



工业和信息化  
人才培养规划教材

Industry And Information  
Technology Training  
Planning Materials

# 软件测试项目实战 之功能测试篇

Software Test Project

简显锐 杨焰 胥林 © 主编  
韩艳 刘静 刘德宝 © 副主编

- + 采用**项目教学法**，学生在实际项目中学习软件  
**测试知识**
- + **校企合作**编写教材，体现企业**软件测试岗位**的  
技能要求



中国工信出版集团



人民邮电出版社  
POSTS & TELECOM PRESS



工业和信息化  
人才培养规划教材

Industry And Information  
Technology Training  
Planning Materials

# 软件测试项目实战 之功能测试篇

Software Test Project

简显锐 杨焰 胥林 © 主编

韩艳 刘静 刘德宝 © 副主编

人民邮电出版社

北京

## 图书在版编目(CIP)数据

软件测试项目实战之功能测试篇 / 简显锐, 杨焰, 胥林主编. — 北京: 人民邮电出版社, 2016.9  
工业和信息化人才培养规划教材  
ISBN 978-7-115-39793-5

I. ①软… II. ①简… ②杨… ③胥… 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(2015)第151623号

## 内 容 提 要

教材采用项目教学方法, 围绕一个真实的腾图办公自动化(OA)系统平台项目展开测试任务。主要内容包括 OA 系统项目分析, 项目测试流程, 项目测试计划与方案, 项目测试需求分析, 被测模块测试用例设计, 功能模块自动化测试, 缺陷报告及管理; 重点是运用 HP 公司的 QTP 自动化测试工具完成了 OA 项目相关功能模块的自动化测试内容。

本书既可作为高等院校软件测试专业的教材, 又可作为社会培训机构的培训教材, 同时也适合从事软件测试工作的读者自学参考。

---

◆ 主 编 简显锐 杨 焰 胥 林  
副 主 编 韩 艳 刘 静 刘德宝  
责任编辑 马小霞  
责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号  
邮编 100164 电子邮件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
三河市海波印务有限公司印刷

◆ 开本: 787×1092 1/16  
印张: 14.5 2016年9月第1版  
字数: 329千字 2016年9月河北第1次印刷

---

定价: 39.80 元

读者服务热线: (010) 81055256 印装质量热线: (010) 81055316  
反盗版热线: (010) 81055315

# 前言 FOREWORD

目前我国软件的规模日益增大,企业对软件产品的质量要求越来越高,对软件测试人员的需求也越来越大。为了满足企业对软件测试人才的需求,各级各类院校纷纷开设了软件测试的课程。我们通过多年的教学研究发现,目前的教材理论知识过多,学生的技能没有达到企业软件测试岗位要求,知识与技能,未能很好地融合,因此院校老师与企业工程师共同打造了这本校企合作教材,旨在使软件测试教学更符合学生的认知规律,提高学生学习质量与效率,以期培养出具备良好职业素养的专门人才。

本书采用项目教学模式,以职业活动和工作过程为导向,以一个复杂、大型、综合的项目为基础,实现知识、理论和实践的整体设计,是一本具有综合性、实战性的一体化教材。教材中的项目是经过作者学校专业教师团队与企业工程师多次交流讨论后,从企业已经完成的若干个项目选择的,具有代表性,企业工程师全程参与了教材的编写过程。

教材内容和结构的安排基于企业真实的软件测试工作。第1章首先介绍软件测试的基本概念,然后引入企业的软件测试工作流程,最后介绍这本书将要完成的测试工作项目OA系统测试。第2章重点讲测试计划和测试方案文档编写和技术分析。第3章重点讲OA系统测试环境的搭建。第4章重点讲软件测试的基础能力——黑盒(手工)测试能力,在本章中将完成测试需求提取、测试用例设计、测试用例执行、缺陷跟踪处理以及测试报告输出等测试工作环节的训练任务。经过第2章~第4章的任务操作,就



已经完成了一次完整的测试工作，具备了初级测试员的能力。第 5 章是对软件测试能力的提升训练，重点讲自动化测试工具：HP 公司的 QTP 测试工具，本章内容依然是以 OA 系统测试工作为导向，训练学生使用工具进行自动化功能测试的能力。

本书由简显锐、杨焰和胥林任主编，韩艳、刘静和刘德宝任副主编。其中简显锐编写第 4 章，杨焰编写第 1 章，胥林编写第 5 章，韩艳和刘静编写第 2 章，刘德宝编写第 3 章。本书在编写过程中参考了许多文献及成果，在此表示深深的敬意和诚挚的感谢。由于编者水平有限，错漏之处在所难免，敬请广大读者、专家批评指正。

编者

2016 年 5 月

## 第1章 软件测试与项目分析 1

1.1 软件测试概念	2	1.4.4 提取测试需求	8
1.2 软件测试内容	3	1.4.5 编写测试用例	8
1.3 软件测试分类	3	1.4.6 搭建测试环境	9
1.3.1 按测试方法划分	3	1.4.7 执行测试用例	9
1.3.2 按测试阶段划分	6	1.4.8 跟踪处理缺陷	10
1.4 软件测试流程	7	1.4.9 执行性能测试	10
1.4.1 成立测试组	8	1.4.10 输出测试报告	10
1.4.2 分析测试需求	8	1.5 OA 系统分析	10
1.4.3 制订测试计划	8	1.6 本章练习	14

