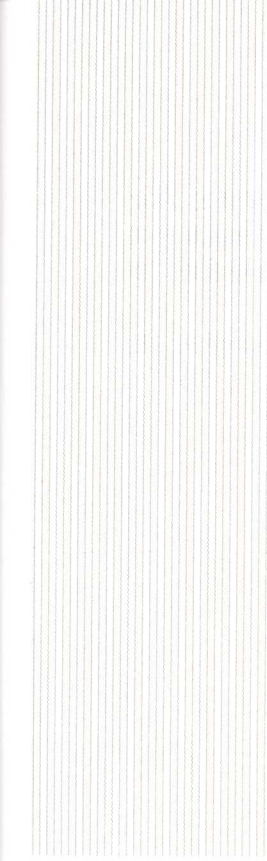


郁莲 编著

软件测试 方法与实践

清华大学出版社





郁莲 编著

软件测试方法与实践

清华大学出版社
北京

内 容 简 介

本书系统介绍现代软件测试的基本原理与一般方法。全书共分 10 章,内容包括软件测试概述、白盒测试、黑盒测试、软件测试覆盖分析、单元测试与集成测试、JUnit 测试工具、回归测试、基于状态的软件测试技术、面向对象的应用测试、Web 应用软件测试技术。各章均有总结、思考与练习题、课后作业和进一步阅读材料、以便巩固加深所学的知识。

本书可作为计算机科学软件工程专业的本科高年级学生及研究生的教科书,以及从事软件测试工作的技术人员的参考书。

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

软件测试方法与实践/郁莲编著. —北京:清华大学出版社,2008.11

ISBN 978-7-302-18458-4

I. 软… II. 郁… III. 软件—测试 IV. TP311.5

中国版本图书馆 CIP 数据核字(2008)第 132659 号

责任编辑:丁 岭 李 晔

责任校对:梁 毅

责任印制:杨 艳

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

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

印 装 者:北京市清华园胶印厂

经 销:全国新华书店

开 本:185×230 印 张:14.75 字 数:326 千字

版 次:2008 年 11 月第 1 版 印 次:2008 年 11 月第 1 次印刷

印 数:1~3000

定 价:28.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:023856-01

前言

在高度信息化的今天,信息技术已经成为社会发展的第一生产力,软件则是信息技术中最重要的组成部分。近年来,软件产业在很多国家都成为了国民经济的主导产业。但随着软件的规模和复杂性的大幅度提升,软件不可靠性的矛盾也变得日益突出,因此如何保证软件的质量成为了必须解决的问题。在 20 世纪,由于需求和认识等方面的原因,更多的人只是关注软件开发,而软件测试一直没有得到足够的重视,发展比较缓慢。随着软件质量保证理论与技术的快速发展,软件测试逐渐受到越来越广泛的重视,并正在形成一种产业,从业人员的数量也在大幅度增加。目前中国有一千多家软件评测中心,从事软件测试的人员有数万人,但仍然有约二十万的人才空缺。这些紧缺人才并不是只会点点鼠标的测试操作者,而是具有与开发人员相同甚至更高能力的测试设计师和分析员。

1. 预备知识要求和目标

本书既可作为初次接触软件测试的读者系统学习的入门教材,也可作为具有一定经验的测试人员随时翻阅的工具书。本书难度适中,希望读者通过阅读和学习,能够了解软件测试的重要性,掌握基本的软件测试技术。不论是哪类读者,要深入理解本书的内容,软件工程的基础知识都是必需的。

另外,最后两章涉及面向对象的应用测试和 Web 应用软件测试,如果读者具有一定的面向对象开发基础和 Web 应用开发基础,便能够更加透彻地理解这两章的内容。当然,这并不是必需的,即使没有这方面的经验,读者也可以利用章节最后列出的进一步阅读材料了解相关的知识。

2. 本书章节安排

软件测试的基础包括黑盒测试技术和白盒测试技术,除此之外,随着人们对软件及软件错误认识的不断深入,新的测试方法也应运而生。本书在重点讲解软件测试基本技术的同时,也对一些新的软件测试技术进行了介绍。

第 1 章 软件测试概述。软件测试是为了确认软件做了所期望的事情(Do the Right Thing),另一方面是确认软件以正确的方式来做了这个事件(Do it Right)。软件测试不仅

Foreword

是在测试软件产品本身,而且还包括软件开发的过程。这一章将介绍软件测试基本概念、软件测试目的、软件测试类型、软件测试原则、软件测试现状与挑战和测试人员职业发展与素质。

