

主要特色

以唾手可得的**实际项目案例**，结合广泛流行的测试工具，辅以
全程经验结晶的视频讲解，完美地解决理论与实际的矛盾

告诉你如何对一个项目开展性能测试，是软件测试项目经
理、软件测试工程师、从业者必备工具书

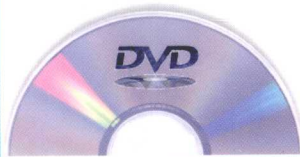


超值光盘

DVD 338分钟多媒体语音视频教程
完全与图书内容对应

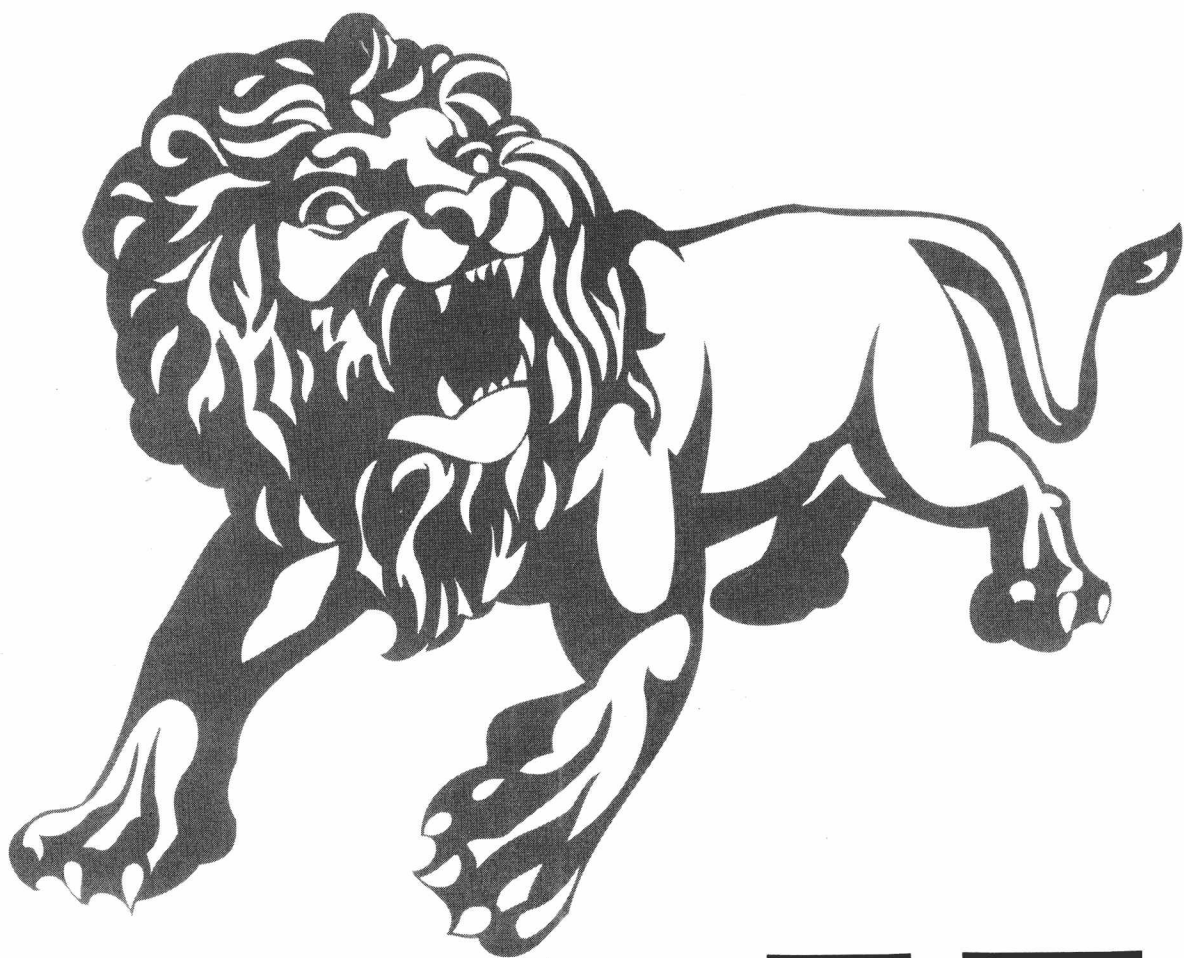
Web项目

测试实战



338分钟多媒体语音视频教程
全程语音讲解+视频操作演示

刘德宝 编著



Web项目

测试实战

刘德宝 编著

科学出版社
北京科海电子出版社
www.khp.com.cn

内 容 简 介

本书贯穿了一个实际的项目案例,从需求调研开始,直至性能测试结束,详细地剖析了软件测试工作的实施流程及过程中所使用的技术。全文共分5章,分别是:软件测试与项目分析,团队组织及任务分配,测试用例编写与管理,功能测试实施,性能测试实施。

本书是对《软件测试工程师培训教程》一书的必要补充。在《软件测试工程师培训教程》中介绍了基本的软件测试概念及技能方面的知识,本书利用其中的知识,进行实际项目的测试,将理论知识运用于实际工作中,从而掌握一般软件的测试流程、测试方法及自动化工具方面的技能。读者可根据自己的需要进行选择,有针对性地学习相关教程。

本书特别适合对软件项目测试缺乏了解的大学生和从业人员,对于想进身为软件测试项目经理的专业人员也是一本很好的参考书。

图书在版编目(CIP)数据

Web 项目测试实战/刘德宝编著. —北京:科学出版社,2009

ISBN 978-7-03-024298-3

I. W… II. 刘… III. 软件—测试 IV. TP311.5

中国版本图书馆 CIP 数据核字(2009)第 041560 号

责任编辑:陈 洁 / 责任校对:刘雪连

责任印制:科 海 / 封面设计:林 陶

科学出版社 出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

北京市科普瑞印刷有限责任公司印刷

科学出版社发行 各地新华书店经销

*

2009 年 5 月 第 一 版

开本:16 开

