

21世纪高等学校规划教材 | 软件工程



软件测试实践教程

兰景英 编著



清华大学出版社

21世纪高等学校规划教材 | 软件工程



软件测试实践教程

兰景英 编著

清华大学出版社
北京

内 容 简 介

本书作为《软件测试技术》的实验教材,以培养工程实践能力为目标,以软件测试流程为主线,以主流的开源软件测试工具应用为基础,为高等院校软件工程专业和计算机相关专业开设软件测试实验课程提供全方位实践教学方案、实践教学平台和实践教学案例。

全书共7章,覆盖软件测试流程中各阶段的测试工具,其中包括测试管理工具 TestLink,缺陷管理工具 Mantis,静态分析工具 Checkstyle、FindBugs、Cppcheck 和 PC-lint,单元测试工具 JUnit 和 CppUnit,功能测试工具 QuickTest 和 Selenium,性能测试工具 LoadRunner 和 JMeter,以及安全测试工具 AppScan 等。

本书内容新颖,体系完整,结构清晰,实践性强,从原理、技术和应用三方面深入细致地介绍了软件测试过程中涉及的各类测试工具。

本书可作为高等院校、高职高专院校、示范性软件学院的软件工程及计算机相关专业的“软件测试实践课程”教材,也可作为软件测试实训的培训教材,同时可供从事软件开发、项目管理、软件测试或质量保证人员参阅。

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

软件测试实践教程/兰景英编著. —北京:清华大学出版社,2016

21 世纪高等学校规划教材·软件工程

ISBN 978-7-302-43274-6

I. ①软… II. ①兰… 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(2016)第 044127 号

责任编辑:付弘宇 薛 阳

封面设计:傅瑞学

责任校对:梁 毅

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:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:26 字 数:653 千字

版 次:2016 年 7 月第 1 版 印 次:2016 年 7 月第 1 次印刷

印 数:1~2000

定 价:49.80 元

产品编号:067879-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上;精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展,顺应并符合21世纪教学发展的规律,代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版

社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括:

(1) 21 世纪高等学校规划教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 21 世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 21 世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

(4) 21 世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

(5) 21 世纪高等学校规划教材·信息管理与信息系统。

(6) 21 世纪高等学校规划教材·财经管理与应用。

(7) 21 世纪高等学校规划教材·电子商务。

(8) 21 世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人:魏江江

E-mail:weijj@tup.tsinghua.edu.cn

前言

软件测试是软件工程的一个重要分支,是软件质量保证的重要基础。软件测试是一门动态、交叉性学科,跨越了软件工程的整个领域。软件测试实验性强,软件测试人才培养需要开展全面综合的实践训练,包括测试计划制订、测试用例设计、测试环境搭建、测试用例执行、测试结果评估和测试过程管理等。目前很多高校的计算机类专业均开设了这门课程,并配有一定学时的实验或独立安排软件测试实践课程。本书充分考虑到软件测试贯穿软件项目整个生命周期,需要用到大量测试技术和测试工具,对国内外主流的开源软件测试工具进行全面的分析、研究和精选,并结合作者近十年的软件测试教学经验,精心设计本书的实验内容,方便广大读者动手实践,提升测试技能,增强就业竞争力。

全书共7章,以软件测试流程为主线,以主流的开源软件测试工具应用为基础,深入细致地介绍各测试阶段需要用到的测试工具。

第1章软件测试管理,介绍软件测试管理各阶段和测试管理中相关测试文档的撰写。在测试管理过程中,为便于软件项目相关人员之间的交流和沟通,以及测试流程的管理,会引入软件测试管理工具。本章以 TestLink 为例,详细介绍了 TestLink 的安装、配置和使用。

第2章介绍缺陷管理的相关知识,包括缺陷分类、缺陷管理流程、缺陷报告原则。对于大型软件项目,通常离不开缺陷管理系统。本章以 Mantis 为例,详细介绍 Mantis 的安装、配置和使用。

第3章围绕代码静态测试展开,介绍了静态测试的概念、静态测试的工具。对于 Java 语言,分析静态测试工具 Checkstyle 和 FindBugs,并详细地介绍它们的安装和使用,并以代码为例分析静态测试过程和方法。对于 C/C++ 语言,介绍静态测试工具 Cppcheck 和 PC-lint 的安装和使用。