第2章 白盒测试。白盒测试(White-box Testing)是一种基于源程序或代码的测试方法,包括静态和动态两种类型。静态方法是指按一定步骤直接检查源代码以发现错误,也称为代码检查法;动态方法是指按一定步骤生成测试用例并驱动被测程序运行以发现错误。这一章介绍了基本路径测试、条件测试、数据流测试及循环测试等动态方法,以及有桌面检查、代码审查及走查等静态方法。

第3章 黑盒测试。黑盒测试(Black-box Testing)注重于测试软件的功能性需求,即黑盒测试需要软件工程师生成输入条件集来检测程序所有的功能需求。黑盒测试用于配合白盒测试发现其他类型的错误,包括功能错误或遗漏、界面错误、数据结构或外部数据库访问错误、性能错误和初始化和终止错误。这一章对黑盒测试技术进行了详细的讲解。

第4章 软件测试覆盖分析。在掌握了白盒测试与黑盒测试技术后,还需要一种方式来了解测试已经执行的程度,即本章要介绍的软件测试覆盖分析(Coverage Analysis)技术。在测试计划阶段,测试者确定用何种测试覆盖分析及相应的覆盖率;在测试执行阶段,将根据既定的覆盖率来检查是否进行了足够的测试。这一章将主要地介绍面向白盒测试技术的代码覆盖分析(控制流覆盖和数据流覆盖),并简要地介绍几种面向黑盒测试技术的覆盖分析。

第5章 单元测试与集成测试。分阶段测试是一种基本的测试策略,其中单元测试(Unit Testing)是对最小的软件设计单元(模块或源程序单元)的验证工作,集成测试(Integration Testing)把单独的软件模块结合在一起,作为一个群接受测试。这一章将介绍单元测试与集成测试概念、目标及过程。

第6章 JUnit 测试工具。JUnit 是一个开源的 Java 编程语言的单元测试框架,最初由 Erich Gamma 和 Kent Beck 编写。它在代码驱动单元测试框架家族里无疑是最为成功的一例。学习 JUnit 不但可以掌握一种有力的测试工具,更能帮助读者深入了解测试过程,对今后使用其他测试工具甚至开发测试工具都有极大的帮助。这一章介绍了 JUnit 的安装与使用,并讲解了 JUnit 自身的设计。

第7章 回归测试。在软件开发、维护、升级的不断演进过程中,由于各种各样的原因例如功能性/非功能性需求的改变、技术更新和升级了的软硬件平台,使得软件系统经常发生改变。这使得回归测试变得非常重要。回归测试(Regression Testing)是对之前已测试过、经过修改了的程序进行的重新测试,以保证该修改没有引入新的错误或者由于更改而发现之前未发现的错误,这是这一章的主要内容。

第8章 基于状态的软件测试技术。基于状态的软件测试是一种基于模型的测试技术。通过建立描述系统行为的状态机来自动生成测试用例。这一章介绍基于状态模型的软件测试技术,这在对实时系统、嵌入式系统、Web 交互性系统中具有广泛的应用。

第9章 面向对象的应用测试。面向对象是现代软件开发的主流,面向对象的应用测试

是相关活动的集合,是为了发现存在于 OOA、OOD、类、方法(操作)以及类间交互方面的错误。为了完成这些活动,在面向对象的应用中要使用包括静态评审和动态执行测试在内的测试策略。这一章详细论述了面向对象的软件测试技术,以及它与传统的软件测试方法的区别。

第 10 章 Web 应用软件测试技术。Web 应用测试是测试每个 Web 应用质量的纬度(Dimension),目的是发现错误或者发现导致质量失败问题。测试集中在内容、功能、结构、易用性、导航、性能、兼容性、互操作、容量、安全等方面。测试应该和 Web 应用设计的评审相结合。这一章对 Web 应用测试进行了详细的讲解。

3. 思考练习与进一步阅读

不管是课堂讲授还是自学,简单的练习对加强理解都是特别有用的,因此本书的每一章后面都有一组思考与练习。这些题目难度适中,既可以在课堂内完成,以帮助老师了解学生对这些内容的理解掌握程度,也可以做为课后的练习,帮助学生复习和巩固学到的知识。

对于那些不满足于课本内容的学生和读者,一系列相关的扩展阅读无疑像是科学探索中一盏盏引路的明灯,因此每一章最后都列出了与本章内容有关的一些学术论文或书籍,供学有余力的读者进一步深入学习。