2009 年 5 月第一次印刷

印张:18.75

印数:0 001-3 000

字数:456 000

定价:36.00 元(含 1DVD 价格)

(如有印装质量问题,我社负责调换)

序 言

在软件测试不断发展的历程中，越来越多的人加入了软件测试的大家庭，为提高软件的质量作出了不少的贡献。但对于初学者，真正掌握软件测试的方法，目前来说还是比较困难的，市面上还没有软件项目实战方面的教程。

本书以一个实际的项目案例，从需求调研开始，直至性能测试结束，详细地剖析了软件测试工作的实施流程及测试过程中所使用的技术。全文共分 5 章，第 1 章“软件测试与项目分析”概要地介绍软件测试的基本概念与案例项目的情况；第 2 章“团队组织及任务分配”讲述了在公司中接到测试任务后测试部的工作情况；第 3 章“测试用例编写与管理”详细描述了如何利用 TestDirector 工具进行测试需求、测试用例的设计与管理等；第 4 章“功能测试实施”讲述了测试开始后功能测试的内容与实施方法；第 5 章“性能测试实施”重点讲解了如何进行性能测试需求分析、性能指标提取、性能测试用例设计、脚本录制与优化，并利用 QuickTestProfessional 生成测试数据、使用 LoadRunner 完成性能测试等，系统全面地讲述了性能测试的分析、设计与结果评价方法。除此之外，还在书后附上了常用的文档案例模板，以软件测试工作流程为主线，以软件测试技术为辅，介绍了软件如何在实际的项目中开展测试并完成功能、性能工作。

本书主要有以下几个特点：

- 本书是作者多年的工作经验总结。作者从事软件测试工作多年，以独到的视角讲解软件测试理论与实际工作的联系，从而帮助读者加深对软件测试概念的理解。
- 书中的案例采用具有代表性的业务模型 OA 系统，来源于实际公司提供的免费试用软件，在不侵害版权所有者的情况下，读者可自行下载学习。
- 书中详细地介绍了软件性能测试的方法，从性能需求分析到设计、执行，直至最终的结果分析与改进建议，完整地向读者介绍了在一个项目中如何开展性能测试。

在《软件测试工程师培训教程》（科学出版社，2008）中介绍了基本的软件测试概念及技能方面的知识，本书是对其的必要补充，用《软件测试工程师培训教程》中的知识，进行实际项目的测试，将理论知识运用于实际工作中，从而掌握一般软件的测试流程、测试