第4章单元测试是提高软件质量最直接和最重要的测试阶段。本章深入分析了白盒测试用例设计的方法和技术,通过典型的单元测试工具详细说明测试的过程。针对 Java 语言,介绍了 JUnit 的技术和应用流程,以及覆盖率测试工具 EcEmma,并以案例方式展示 JUnit 实施过程。针对 C++ 语言,介绍了 CppUnit 的技术、测试环境和测试过程。

第5章功能测试,介绍黑盒测试用例设计技术,分析常用的功能测试工具。针对商用测试工具 QuickTest,介绍其测试原理、测试流程,并以博客系统为例介绍 QuickTest 实施过程。接下来介绍开源测试工具 Selenium 的环境配置,测试过程,以及通过 JUnit 和 TestNG 执行 Selenium 测试脚本的过程。

第6章以性能测试为主题,阐明性能测试相关概念,详述性能测试指标和计数器。深入细致地介绍了最流行的性能测试工具 LoadRunner 的功能部件: Virtual User Generator、Controller 和 Analysis,以博客系统为例分析 LoadRunner 进行性能测试的实施流程。另外,还介绍了开源的性能测试 JMeter 的整个使用过程。

第7章针对 Web 安全测试展开讨论,对 Web 常见攻击进行分析,阐明 Web 安全测试的内容和常见的 Web 安全测试工具。详细介绍安全测试工具 AppScan 的使用过程,并以博客系统为例,展现应用 AppScan 进行安全测试的全过程。

本书最后附有软件测试文档模板、测试工具网站等资料。

本书涉及的软件测试知识广泛,实验内容全面、案例丰富、方案完整、步骤翔实、过程清晰,可逐步引导读者深入实践各类测试工具。实验内容覆盖了软件测试全过程所涉及的测试工具,教师可根据教学实际情况进行剪裁或扩充。本书适合学生学习、教师指导实验,以及培训机构开展软件测试实训。

感谢清华大学出版社提供的这次合作机会,使本教程能够早日与读者见面。感谢范勇教授、潘娅副教授和言若金叶的王顺老师为书籍出版所提供的支持和帮助,感谢家人的理解和支持。本书的大量内容取材于互联网,由于各种原因无法找到原创者,在参考文献中无法准确标注,在此表示歉意,并对原创者表示感谢。

由于作者水平和时间的限制,书中难免存在疏漏,欢迎读者及各界同人批评指正。

编 者

目 录

第 1 章 软件测试过程管理	1
1.1 软件测试管理基础	1
1.1.1 软件测试管理	1
1.1.2 软件测试过程管理	1
1.1.3 软件测试相关文档	2
1.1.4 软件测试管理工具	5
1.2 TestLink	7
1.2.1 XAMPP 的安装	7
1.2.2 安装 TestLink	10
1.2.3 TestLink 简介	14
1.2.4 TestLink 的使用	15
1.3 软件测试管理实验	30
第 2 章 软件缺陷管理	32
2.1 软件缺陷基础	32
2.1.1 软件缺陷	32
2.1.2 软件缺陷管理	35
2.1.3 软件缺陷管理工具	37
2.2 Mantis	38
2.2.1 Mantis 简介	38
2.2.2 Mantis 的安装	41
2.2.3 管理员的操作	43
2.2.4 权限用户的操作	56
2.2.5 指派给我的工作	57
2.3 软件缺陷管理实验	59
第 3 章 代码静态测试	61
3.1 代码静态测试	61
3.1.1 静态测试	61
3.1.2 静态测试工具	62
3.2 Checkstyle	63
3.2.1 Checkstyle 简介	63