4. 错误

无论作者有多少发现错误的技巧,总会有一些错误没被发现。如果读者发现任何可能是错误的地方,欢迎提出纠正建议,通过电子邮件 lianyu@ss.pku.edu.cn 反馈给作者,以便在本书下一次印刷时更正。同时欢迎读者为下一个版本的补充练习或要求提出宝贵建议。衷心感谢读者的帮助。

5. 感谢

作者近几年来一直在北京大学软件与微电子学院从事软件测试技术的研究与授课,并参与了很多软件质量保证方面的课题研究与工程项目,与国内外知名的研究单位、软件公司建立了广泛的学术联系。感谢学院对我的工作给予的极大支持与重视,感谢学院领导与老师们对我讲授的“软件测试技术”课程的大力支持。

硕士研究生周季、相慧如、伍晓东、张曲艳、吴芳、张乐、张辉辉、邸晓峰、许夏、宋胜攀等同学参与了本书的校对与整理工作,付出了很多努力,在此表示感谢。感谢北京大学软件与微电子学院 0509、0609 和 0709 届所有选修过“软件测试技术”课的同学,你们的反馈意见对我的教学工作与本书的编写都有很大的帮助。

郁 莲

2008 年 6 月

读者意见反馈

亲爱的读者：

感谢您一直以来对清华版计算机教材的支持和爱护。为了今后为您提供更优秀的教材，请您抽出宝贵的时间来填写下面的意见反馈表，以便我们更好地对本教材做进一步改进。同时如果您在使用本教材的过程中遇到了什么问题，或者有什么好的建议，也请您来信告诉我们。

地址：北京市海淀区双清路学研大厦 A 座 602 室 计算机与信息分社营销室 收

邮编：100084

电子邮件：jsjic@tup.tsinghua.edu.cn

电话：010-62770175-4608/4409

邮购电话：010-62786544

教材名称：软件测试方法与实践

ISBN 978-7-302-18458-4

个人资料

姓名：_____ 年龄：_____ 所在院校/专业：_____

文化程度：_____ 通信地址：_____

联系电话：_____ 电子信箱：_____

您使用本书是作为：☐指定教材 ☐选用教材 ☐辅导教材 ☐自学教材

您对本书封面设计的满意度：

☐很满意 ☐满意 ☐一般 ☐不满意 改进建议_____

您对本书印刷质量的满意度：

☐很满意 ☐满意 ☐一般 ☐不满意 改进建议_____

您对本书的总体满意度：

从语言质量角度看 ☐很满意 ☐满意 ☐一般 ☐不满意

从科技含量角度看 ☐很满意 ☐满意 ☐一般 ☐不满意

本书最令您满意的是：

☐指导明确 ☐内容充实 ☐讲解详尽 ☐实例丰富

您认为本书在哪些地方应进行修改？（可附页）

您希望本书在哪些方面进行改进？（可附页）

电子教案支持

敬爱的教师：

为了配合本课程的教学需要，本教材配有配套的电子教案(素材)，有需求的教师可以与 我们联系，我们将向使用本教材进行教学的教师免费赠送电子教案(素材)，希望有助于教学活动的开展。相关信息请拨打电话 010-62776969 或发送电子邮件至 jsjic@tup.tsinghua.edu.cn 咨询，也可以到清华大学出版社主页(<http://www.tup.com.cn> 或 <http://www.tup.tsinghua.edu.cn>)上查询。

目 录

第 1 章 软件测试概述

/1

1.1	什么是软件测试	2
1.2	软件测试目的	2
1.3	软件测试原理	3
1.4	软件测试过程	4
1.5	软件测试类型	6
1.5.1	按照开发阶段划分	6
1.5.2	按照测试技术划分	8
1.5.3	按照执行状态划分	8
1.5.4	按照执行主体划分	9
1.6	软件测试的注意事项(Tip)	9
1.7	软件测试的现状和趋势与面临的挑战	10
1.7.1	现状和趋势	10
1.7.2	面临的挑战	11
1.8	测试人员职业发展与具备的素质	12
1.8.1	从测试工程师的市场角度来分析	13
1.8.2	从测试工程师的自身素质提高的角度来看	13
1.9	总结	13
1.10	参考文献	14
1.11	思考与练习	14
1.12	进一步阅读	14
1.13	课后作业	15