方法及自动化工具方面的技能。读者可根据自己的需要进行选择，有针对性地学习相关教程。

限于水平，错误与不妥之处在所难免，希望广大读者批评指正！在写作过程中，孙亚刚校长给予了不少帮助，在此表示感谢，同时要谢谢我夫人雪花的大力支持，她是我写作的动力。

作者

2009年4月

目 录

Chapter 01 软件测试与项目分析

1.1 软件测试概念	2
1.2 软件测试内容	3
1.3 软件测试分类	4
1.3.1 按测试方法划分	4
1.3.2 按测试阶段划分	8
1.4 软件测试流程	10
1.4.1 成立测试组	10
1.4.2 分析测试需求	10
1.4.3 制定测试计划	10
1.4.4 提取测试需求	11
1.4.5 编写测试用例	11
1.4.6 搭建测试环境	11
1.4.7 执行测试用例	11
1.4.8 跟踪处理缺陷	11
1.4.9 执行性能测试	12
1.4.10 输出测试报告	12
1.5 OA 项目分析	12
1.5.1 图书管理	14
1.5.2 考勤	14
1.6 软件项目流程	14
1.6.1 市场调研	15
1.6.2 可行性研究	16
1.6.3 产品立项	16
1.6.4 需求调研	16
1.6.5 设计开发	17
1.6.6 系统测试	17
1.6.7 产品发布	17
1.6.8 产品维护	17
1.6.9 产品升级	18

Chapter 02 团队组织及任务分配

2.1 测试团队建立	20
2.1.1 测试组长任命	20
2.1.2 测试小组建立	22
2.2 测试工作任务	23
2.2.1 测试任务分配	23
2.2.2 组员日常工作事务	24
2.3 测试计划制定	25

Chapter 03 测试用例编写与管理

3.1 测试需求提取	28
3.1.1 创建被测项目	29
3.1.2 设置项目组别	33
3.1.3 设置项目成员	37
3.1.4 创建测试需求	40
3.2 测试用例设计	47
3.3 测试用例管理	53

Chapter 04 功能测试实施

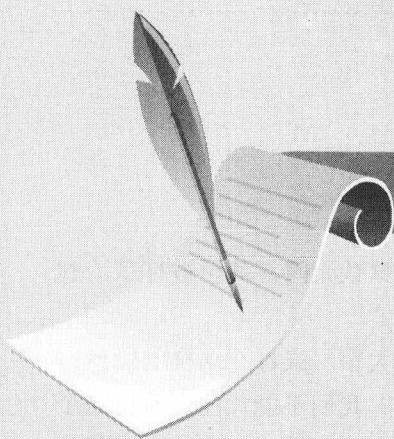
4.1 测试环境搭建	56
4.1.1 测试环境配置要求	57
4.1.2 硬件需求配置	59
4.1.3 操作系统安装与配置	59
4.1.4 JDK 安装与配置	60
4.1.5 MySQL 安装与配置	69
4.1.6 Tomcat 安装与配置	83
4.1.7 被测应用程序部署	89

4.2 测试用例执行	97
4.2.1 测试集创建	97
4.2.2 测试集执行	101
4.3 缺陷跟踪处理	106
4.4 功能测试报告输出	116

Chapter 05 性能测试实施

5.1 性能测试需求分析	124
5.1.1 性能测试基本概念	125
5.1.2 性能测试需求提取	128
5.1.3 性能测试用例设计	133
5.2 测试脚本录制与优化	135
5.2.1 测试脚本录制	135
5.2.2 测试脚本优化	143
5.2.3 测试脚本回放	187
5.3 测试场景设计与执行	190
5.3.1 测试场景分析与设计	191
5.3.2 测试场景执行	237
5.4 测试结果收集与分析	245
5.4.1 测试结果收集	245
5.4.2 测试结果分析	246
5.5 性能测试报告输出	263

附录 A 《OA 系统测试计划》	265
附录 B 《OA 系统功能测试报告》	271
附录 C 《OA 系统考勤业务模块性能测试方案》	276
附录 D 《OA 系统考勤业务模块性能测试报告》	281