3.2.2	Checkstyle 规则文件	64
3.2.3	Checkstyle 的安装	76
3.2.4	Checkstyle 的应用	77
3.3	FindBugs	83
3.3.1	FindBugs 简介	83
3.3.2	FindBugs 的安装	83
3.3.3	FindBugs 的使用	84
3.3.4	配置 FindBugs	86
3.4	Cppcheck	89
3.4.1	Cppcheck 简介	89
3.4.2	Cppcheck 的安装	90
3.4.3	Cppcheck 的使用	91
3.5	PC-lint	94
3.5.1	PC-lint 简介	94
3.5.2	PC-lint 的安装与配置	95
3.5.3	PC-lint 的代码检查功能	102
3.5.4	PC-lint 错误信息	103
3.5.5	PC-lint 的应用举例	114
3.6	代码静态测试实验	116
第 4 章	单元测试	118
4.1	单元测试基础	118
4.1.1	单元测试概念	118
4.1.2	白盒测试用例设计	118
4.1.3	白盒测试工具	124
4.2	xUnit 测试框架	124
4.3	JUnit	125
4.3.1	JUnit 简介	125
4.3.2	JUnit 测试技术	128
4.3.3	JUnit 的应用流程	131
4.3.4	JUnit 下的代码覆盖率工具 EcEmma	141
4.3.5	JUnit 测试应用举例	145
4.4	CppUnit	156
4.4.1	CppUnit 简介	156
4.4.2	CppUnit 测试技术	156
4.4.3	CppUnit 测试环境	162
4.4.4	CppUnit 示例	172
4.4.5	CppUnit 测试案例	176
4.5	单元测试实验	185

第 5 章 功能测试	186
5.1 功能测试基础	186
5.1.1 功能测试概念	186
5.1.2 黑盒测试用例设计	186
5.1.3 功能测试工具	196
5.2 QuickTest	200
5.2.1 QuickTest 简介	200
5.2.2 QuickTest 的安装	201
5.2.3 QuickTest 的使用	201
5.2.4 QuickTest 测试案例	212
5.3 Selenium	220
5.3.1 Selenium 简介	220
5.3.2 Selenium IDE 环境配置	221
5.3.3 Selenium IDE 应用	223
5.3.4 Firebug 应用	229
5.3.5 XPath Checker 应用	231
5.3.6 Selenium WebDriver 功能	232
5.3.7 Selenium WebDriver 环境配置	238
5.3.8 通过 JUnit 执行 Selenium 实例	239
5.3.9 通过 TestNG 执行 Selenium 实例	243
5.4 功能测试实验	247
第 6 章 性能测试	249
6.1 性能测试基础	249
6.1.1 性能测试概念	249
6.1.2 性能测试指标	252
6.1.3 性能计数器	255
6.1.4 性能测试工具	260
6.2 LoadRunner	265
6.2.1 LoadRunner 概述	265
6.2.2 脚本生成器	268
6.2.3 控制器	286
6.2.4 分析器	296
6.3 LoadRunner 性能测试案例	307
6.3.1 计划测试	307
6.3.2 建立测试环境	308
6.3.3 创建测试脚本	310
6.3.4 执行测试	318

6.3.5	分析测试结果	323
6.4	JMeter	324
6.4.1	JMeter 基础	324
6.4.2	JMeter 主要部件	326
6.4.3	JMeter 基本操作	333
6.4.4	Badboy 录制脚本	350
6.4.5	JMeter 性能测试案例	353
6.5	性能测试实验	357
第 7 章	Web 安全性测试	359
7.1	Web 安全测试基础	359
7.1.1	Web 常见攻击	359
7.1.2	Web 安全测试简介	364
7.1.3	Web 安全测试工具	365
7.2	AppScan	367
7.2.1	AppScan 概述	367
7.2.2	AppScan 窗口	370
7.2.3	AppScan 操作	373
7.3	Web 安全测试案例	385
7.3.1	创建扫描	385
7.3.2	执行扫描	386
7.3.3	扫描结果	387
7.3.4	结果报告	387
7.4	Web 安全测试实验	390
参考文献		392
附录		396
附录 A	软件测试文档模板	396
附录 B	测试工具网址	399
附录 C	博客系统	402

第1章

软件测试过程管理

1.1 软件测试管理基础

1.1.1 软件测试管理

软件测试过程中最重要的是进行有效的测试管理。测试管理包括对人的管理、对流程的管理、对软件产品版本的管理等内容。软件测试管理实际上是一系列活动,可以对各阶段的测试计划、测试用例、测试流程、测试文档等进行跟踪、管理并记录其结果,以实现测试的有效控制和管理,进一步提高测试的效率和质量。

从广义上讲,软件测试管理包括软件测试过程的定义、测试需求管理、测试计划管理、测试用例管理、缺陷管理、测试用例执行、测试报告、测试配置管理、自动化软件测试过程管理等内容。其中,测试过程管理、测试用例管理、测试用例执行和缺陷管理是软件测试管理的核心内容。