第 2 章 白盒测试

/17

2.1	基本路径测试	18
-----	--------------	----

Contents

2.1.1	流图符号	18
2.1.2	独立程序路径	19
2.1.3	环形复杂性	20
2.1.4	导出测试用例	21
2.1.5	图矩阵法	22
2.2	控制结构测试	24
2.2.1	条件测试	24
2.2.2	数据流测试	26
2.2.3	循环测试	29
2.3	代码检查法	31
2.3.1	代码审查	31
2.3.2	桌面检查	37
2.3.3	走查	41
2.4	总结	42
2.5	参考文献	42
2.6	思考与练习	43
2.7	进一步阅读	44

第3章 黑盒测试

/45

3.1	基于图的测试方法	46
3.2	等价划分	48
3.3	边界值分析	49
3.4	因果分析法	49
3.4.1	因果图——图形符号	50
3.4.2	因果图——举例	51
3.5	正交数组测试	53
3.6	测试插桩	55
3.6.1	测试预言	56
3.6.2	随机数据生成器	58
3.7	总结	59
3.8	参考文献	60
3.9	思考与练习	60
3.10	进一步阅读	61

第4章 软件测试覆盖分析

/62

4.1	代码覆盖分析	63
4.2	控制流覆盖	64
4.2.1	语句覆盖	64
4.2.2	判定覆盖	65
4.2.3	条件覆盖	66
4.2.4	条件判定组合覆盖	66
4.2.5	多条件覆盖	67
4.2.6	修正条件/判定覆盖	69
4.2.7	路径覆盖	70
4.3	数据流覆盖	71
4.3.1	Rapps 和 Weyuker 的标准	71
4.3.2	Ntafos 的标准	75
4.3.3	Ural 的标准	76
4.3.4	Laski 和 Korel 的标准	76
4.4	其他覆盖标准	78
4.4.1	数据域覆盖	78
4.4.2	统计或可靠性覆盖	78
4.4.3	风险覆盖	78
4.4.4	安全覆盖	79
4.4.5	状态模型的覆盖标准	79
4.4.6	覆盖标准有关问题、局限性	79
4.4.7	实际应用的建议	80
4.5	总结	81
4.6	参考文献	81
4.7	思考与练习	82
4.8	进一步阅读	82

第5章 单元测试与集成测试

/84

5.1	单元测试	85
5.1.1	单元测试考虑事项	85
5.1.2	单元测试规程	87
5.1.3	单元测试局限性	88

5.2	集成测试	88
5.2.1	自顶向下集成	90
5.2.2	自底向上集成	91
5.2.3	混合式集成	92
5.2.4	端到端集成测试	94
5.3	总结	100
5.4	参考文献	100
5.5	思考与练习	101
5.6	进一步阅读	101
第6章 JUnit 测试工具 /102		
6.1	使用 JUnit	103
6.1.1	一个简单的例子	103
6.1.2	JUnit 安装与运行	104
6.1.3	JUnit 常见问题	108
6.1.4	一个自动售货机的例子	113
6.2	JUnit 的设计	117
6.2.1	设计目标	118
6.2.2	JUnit 设计	118
6.2.3	小结	124
6.3	模仿对象测试	127
6.3.1	模仿对象简介	127
6.3.2	模仿对象与重构	129
6.3.3	利用工具建立模仿对象	136
6.3.4	小结	138
6.4	DbUnit 单元测试	138
6.4.1	DbUnit 简介	138
6.4.2	使用 DbUnit	139
6.4.3	小结	143
6.5	JUnit4 简介	144
6.5.1	一个小例子	144
6.5.2	JUnit 4 的注解	146
6.5.3	小结	146
6.6	JUnit、Mock Object 和 DbUnit 的作业	147
6.7	参考文献	148

第7章 回归测试

/149

7.1 回归测试的特点	150
7.2 回归测试的过程	151
7.2.1 重新确认测试用例	152
7.2.2 识别错误	152
7.3 回归测试的策略	153
7.4 波及效应分析	154
7.4.1 波及效应分析步骤	154
7.4.2 程序切片	156
7.5 回归测试的花费	157
7.6 总结	158
7.7 参考文献	159
7.8 思考与练习	159
7.9 进一步阅读	159

第8章 基于状态的软件测试技术

/160

8.1 状态转换图	161
8.2 状态图	165
8.2.1 Harel 状态图的属性	166
8.2.2 从状态图变换到 STD	170
8.2.3 UML 状态图	171
8.3 基于状态的测试	171
8.3.1 测试步骤	172
8.3.2 产生测试用例	173
8.3.3 覆盖分析	177
8.4 总结	178
8.5 参考文献	179
8.6 思考与练习	179
8.7 进一步阅读	180