Chapter

01

软件测试与项目分析

本章要点

本章简要地介绍软件测试的基本概念后，对本书所采用的实例项目进行分析。通过本章的学习，掌握软件测试的基本概念及一般的软件系统生产流程。

1.1 软件测试概念

说到软件测试，相信现在很多人都不陌生了。从前些年的默默无闻，甚至被打压，到现在的“香饽饽”，软件测试经历了一个漫长而又短暂的奋斗史。

现在很多朋友都选择了软件测试，或者即将加入软件测试大军。或许有这样的疑惑：软件测试到底是什么？为什么很多地方都在宣传软件测试好呢？我们不能片面地看待这个问题，要想知道问题的答案，就必须了解软件测试的过去与将来，这样你就有一个明确的答案，而不需别人帮你做出选择了。

记得刚上大学的时候，笔者学的是计算机专业。除了硬件知识，软件方面知道的仅是网页设计、软件开发等，并不清楚还有软件测试这一行业，对软件测试的仅有认识，也只限于软件工程课程里学到的一点软件测试知识。大三的时候，学校安排一些学生到北京的一些公司做校外实习。进入公司一个月后，发现自己不喜欢编码，于是转向了测试部。经过半年的学习与工作，渐渐地对软件测试有了新的认识（虽然那个时候，并没有多人愿意做软件测试）。2004年在长沙做需求调研的时候，又在 <http://uml.org.cn> 网站上学习了很多关于软件测试的理论。学习是个不断积累的过程，贵在坚持，有些朋友在做了一段软件测试工作后会觉得迷茫，不知道怎么发展，我想那是因为我们没有坚持自己的目标，所以觉得迷茫。

啰嗦了几句，那到底什么是软件测试呢？从字面意思来看，软件测试就是对软件的测试，那么测试又什么意思呢？“测试”在汉语词典中的解释是“测定、检查、试验”，为了检查某一事物是否已经达到我们的需求而进行的活动，可以称之为“测试”。再联系起软件，就是为了检查软件是否达到了客户的需求而进行的检查、验证活动，称之为“软件测试”。

从广义上讲，软件测试是指软件产品生存周期内所有的检查、评审和确认活动。如：设计评审、文档审查、需求测试、单元测试、集成测试、系统测试、验收测试等检查活动。从狭义上讲，软件测试是对软件产品质量的检验和评价的过程。它一方面检查、揭露软件产品质量中存在的质量问题，同时又需对产品质量进行客观的评价并提出改进意见。软件测试使用人工或自动化手段对被测对象进行确认验证活动，从而找出被测对象与最终用户需求之间的差别。在通常的软件生产活动中，软件测试贯穿于整个软件的生命周期，从初期的项目需求调研到后期的产品维护，每个阶段都离不开检查、评审与确认活动。基于不同的角度，软件测试的目的是不一样的，从用户的角度出发，普遍希望通过软件测试暴露软件中隐藏的 errors 和缺陷，以考虑是否可接受该产品。而从软件开发者的角度出发，则希望测试成为表明软件产品中不存在错误的过程，验证被测软件已正确地实现了用户的需求，确立人们对软件质量的信心。

早些年,由于缺少对软件质量的认识,导致测试工程师与开发工程师职位上的“对立”,软件开发是建设性的工作,而软件测试则是破坏性的。试想,有那么几个人,成天挑程序代码的错,找程序员的茬,程序员怎么可能觉得舒服,并且那个时候,程序员又处于主导地位,这样就造成了测试工程师地位低下、待遇低下等一些“怪异”的现象。直到现在,还有些面试官会问这样的问题,如何处理测试工程师与开发工程师的关系。如果我们知道软件测试是软件质量保证中的一个必不可少的环节,并且是极其重要的环节,软件测试与软件开发的最终目的都是生产出一个高质量的软件时,这样的问题不难回答。殊途同归,目的一样,只是方法不一样罢了,就像生活中总得有人扮演黑脸一样。

20 世纪 50 年代,软件伴随着第一台电子计算机的问世诞生了,以写软件为职业的人也开始出现,他们多是经过训练的数学家和电子工程师。20 世纪 60 年代美国大学开始出现授予计算机专业的学位,教人们写软件。早期人们在编写代码的时候,基本都是自己写,自己调试,直到 20 世纪 50 年代末,测试才与调试区分来,但由于受调试思想的影响,测试一直处于被压制状态,“为了让我们看到产品在工作,就得将测试工作往后推一点”。直到产品代码,甚至是项目后期,才开始软件测试工作。1972 年,在美国北卡罗来纳大学举行了首届软件测试正式会议。1979 年, Glenford Myers 的《软件测试艺术》(*The Art of Software Testing*)中作出了当时最好的软件测试定义:“测试是为发现错误而执行的一个程序或者系统的过程。”至此,软件测试才正式登上历史的舞台,软件测试是软件生产流程中质量保证的重要手段。