在软件测试过程中,使用测试管理工具对整个测试过程进行管理,可以提高测试的效率、缩短测试时间、提高测试质量、提升用例复用率、提高需求覆盖率等。一个完整的软件测试管理工具,能用于测试计划、测试用例、测试执行和缺陷跟踪等测试行为的管理,并能提供对人工测试和自动测试的分析、设计和管理功能,把应用程序测试中所涉及的全部任务集成起来,跟踪测试中的依赖关系和相互关联,并能对质量目标进行定义、测量和跟踪。

1.1.2 软件测试过程管理

软件测试不等于程序测试,软件测试贯穿于软件开发整个生命周期。软件测试过程包括测试准备、测试计划、测试设计、测试执行、测试结果分析。

1. 测试准备

测试准备阶段需要组建测试小组,参加有关项目计划、分析和设计会议,获取必要的需求分析、系统设计文档,以及相关产品/技术知识的培训。

2. 测试计划

测试计划阶段的主要工作是确定测试内容或质量特性,确定测试的充分性要求,制定测

试策略和方法,对可能出现的问题和风险进行分析和估计,制订测试资源计划和测试进度计划以指导测试的执行。

3. 测试设计

软件测试设计建立在测试计划之上,通过设计测试用例来完成测试内容,以实现所确定的测试目标。软件测试设计的主要内容如下。

1) 制定测试方案

分析测试技术方案是否可行、是否有效、是否能达到预定的测试目标。

2) 设计测试用例

选取和设计测试用例,获取并验证测试数据。根据测试资源、风险等约束条件,确定测试用例执行顺序。分析测试用例是否完整、是否考虑边界条件、能否达到其覆盖率要求。

3) 测试脚本开发

获取测试资源,如数据、文件等。开发测试软件,包括驱动模块、桩模块,录制和开发自动化测试脚本等。

4) 设计测试环境

建立并校准测试环境,分析测试环境是否和用户的实际使用环境接近。

5) 测试就绪审查

审查测试计划的合理性,审查测试用例的正确性、有效性和覆盖充分性,审查测试组织、环境和设备工具是否齐备并符合要求。在进入下一阶段工作之前,应通过测试就绪评审。

4. 测试执行

建立和设置好相关的测试环境,准备好测试数据,执行测试用例,获取测试结果。分析并判定测试结果,根据不同的判定结果采取相应的措施。对测试过程的正常或异常终止情况进行核对。根据核对的结果,对未达到测试终止条件的测试用例,决定是停止测试还是需要修改或补充测试用例集,并进一步测试。

5. 测试结果分析

测试结束后,评估测试效果和被测软件项,描述测试状态。对测试结果进行分析,以确定软件产品的质量,为产品的改进或发布提供数据和支持。在管理上,应做好测试结果的审查和分析,做好测试报告的撰写和审查工作。

1.1.3 软件测试相关文档

1. 测试计划

测试计划是描述要进行的测试活动的目的、范围、方法、资源和进度的文档。《ANSI/IEEE 软件测试文档标准 829—1983》将测试计划定义为:“一个描述了预定的测试活动的范围、途径、资源及进度安排的文档。它确认了测试项、被测特征、测试任务、人员安排,以及任何偶发事件的风险。”

测试计划是指导测试过程的纲领性文件,包含产品概述、测试策略、测试方法、测试区

域、测试配置、测试周期、测试资源、测试交流、风险分析等内容。借助软件测试计划,参与测试的项目成员可以明确测试任务和测试方法,保持测试实施过程的顺畅沟通,跟踪和控制测试进度,应对测试过程中的各种变更。

