在，智能产品几乎出现在所有的家电市场和工业市场上。智能产品中所谓的智能主要是因为产品中嵌入了具有一定程度的智能功能的软件。一般来说，嵌入式软件仅完成一些非常有限但又比较重要的功能和控制能力。例如微波炉上的操作键的控制、汽车中的油门控制等。

#### (5) 人工智能软件。

人工智能软件利用非数学算法解决复杂问题，这些问题往往不是单纯用计算和直觉分析所能解决的。例如，专家系统等。

### 1.1.2 软件错误

一个软件包为一个机构长期成功地服务，可能突然间就变得有毛病了，这是由软件的特性决定的。这种软件错误是不良软件质量的原因，它可以分为代码错、过程错、文档错和软件数据错，如图1-5所示。

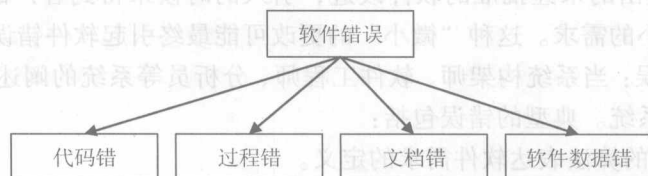


图1-5 软件错误的内容

应该注意到，这些软件错误的原因都是人为造成的。软件错误的原因按照它们发生在

软件开发过程的阶段可以进一步分类，如图 1-6 所示。

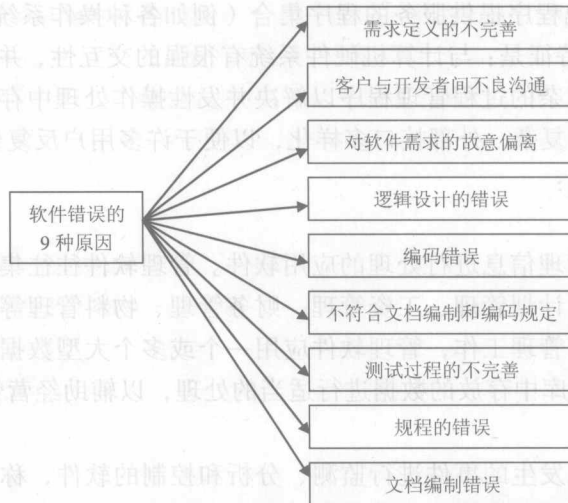


图 1-6 软件错误的 9 种原因

需求定义的不完善，它是软件错误的主要原因之一，包括：

- (1) 需求的不完备定义。
- (2) 需求的错误性定义。
- (3) 缺少至关重要的需求。
- (4) 包括了不必要的需求或者是在近期不需要的功能。

客户与开发者间不良沟通：由于客户与开发者间不良沟通所造成的误解是软件开发早期阶段盛行错误的另一重要原因：

- (1) 错误理解客户在需求文档中所陈述的指令。
- (2) 错误理解客户在开发阶段以书面形式向开发者提出的需求更改。
- (3) 错误理解客户在开发阶段以口头形式向开发者提出的需求更改。
- (4) 错误理解客户对开发者提出设计问题的应答。
- (5) 没有重视需求更改的客户消息和客户对开发者从开发者一方提出的问题的应答。

对软件需求的故意偏离：开发者故意偏离文档化的需求经常引起软件错误，这些故意偏离包括：

- (1) 开发者直接重用以前项目的模块，却没有对正确满足所有新需求所需的更改和适应进行足够的分析。
- (2) 由于时间或经费紧张，开发者为应对这些压力删去部分需要的功能。
- (3) 开发者提出的未经批准的软件改进，引入的时候未得到客户的批准，往往忽视那些开发者看来微小的需求。这种“微小”的更改可能最终引起软件错误。

逻辑设计的错误：当系统构架师、软件工程师、分析员等系统的阐述软件需求的时候，软件错误能够进入系统。典型的错误包括：

- (1) 通过错误的算法表达软件需求的定义。
- (2) 包括顺序性错误的过程定义。
- (3) 边界条件的错误定义。

(4) 漏掉需要的软件系统状态。

(5) 漏掉定义有关软件系统非法操作的反应。

编码错误：程序员产生编码错误的原因是多种多样的。这包括错误理解设计文档、编程语言中的语言性错误、CASE 和其他开发工具的应用错误、选择错误的数据库等。

不符合文档编制和编码规定：基本上所有的开发单位都有自己的文档编制与编码标准，这些标准定义文档的内容、次序和格式以及由成员组编制的代码。由于不遵守这些规定所引起的软件错误，即使在软件开发的前期不被人重视，也会给未来的维护工作带来相当大的难度：