了解了软件测试的概念与发展历史后,就不难理解软件测试工作在软件生产流程中的重要性了,那么软件测试工作被重视是应该的。下面我们再来看看软件测试都做些什么事情。

1.2 软件测试内容

软件测试的定义是为了发现错误而审查软件文档、检查软件数据和执行程序代码的过程。从该定义可以看出,软件测试的对象并不仅仅是程序源代码,还包括与之相对应的文档及配置数据,那么在软件生产活动中,一般都有哪些文档呢?配置数据又都有哪些?

通常情况下,软件项目开展过程中,会有可行性报告、项目立项申请报告、项目进度安排计划、需求规格说明书、开发进度计划、测试计划、概要设计文档、详细设计文档、数据库设计文档、数据字典、源码清单、单元测试用例表等文件,配置数据主要包括系统运行所必须的基础数据,比如建库 SQL 语句、建表 SQL 语句、存储过程、数据库连接配置文件、系统初始驱动程序等。在上面众多的文档与配置数据中,测试工程师需要对这些资料进行检查、评审与确认。

前段时间有朋友问我,软件测试工程师是不是还肩负着客户培训之类的任务,回答是

肯定的。可能在我们平时看到的理论知识里，并没有看到这样的定义，但现实毕竟是现实。软件测试工程师的工作内容是繁杂的，说实在的，在公司里不要轻易与领导说“不”，让你干什么，只要不违背原则，都可以做，说句阿 Q 的话，至少你还有被使唤、利用的价值，一旦哪天你的领导不分配你任务了，同事不搭理你了，估计人事跟你“say goodbye!”的日子也不远了。软件测试工程师的工作内容取决于软件测试工程师的领导，而不依赖于软件测试定义。软件测试工程师在实际的项目活动中，做的工作往往是身兼数职，我曾经做过需求调研、原型设计、客户展示，写过用户手册、产品说明，还有报表开发等等，可以说软件生产活动中的几大工种都有涉足。做软件测试，甚至其他的事情，我们都不要做一个教条主义者，理论是服务于现实的。

软件测试份内的工作是进行项目软件功能、性能、文档、配置数据等方面的测试，除此之外，还可能需求调研、用户手册编写等等工作。软件测试工程师作为对系统最了解的人，应该担负起这些事，我们不能放弃任何一次接触整个软件、掌握业务知识的机会，要知道，经验就是这么来的，资深，资深，资者历也，深者术也。没有丰富的项目经验，哪来的资历。记得当时华为有位同事刘大姐，研发兄弟碰到业务不理解的地方，都会请教她，请她做讲解，这才叫资深。

在平时的工作中，软件测试工程师大部分时间都在根据测试用例中的步骤执行被测软件，将预期结果与软件的实际结果进行比较，从而找出被测系统中与最终用户需求不一致的地方，也就是通常意义上的 Bug。经过一轮又一轮的版本迭代，使被测软件达到预期的效果。

1.3 软件测试分类

软件测试是个广泛的概念，细究其内涵，不同的划分方法就有不同的分类，下面介绍一些常用的软件测试分类概念。

1.3.1 按测试方法划分

就像软件开发一样，软件测试同样可以采用多种方法，利用不同的方法可以得到不同的效果，并且最终保证被测对象符合预期的用户需求。按照测试方法分，主要有以下几种：黑盒测试、白盒测试、灰盒测试、静态测试、动态测试、手动测试、自动化测试。

1. 黑盒测试

黑盒测试又叫功能测试、数据驱动测试或基于需求规格说明书的功能测试，通过测试活动来检查被测对象每个功能能否正常使用，是否满足用户的需求。

黑盒测试方法能更好更真实地从用户角度来检查被测系统的界面、功能等方面需求的实现情况，但黑盒测试是基于用户需求进行的，也会带来隐患，一旦用户的需求存在问题，用黑盒测试的方法恐怕就发现不了一些特殊的缺陷了。这种问题在需求不明确或者变动频繁的时候尤为突出。所以，在实施软件测试工作时，有一个明确可靠的用户需求是非常重要的。