下面是编写测试计划的6要素。

why: 为什么要进行测试。

what: 测试哪些方面,不同阶段的工作内容是什么。

when: 测试不同阶段的起止时间。

where: 相应文档,缺陷的存放位置,测试环境等。

who: 项目有关人员组成,安排哪些测试人员进行测试。

how: 如何去做,使用哪些测试工具以及测试方法进行测试。

测试计划中一般包括以下关键内容。

(1) 测试需求: 明确测试的范围,估算出测试所花费的人力资源和各个测试需求的测试优先级。

(2) 测试方案: 整体测试的测试方法和每个测试需求的测试方法。

(3) 测试资源: 测试所需要用到的人力、硬件、软件、技术的资源。

(4) 测试组角色: 明确测试组内各个成员的角色和相关责任。

(5) 测试进度: 规划测试活动和测试时间。

(6) 可交付工件: 在测试组的工作中必须向项目组提交的产物,包括测试计划、测试报告等。

(7) 风险管理: 分析测试工作所可能出现的风险。

测试计划编写完毕后,必须提交给项目组全体成员,并由项目组中各个角色组联合评审。

2. 测试用例

测试用例(Test Case)是指对一项特定的软件产品进行测试任务的描述,体现测试方案、方法、技术和策略。测试用例一般包括下列信息。

1) 名称和标识

每个测试用例应有唯一的名称和标识。

2) 用例说明

简要描述测试的对象、目的和所采用的测试方法。

3) 测试初始化要求

设计测试用例时应考虑硬件配置、软件配置、测试配置、参数设置,以及其他对于测试用例的特殊说明。

4) 测试输入

测试输入是指在测试用例执行中发送给被测对象的所有测试命令、数据和信号等。对于每个测试用例应提供下列信息。

(1) 每个测试输入的具体内容(如确定的数值、状态或信号等),以及输入的性质(如有效值、无效值、边界值等)。

(2) 测试输入的来源,以及输入所使用的方法。例如,由测试程序产生、磁盘文件、通过

网络接收、人工键盘输入等。

(3) 测试输入是真实的还是模拟的。

(4) 测试输入的时间顺序或事件顺序。

5) 期望结果

期望结果是指测试用例执行中由被测软件所产生的期望结果,即经过验证,认为正确的结果。必要时,应提供中间的期望结果。期望测试结果应该有具体内容,如确定的数值、状态或信号等,不应是不确切的观念或笼统的描述。

6) 操作过程

即实施测试用例的执行步骤。把测试的操作过程定义为一系列按照执行顺序排列的相对独立的步骤,对于每个操作应提供下列信息。

(1) 每一步所需的测试操作动作、测试程序的输入、设备操作等;

(2) 每一步期望的测试结果;

(3) 每一步的评估标准;

(4) 程序终止伴随的动作或错误指示;

(5) 获取和分析实际测试结果的过程。

7) 前提和约束

在测试用例说明中施加的所有前提条件和约束条件,如果有特别限制、参数偏差或异常处理,应该标识出来,并要说明它们对测试用例的影响。

8) 测试终止条件

说明测试正常终止和异常终止的条件。

3. 测试报告

测试报告是组成测试后期工作文档的最重要的技术文档。测试报告必须包含以下重要内容。

1) 测试概述

简述测试的一些声明、测试目的、测试范围、测试方法、测试资源等。

2) 测试内容和执行情况

描述测试的详细内容,说明测试执行情况,记录的测试数据。

3) 测试结果摘要

分别描述各个测试需求的测试结果,产品实现了哪些功能点,哪些还没有实现。

4) 缺陷统计与分析

按照缺陷的属性分类进行统计和分析。

5) 测试覆盖率

覆盖率是度量测试完整性的一个手段,是测试有效性的一个度量。测试报告中需要分析代码覆盖情况和功能覆盖情况。

6) 测试评估

从总体对项目质量进行评估。

7) 测试建议

从测试组的角度为项目组提出工作建议。

在软件测试过程中需要加强过程管理和缺陷管理,并提交高质量的测试文档。软件测试相关的文档模板请参见附录 A。

1.1.4 软件测试管理工具

测试管理工具是指在软件开发过程中对测试需求、测试计划、测试用例和实施过程进行管理,并对软件缺陷进行跟踪处理的工具。通过使用测试管理工具,测试人员和开发人员可以更方便地记录和监控测试活动、阶段结果,找出软件的缺陷和错误,记录测试活动中发现的缺陷和改进建议。通过使用测试管理工具,测试用例可以被多个测试活动或阶段复用,可以输出测试分析报告和统计报表。有些测试管理工具可以支持协同操作,共享中央数据库,支持并行测试和记录,从而大幅提高测试效率。

目前市场上主流的软件测试管理工具有 Quality Center(HP)、TestManager(IBM)、SilkCentral Test Manager(Borland)、QADirector(Compuware)、TestCenter(泽众软件)、TestLink(开源组织)和 QATraq(开源组织)。