第9章 面向对象的应用测试

/181

9.1 OO 测试方法	182
9.1.1 OO 概念对测试用例设计影响	183

9.1.2	传统测试用例设计方法的可用性	183
9.1.3	基于故障的测试	183
9.1.4	OO 编程对测试的影响	184
9.1.5	测试用例和类层次	185
9.1.6	基于场景的测试	186
9.1.7	测试表层结构和深层结构	187
9.2	在类级别上可用的测试方法	188
9.2.1	面向对象的随机测试	188
9.2.2	在类级别上的划分测试	188
9.3	类间测试用例设计	189
9.3.1	多个类测试	189
9.3.2	从行为模型导出的测试	190
9.4	总结	192
9.5	参考文献	192
9.6	思考与练习	193
9.7	进一步阅读	193

第 10 章 Web 应用软件测试技术

/194

10.1	Web 应用测试概念	195
10.1.1	质量的纬度	195
10.1.2	Web 应用环境中的错误	196
10.1.3	测试策略	197
10.1.4	测试计划	197
10.2	测试过程概述	198
10.3	内容测试	200
10.3.1	内容测试目标	201
10.3.2	数据库测试	202
10.4	用户界面测试	203
10.4.1	界面测试策略	203
10.4.2	测试界面机制	204
10.4.3	测试界面语义	206
10.4.4	易用性测试	206
10.4.5	兼容性测试	207
10.5	组件级测试	209
10.6	导航测试	211

10.6.1	测试导航语法	211
10.6.2	测试导航语义	212
10.7	配置测试	213
10.7.1	服务器端问题	213
10.7.2	客户端问题	213
10.8	安全测试	214
10.9	性能测试	215
10.9.1	性能测试目标	216
10.9.2	负载测试	216
10.9.3	压力测试	217
10.10	总结	217
10.11	参考文献	218
10.12	思考与练习	218
10.13	进一步阅读	219

第1章

软件测试概述

软件测试是软件质量保证的重要手段。有研究数据显示,国外软件开发机构 40% 的工作量花在软件测试上,软件测试费用占软件开发总费用的 30%~50%。对于一些要求高可靠、高安全的软件,测试费用可能相当于整个软件项目开发所有费用的 3~5 倍。由此可见,要成功开发出高质量的软件产品,除了从思想上重视软件测试工作,还必须掌握测试技术,有效地实施测试工作。

本章的内容包括软件测试基本概念、软件测试目的、软件测试类型、软件测试原则、软件测试现状与挑战以及测试人员职业发展与素质。

快速阅览:

什么是软件测试? Myers (1979) 定义测试 (Testing) 是执行程序的过程,其目的是发现错误。IEEE 610.12 标准(1990)给出了两个测试定义:

- (1) 在特定的条件下运行系统或构件,观察或记录结果,对系统的某个方面做出评价。
- (2) 分析某个软件项以发现现存的和要求的条件之差别(即错误)并评价此软件项的特性。

由谁来负责软件测试? 在测试初期,由软件工程师实施所有测试。然而,随着测试过程进行,测试专业人员应该加入进来。

为什么软件测试如此重要? 没有经过测试的软件产品,无法知晓该软件产品运行时是否满足用户功能、性能需求,甚至导致最终用户生命、财产的损失。为了在把软件产品交付给用户之前尽可能多地发现错误(Error),必须使用专业技术设计测试用例,进行系统化测试。

软件测试步骤各是什么? 软件测试过程主要包括 4 个步骤:制定测试计划、生成测试用例、执行测试和分析测试结果。

有哪些工件形成? 在一些情况下,会生成测试计划、测试用例和测试结果报告。测试结果存档以便将来软件维护时使用。

如何确保我们准确地完成了任务? 尽管永远不能保证已经执行了所有可能的测试,但能肯定测试已经发现了错误(并且已修正了这些错误)。另外,如果已经制定了一个测试计

划,则可以进行检查以保证所有计划测试已被完成。

1.1 什么是软件测试

Glenford J. Myers (1979)^[1] 定义测试 (Testing) 是执行程序的过程,其目的是发现错误。IEEE 标准 610.12(1990)^[2] 给出了两个更为规范、约束的测试定义:

(1) 在特定的条件下运行系统或构件,观察或记录结果,对系统的某个方面做出评价。

(2) 分析某个软件项以发现现存的和要求的条件之差别(即错误)并评价此软件项的特性。

应该说,IEEE 610.12 标准的定义扩展了 Myers 的定义。IEEE 610.12 标准的定义(2)并不要求运行程序作为测试过程的一部分,静态验证的一些方法可作为测试手段。定义(1)被称为动态测试,而定义(2)被称为静态测试。软件是由文档、数据以及程序等工件组成,软件测试应该对于软件形成过程的文档、数据,以及程序进行测试。定义(2)的技术可以应用于非程序的工件。本书将介绍几种常见的静态测试,主要介绍软件的动态测试。

注意,英文 testing 和 test 在翻译成中文时,虽然都译为“测试”,但它们是有区别的。testing 是指一个过程,而 test 一般是执行测试的一次活动。在看到“测试”一词时,要根据上下文来判断是哪种意思。本书中的“测试”是指 testing。

软件测试是软件开发过程的重要组成部分,用来确认一个程序的品质或性能是否符合开发之前所提出的一些要求。软件测试是在软件投入运行前,对软件需求分析、设计规格说明和编码的最终评审(Review),是软件质量保证的关键步骤。

软件测试是为了发现错误而执行程序的过程。软件测试在软件生命期中要横跨其中的两个阶段:

(1) 通常在编写出每一个模块之后就对它做必要的测试(称为单元测试)。编码和单元测试属于软件生存期中的同一个阶段。

(2) 在结束这个阶段后对软件系统还要进行各种综合测试,这是软件生存期的另一个独立阶段,即测试阶段。

注意,有资料表明,60%以上的软件错误并不是程序错误,而是软件需求和软件设计错误。错误的理解用户需求会导致开发技术完美、优良,但却不是正确的产品。因此,做好软件需求和软件设计阶段的质量保证工作也是非常重要的。

1.2 软件测试目的

Myers 这样来描述软件测试的目的:“测试是程序的执行过程,目的在于发现错误;一个好的测试用例是指很可能找到迄今为止尚未发现的错误的用例。一个成功的测试是指发

现了至今尚未发现的错误的测试。”

Bill Hetzel^[3]提出了测试目的不仅仅是为了发现软件缺陷与错误,而且也是对软件质量进行度量和评估,以提高软件的质量。

测试的目的是要以最少的人力、物力和时间找出软件中的各种错误与缺陷,通过修正各种错误与缺陷来提高软件质量,避免软件发布后由于潜在的软件缺陷和错误造成的隐患所带来的经济风险。同时,测试是以评价一个程序或者系统属性为目标的活动,测试是对软件质量的度量与评价,以验证软件的质量满足用户的需求的程度,为用户选择与接受软件提供有力的依据。

此外,通过分析错误产生的原因还可以帮助发现当前开发工作所采用的软件过程的缺陷,以便进行软件过程改进。同时通过对软件结果的分析整理,为风险评估提供信息,还可以修正软件开发规则,并以软件可靠性分析提供依据。当然,通过最终的验收测试,也可以证明软件满足了用户的需求,树立人们使用软件的信心。

软件质量可用几个方面来衡量:

(1) 在正确的时间用正确的方法做正确的事情(Doing the Right Things Right at the right time)。

(2) 符合一些应用标准的要求,比如不同国家中用户不同的操作习惯和要求,项目工程中的可维护性、可测试性等要求。

(3) 质量本身就是软件达到了最开始所设定的要求,而优美或精巧的表现技巧并不代表软件的高质量(Quality is defined as conformance to requirements, not as “goodness” or “elegance”)。

(4) 质量也代表着它符合客户需要(Quality also means “meet customer needs”)。在软件测试这个行业中,最重要的一件事就是从用户需求出发,从用户的角度去看产品,用户会怎么去使用这个产品,使用过程中会遇到什么样的问题。只有这些问题都解决了,软件产品的质量才可以说是得到了保证。

测试人员的总体目标是确保软件的质量,他们在软件开发过程中的任务是:寻找错误(Bug),避免软件开发过程中的缺陷,衡量软件的品质,关注用户需求。

1.3 软件测试原理

上面讲到测试的目的是为了寻找软件的错误与缺陷,评估与提高软件质量。那么为了要达到测试目的,应遵循以下软件测试的基本原则。

- 所有的测试都应追溯到用户需求,软件测试的目标在于揭示错误。而最严重的错误(从用户角度来看)是那些导致程序无法满足需求的错误。
- 测试计划的制定应先于测试的执行。测试计划可以在需求模型一完成就开始,详细