黑盒测试重点检查的是被测对象界面、功能方面的需求，主要的检查点包括以下几个方面。

功能不正确或遗漏

假如有了明确的用户需求，检查此类的错误轻而易举，测试工程师在测试过程中仅需根据详细的用户需求规格说明书一一检查。然而这仅是理想的情况，在现实的测试活动中，可能明确可靠的用户需求仅是一种奢望，这个时候我们又该如何进行检查呢？针对没有明确的用户需求的情况，可以从这三方面进行测试：

- (1) 由业务部门提供概要的需求文档；
- (2) 由研发部门提供 Function List（功能列表）；
- (3) 根据业务经验判断。

软件测试工程师与软件开发工程师一个明显的不同是，软件测试工程师可能在一个月內接触几个，甚至更多不同业务类型的软件项目，需要广泛的业务知识，而软件开发工程师可能在一年內仅参与一两种业务类型软件的开发工作。在没有明确的开发需求情况下，测试工程师丰富的业务知识就凸显其重要性了，而且在现实的软件开发中，期望软件生产流程很正规也是不现实的。

界面错误

与功能检查一样，查找界面错误在有明确用户需求的时候也很容易，如果没有，同样可以采用上述三个方面的检查点进行测试。当然，有个不能忽略的问题就是界面测试往往没有一个明确的标准，多数时候靠测试工程师自身的“审美”观点进行评价，这样难免有偏颇，测试工程师在提交此类的缺陷时需谨慎。界面错误一般集中在错别字、界面布局等方面。

数据访问性错误

数据访问性错误通常发生在接口上。比如，A 系统需调用 B 系统的某些数据，并设定了定时定点自动调用数据的功能，在实际工作中，随着时间的推移，经常出现不能及时调用的错误，甚至不工作。像此类的错误非常严重，特别对于那些异步处理的软件来讲，这些错误往往是致命的。记得前些年，广州的“羊城通”充值系统就存在这样的缺陷，卡已充值，可数据并未传送到服务器，造成了不小的损失。测试工程师在做这些软件测试时需多考虑各种异常访问情况，以避免在实际使用过程中出现严重的错误。

性能错误

被测对象的性能问题，往往需要进行专门的性能测试。黑盒测试阶段，我们可以从被测对象的业务响应速度、业务并发处理能力、业务成功率、系统资源耗用等方面去衡量，而不需考虑程序内部代码的质量。比如做 B/S 结构软件测试，在打开页面的时候，我们就可以明确感知到页面的展示速度，这种感知就是对被测对象响应速度的判断。

初始化和终止性错误

玩过游戏的朋友都知道，打开游戏的时候通常都有一段等待时间，游戏会加载一些运行时必须的配置信息，比如 Counter Strike（反恐精英），游戏进入地图时，会有一段等待时间，等待加载地图、角色信息等，这个过程就是初始化的过程。一旦这个过程出问题，后面也就没法玩了。所谓终止性错误，可能大家接触的比较多，比如我们常用的 Windows XP 系统，一旦某个系统文件被我们不小心修改或删除后，在打开某些文件时，就会出现一些“内存不能为 read”、“内存不能为 written”等错误，确定后文件自动关闭，像这类的问题就是典型的终止性错误。初始化与终止性错误，在安装与卸载测试活动中尤为常见，所以，在进行被测对象的安装与卸载测试时，需特别注意此类错误。

我们知道穷举法在软件测试工作中是不太适用的，在黑盒测试中，通常采用等价类、边界值、错误推断、因果图、决策表、正交试验等方法进行测试用例的设计。

等价类设计方法的意思是将被测对象的输入域划分为若干个不相交的区域，然后从这些区域中提取出一些具有代表性的数据作为测试用例，这些值在测试活动中的作用与各个区域中的其他值是等效的。举个例子，测试 $a+b$ 的算法， a 、 b 取值范围为正整数，假设 a 取 2， b 取 3，如果 $2+3$ 得出的结果是 5，那么可以认为在正整数范围内取其他值进行测试，与取 2 和 3 的效果是一样的。