1. Quality Center

HP 公司的 Quality Center 是一个基于 Web 的测试管理工具,可以组织和管理应用程序测试流程的所有阶段,包括指定测试需求、计划测试、执行测试和跟踪缺陷。通过 Quality Center 还可以创建报告和图来监控测试流程。

Quality Center 是一个强大的测试管理工具,合理地使用 Quality Center 可以提高测试的工作效率,节省时间,起到事半功倍的效果。利用 HP-Mercury Quality Center,可以实现下列功能。

- (1) 制定可靠的部署决策。
- (2) 管理整个质量流程并使其标准化。
- (3) 降低应用程序部署风险。
- (4) 提高应用程序质量和可用性。
- (5) 通过手动和自动化功能测试管理应用程序变更影响。
- (6) 确保战略采购方案中的质量。
- (7) 存储重要应用程序质量项目数据。
- (8) 针对功能和性能测试面向服务的基础架构服务。
- (9) 确保支持所有环境,包括 J2EE、.NET、Oracle 和 SAP。

Quality Center 的前身是 Mercury Interactive(美科利)公司的 TestDirector(简称为 TD),后被 HP 公司收购,正式起名为 HP Quality Center。

网站地址: <http://www8.hp.com/us/en/software/enterprise-software.html>

2. IBM Rational TestManager

IBM Rational TestManager 是一个开放的、可扩展的框架,它将所有的测试工具、工件和数据组合在一起,帮助团队制定并优化其质量目标。其工作流程主要支持测试计划、测试设计、测试实现、测试执行和测试评估等几个测试活动。

TestManager 可以创建和运行测试计划、测试设计和测试脚本,可以插入测试用例目录

和测试用例,进行测试用例设计,对迭代阶段、环境配置和测试输入进行有效的关联。TestManager 可以创建和打开测试报告,其中有测试用例执行报告、性能测试报告,以及很多其他类的报告。除此以外,TestManager 还有很多辅助的设置,其中包括:创建和编辑构造版本、迭代阶段、计算机列表、配置属性、数据池、数据类型、测试输入类型、测试脚本类型等,还可以定制系统需要的属性。

TestManager 是针对测试活动管理、执行和报告的中央控制台,在整个项目生命周期中提供流程自动化、测试管理以及缺陷和变更跟踪功能。TestManager 具有下列功能和特性。

- (1) 支持所有的测试类型。
- (2) 定制的测试管理。
- (3) 支持本地和远程测试执行。
- (4) 建立和管理可跟踪性。
- (5) 详细的测试评估。
- (6) 生成有意义的报告。

网站地址: <http://www.ibm.com/software/rational>

3. TestLink

TestLink 是基于 Web 的测试管理和执行系统,是 sourceforge 的开放源代码项目之一。通过使用 TestLink 提供的功能,可以将测试过程从测试需求和测试设计到测试执行完整地管理起来。同时,它还提供了多种测试结果的统计和分析,使我们能够简单地开始测试工作和分析测试结果。TestLink 可以和 Bugzilla、Mantis、Jira 等缺陷管理工具进行集成。

TestLink 的主要功能包括测试需求管理、测试用例管理、测试用例对测试需求的覆盖管理、测试计划的制定、测试用例的执行、大量测试数据的度量和统计功能。

TestLink 的详细使用过程将在后面章节介绍。

网站地址: <http://www.testlink.org/>

4. SilkCentral Test Manager

SilkCentral Test Manager 是一种全面的测试管理系统,能够提高测试流程的质量和生产力,加速企业应用成功上市的速度。用户可以使用这一工具对整个测试周期进行计划、记录和管理,包括获取和组织主要业务需求、跟踪执行情况、设计最佳测试计划、调度自主测试、监视手工和自动测试的进度、查找功能缺陷以及对应用进行上市前评估。

软件开发中约 80% 的成本用于解决应用缺陷。SilkCentral Test Manager 帮助用户降低成本、加速缺陷等问题的解决。它能促成灵活多变的工作流,能够很好地与业务流程配合,将问题自动引导向下一阶段,从而优化了缺陷跟踪流程。基于 Web 的用户接口便于对中央储存器上的缺陷信息进行 24×7×365 的访问,极大地方便了分散在不同地点的工作团队的使用,促进不同部门之间的协作。同时,富有见地的报告帮助用户确定项目的进展情况。

网站地址: <http://www.borland.com/Products/Software-Testing/Test-Management/Silk-Central>