(1) 负责协调不规范的代码模块的组员试图去理解其他规范软件的时候，将会遇到许多并不常见的困难。

(2) 由于某种原因代替那些“不规范”组员工作的人将会发现很难完全理解其工作。

(3) 设计评审人员将发现难以评审由“不规范”软件开发小组编制的设计。

(4) 测试人员将发现更加难以测试这种模块，因此，他们的效率会降低并且会遗漏掉更多未检测到的错误。

(5) 软件的友好性将会降低。用户在使用的过程中可能会发现更多的“bug”，因此，会增加软件维护的成本和难度。

测试过程的不完善：

(1) 不完善的测试过程带来了大量的未检测到和未改正的错误，增加了出错率。

(2) 不完善的测试计划留下了未处理的部分或应用功能或系统状态。

(3) 软件故障因为没有合适地指出故障原因而没有得到及时改正。

(4) 没有将检测到的错误和故障及时记入文档和报告。

(5) 由于疏忽或时间压力，对检测到错误的不完全改正。

规程的错误：规程是软件的一个重要组成部分，它确定了程序的执行流程、进度、方法应用并且负责为应用软件完成所需活动的人。规程向用户指引每个处理步骤所需要的活动。它们对复杂的软件系统特别重要，由于处理是分为若干步骤来进行的，每一步都要输入各种类型的数据并且允许检查中间结果。

文档编制错误：软件体中的错误是设计文档中的错误和文档编制的错误的集成，它们就是给开发和维护人员制造麻烦的文档编辑错误。它们往往会在开发的后面阶段或维护阶段带来更加严重的错误。

还有一种主要影响用户的文档编辑错误，它们通常在用户手册和软件的帮助中有详细的说明，包括：

(1) 遗漏的软件功能。

(2) 给用户的解释和指令中的错误以及会产生“死胡同”或者不正确的应用。

(3) 不存在的软件功能，即在早期的软件设计阶段包含了的，而未被实现的功能和因为软件升级等原因所取消的功能。

### 1.1.3 软件危机

软件危机是指计算机软件开发与维护过程中所遇到的一系列严重问题。这些问题并不限于那些被认为是“不能正常运行”的软件。



软件危机在 60 年代末 70 年代初全面爆发。

软件危机表现在以下几个方面：

- (1) 软件价格高。
- (2) 软件成本在计算机系统总成本中所占的比重呈现日益扩大的趋势。
- (3) 软件工作量难于估计。
- (4) 软件的特点决定了软件开发与维护的工作量难于量化，工作量与人力之间是非线性关系，在已拖延的软件项目上，增加人力只会使其更难按期完成。
- (5) 软件质量低。

软件的质量问题与其他商品有着很大的不同，软件的质量在很大程度上取决于用户对软件产品的认同（功能、性能、可靠性、用户界面等），而软件设计人员对计算机的了解与认识与用户的想法存在着明显的、不可克服的差距。常常表现为用户对软件产品各种各样的不满意或抱怨。

- (6) 软件修改与维护困难。

软件的修改与维护，看起来简单，实际上非常复杂。表现在：软件是一个庞大的逻辑体系，常常会发生“纠正一个错误会带来更多新的错误”的问题；软件的维护包括改正性维护、适应性维护及提高性维护。

- (7) 软件的供需差在不断扩大。

由于信息化的迅速发展，对软件的需求量大幅度增加，但软件生产率低，只能提供有限的产品，因此供需矛盾越来越大。

- (8) 软件的开发与维护过程往往失去控制。

大型复杂的软件系统开发，不仅需要大量的技术工作，同时需要大量的管理工作，从某种意义上讲，管理工作比技术工作更重要。

软件危机的原因可以归结为以下几点：

- (1) 软件规模越来越大。
- (2) 软件复杂度越来越高。
- (3) 软件的灵活性差。
- (4) 影响软件生产率及质量的因素十分复杂。

随着计算机技术的迅猛发展及其应用领域的日益扩大，软件规模越来越大。软件系统的开发往往需要花费几十、几百年。例如，非常大规模软件系统的定义已从 50 万行增加到 1000 万行。

软件不仅仅在规模上快速地发展扩大，而且其复杂性也急剧增加，表现在系统规模的扩大，子系统增多，子系统之间的接口及交互作用复杂性提高；计算机应用从数值计算扩展到非数值计算的领域，其复杂性迅速提高。

在市场及环境变化莫测的社会里，对软件的适应性要求越来越高，而最优秀的软件，其适应变化的能力都是有一定的限度的。

影响软件生产率及质量的因素众多而复杂，而且难以管理与控制，包括：

- (1) 个人能力：本身的素质、对相关领域的熟悉程度。
- (2) 团队联系：个人之间的合作方式。
- (3) 产品复杂度：产品越复杂，生产率越低。