边界值分析一般与等价类结合使用，该方法通过选择等价类的边界进行用例的设计，不仅仅关注输入域的边界，还考虑输出域的边界。

错误推断是前面两种方法的必要补充，资深的测试工程师不仅能使用较多的用例设计方法设计测试用例，同时能够利用自身的经验和直觉推断出被测对象中可能存在的问题，从而充实测试用例，找到更多的软件缺陷。

其他几种方法不常用，这里就不介绍了。

2. 白盒测试

与黑盒测试相对的软件测试方法，称为白盒测试。白盒测试又称结构测试、逻辑驱动测试或基于程序代码内部构成的测试。此时，测试工程师将深入考查程序代码的内部结构、逻辑设计等。白盒测试需要测试工程师具备很深的软件开发功底，精通相应的开发语言，一般的软件测试工程难以胜任该工作。

白盒测试方法主要包括代码检查法、静态结构分析法、静态质量度量法、逻辑覆盖法、

基本路径测试法，其中最为常用的方法是代码检查法。

代码检查包括桌面检查、代码审查和走查等，主要检查代码和设计的一致性，代码对标准的遵循、可读性，代码逻辑表达的正确性，代码结构的合理性等方面；发现违背程序编写标准的问题，程序中不安全、不明确和模糊的部分，找出程序中不可移植部分、违背程序编程风格的问题，包括变量检查、命名和类型审查、程序逻辑审查、程序语法检查和程序结构检查等内容。一般公司都有比较成熟的编程规范，在代码检查的时候，可以根据编程规范进行检查。

3. 灰盒测试

与前面的黑盒测试、白盒测试相比，灰盒测试介于两者之间。黑盒测试仅关注程序代码的功能性表现，不关注内部的逻辑设计、构成情况，白盒测试则仅从程序代码的内部构成考虑，检查其内部代码设计结构、方法调用等，而灰盒测试结合这两种测试方法，一方面考虑程序代码的功能性表现，另一方面，又需要考虑程序代码的内部结构。

4. 静态测试

静态测试，顾名思义就是静态的、不执行被测对象程序代码而寻找缺陷的过程。通俗地讲，静态测试就是用眼睛看，阅读程序代码、文档资料等，与需求规格说明书中的客户需求进行比较，找出程序代码中设计不合理，以及文档资料有错误的地方。

一般在企业、公司里会召开正规的评审会，通过评审的方式，找出文档资料、程序代码中存在缺陷的地方，并加以修改。

5. 动态测试

动态测试即为实际地执行被测对象的程序代码，执行事先设计好的测试用例，检查程序代码运行得到的结果与测试用例中设计的预期结果之间是否存在差异，判定实际结果与预期结果是否一致，从而检验程序的正确性、可靠性和有效性，并分析系统运行效率和健壮性等性能状况。

动态测试由四部分组成：设计测试用例、执行测试用例、分析比较输出结果、输出测试报告。

动态测试有三种主要的方法：黑盒测试、白盒测试以及灰盒测试。

6. 手动测试

在未真正接触软件测试之前，很多人都认为，软件测试工作就是执行一些鼠标单击的动作来查找缺陷。的确，在手动测试阶段，大部分的测试工作就是模拟用户的业务流程来使用软件产品，与用户需求规格说明书中的需求定义进行比较，从而发现软件产品中的缺陷。手动测试是最传统的测试方法，也是现在大多数公司都使用的测试形式。它是测试工

工程师设计测试用例并执行测试用例，然后根据实际的结果去和预期的结果相比较并记录测试结果，最终输出测试报告的测试活动。这样的测试方法，可以充分发挥测试工程师的主观能动性，将其智力活动体现于测试工作中，能发现很多的缺陷，但同时这样的测试方法又有一定的局限性与单调枯燥性。

7. 自动化测试

随着软件行业的不断发展，软件测试技术也在不断地更新，出现了众多的自动化测试工具，如惠普的 QuickTest Professional、LoadRunner，微软的 WAS，IBM 的 Rational 一套等等。所谓的自动化测试，就是利用一些测试工具，模拟用户的业务使用流程，让它们自动运行来查找缺陷。也可以编写一些代码，设定特定的测试场景，来自动寻找缺陷。自动化测试的引入，大大提高了测试的效率和测试的准确性，而且比较好的测试脚本还可以在软件生命周期的各个阶段重复使用。

1.3.2 按测试阶段划分

前面概要讲述了按测试方法划分软件测试，下面我们再按测试阶段进行划分，主要有需求测试、单元测试、集成测试、系统测试、用户测试、回归测试等。

1. 需求测试

完成需求调研后，由测试部门进行需求的测试，从需求文档的规范性、正确性等方面检查需求调研阶段生成的文档，测试工程师最好是有经验的需求分析人员，并且得到了需求调研期间形成的 DEMO。在许多失败的项目中，70%~85%的返工是由于需求方面的错误所导致的，并且因为需求的缘故而导致大量的返工，造成进度延迟、缺陷的发散，甚至项目的失败，这是一件极其痛苦的事情，所以，在有条件开展需求测试的时候，一定要实施需求测试。

2. 单元测试

单元测试又称为模块测试，顾名思义，就是对程序代码中最小的设计模块单元进行测试。单元测试是在软件开发过程中进行的最低级别的测试活动。在单元测试活动中，我们主要采用静态测试与动态测试相结合的办法。首先采用静态的代码走查，检查程序代码中不符合编程规范，存在错误或者遗漏的地方；同时使用代码审查的方法，项目小组检查项目代码，以期发现更多的问题；然后再使用单元测试工具，比如 JUnit 等工具进行程序代码内逻辑结构、函数调用等方面的测试。据业界统计，单元测试一般可以发现大约 80% 的软件缺陷。

3. 集成测试

集成测试又称为组装测试，就是将软件产品中各个模块集成组装起来，检查其接口是否存在问题，以及组装后的整体功能、性能表现。在开展集成测试之前，我们需进行深入的单元测试（当然，实际工作中大多公司不会做单元测试，仅有程序员各自检查自己的代码）。从个体来讲，可能解决了很多的缺陷，但所有的个体组合起来，就可能出现各种各样的问题。 $1+1<2$ 的问题，此刻尤为突出。

集成测试一般可采用非增式集成方法、增式集成方法（自底向上集成、自顶向下集成、组合方式集成）等策略进行测试，利用以黑盒测试为主、白盒测试为辅的测试方法进行测试。集成测试工程师一般由测试工程师担当，开发工程师将经过单元测试的代码集成后合成一个新的软件测试版本，交给配置管理员，然后测试组长从配置管理员处提取集成好的测试版本进行测试。

集成测试阶段主要解决的是各个软件组成单元代码是否符合开发规范、接口是否存在问题、整体功能有无错误、界面是否符合设计规范、性能是否满足用户需求等问题。

4. 系统测试

系统测试是将通过集成测试的软件，部署到某种较为复杂的计算机用户环境进行测试，这里所说的复杂的计算机用户环境，其实就是我们一般用户的计算机环境。

系统测试的目的在于通过与系统的需求定义作比较，发现软件与系统的定义不符合或与之矛盾的地方。这个阶段主要进行的是安装与卸载测试、兼容性测试、功能确认测试、安全性测试等。系统测试阶段采用黑盒测试方法，主要考查被测软件的功能与性能表现。如果软件可以按照用户合理期望的方式来工作的时候，即可认为通过系统测试。

系统测试过程其实也是一种配置检查过程，检查在软件生产过程中是否有遗漏的地方，在系统测试过程中做到查漏补缺，以确保交付的产品符合用户质量要求。

5. 用户测试

在系统测试完成后，将会进行用户测试。这里的用户测试，其实可以称为用户确认测试。在正式验收前，需要用户对本系统做出一个评价，用户可对交付的系统做测试，并将测试结果反馈回来，进行修改、分析。面向应用的项目，在交付用户正式使用之前要经过一定时间的用户测试。

6. 回归测试

回归测试是在发现缺陷后，重新执行测试用例的过程。回归测试阶段主要的目的是检查以前的测试用例能否再次通过，是否还有需要补充的用例等。

有些公司会采用自动化测试工具来进行回归测试，比如利用 QTP，对于产品级、变动量小的软件而言，我们可以利用这样的工具去执行测试。但一般情况下，都由测试工程师