## 第2章 测试计划与测试方案 15

2.1 软件测试流程	16	2.3.1 测试方案目的	24
2.2 测试计划	16	2.3.2 测试方案内容	24
2.2.1 测试计划目的	16	2.3.3 测试方案编写	25
2.2.2 测试计划内容	16	2.3.4 测试方案示例	26
2.2.3 测试计划示例	17	2.4 本章练习	29
2.3 测试方案	24		

## 第3章 OA 系统测试环境搭建 30

3.1 测试环境搭建流程	31	3.2.4 JDK 安装与配置	35
3.2 测试环境搭建	31	3.2.5 MySQL 安装与配置	42
3.2.1 测试环境配置要求	32	3.2.6 Tomcat 安装与配置	49
3.2.2 硬件需求配置	34	3.2.7 被测应用程序部署	53
3.2.3 操作系统安装	34	3.3 本章练习	59

## 第4章 OA 系统测试项目实施 60

4.1 测试需求提取	61	4.3 设置项目组别	72
4.1.1 原始测试需求分析	61	4.4 设置项目成员	75
4.1.2 测试项分析	63	4.5 创建测试需求	78
4.1.3 测试子项分析	63	4.6 测试用例设计	81
4.2 创建被测项目	69	4.6.1 测试用例定义	81

# CONTENTS

## 目录

4.6.2	用例设计方法	84	4.7.2	测试集执行	116
4.6.3	OA 用例设计	107	4.8	缺陷跟踪处理	119
4.7	测试用例执行	114	4.9	测试报告输出	124
4.7.1	测试集创建	114	4.10	本章练习	128

### 第 5 章 自动化测试与 QTP 129

5.1	自动化测试简介	130	5.3.4	变量与参数化	164
5.2	QTP 简介	131	5.3.5	描述性编程	168
5.2.1	QTP 安装	131	5.3.6	VBS 自动化编程	170
5.2.2	QTP 产品介绍	137	5.4	QTP 自动化实践	200
5.2.3	QTP 实现原理	139	5.4.1	脚本开发流程	200
5.3	QTP 功能基础	141	5.4.2	录制脚本	200
5.3.1	对象与对象库	141	5.4.3	增强脚本	203
5.3.2	录制与回放	142	5.5	本章练习	220
5.3.3	检查点	145			

### 附录 软件测试常见面试题 221



# 第 1 章 软件测试与项目分析

本章从软件测试概念开始，介绍当下软件测试的主要内容，通过对软件测试分类的内容介绍引入软件测试流程，最后在掌握相关内容的基础上，对贯穿全书的办公自动化（Office Automation,OA）系统做出了分析说明。

## 学习目标

- 了解软件测试概念及内容
- 理解软件测试分类及流程
- 了解被测试 OA 系统相关内容



## 1.1 软件测试概念

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，测试才与调试区分开来，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”。直到产品代码写完，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。

对于软件而言，测试是通过人工或者自动的检测方式，检测被测对象是否满足用户要求或弄清楚预期结果与实际结果之间的差异，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。软件测试是质量检测过程，包含了若干个测试活动。

早些时候，很多人对软件测试的认识仅限于运行软件执行测试，实际上软件测试还包括静态测试和验证活动。软件包括实现用户需求的源代码、描述软件功能及性能表现的说明书，支撑软件运行的配置数据，软件测试对象同样包括了文档及配置数据的测试，不仅是执行软件。

软件测试工程师职责定义的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。软件测试活动是对软件产品质量的检验和评价的过程。一方面检查、揭露软件产品质量中存在的质量问题，另一方面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。

了解软件测试发展历史与基础概念后，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，软件测试对象并不仅是程序源代码，还包括与之相对应的文档及配置数据，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、源代码清单、测试用例等。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。

软件测试核心工作是实施软件系统功能、性能、文档、配置数据等方面的测试活动，除此之外，还有可能有需求调研、用户手册编写等工作。日常测试工作中，测试工程师经常利用测试用例执行被测软件，利用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测试，使被测软件达到预期质量要求。

## 1.3 软件测试分类

从测试方法来看，软件测试可分为黑盒测试、白盒测试、灰盒测试、静态测试、动态测试、手工测试、自动化测试等几个方面，从测试阶段来分，可分为需求测试、单元测试、集成测试、系统测试、验收测试等几个阶段。

### 1.3.1 按测试方法划分

与软件开发有若干框架一样，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。按照测试方法划分，主要有以下几种。

#### 1. 黑盒测试

黑盒测试又称功能测试、数据驱动测试或基于需求规格的功能测试，通过测试活动来检查被测对象每个功能能否正常使用，是否满足用户需求。

黑盒测试方法能更好更真实的从用户角度来检查被测对象界面、功能等方面需求实现情况，但黑盒测试基于用户需求进行，会带来隐患。黑盒测试方法难以发现一些隐藏在程序内部的缺陷，如内存泄露等。

以 OA 系统为例，如果从用户需求角度考虑，对图书管理、资产管理或车辆管理等模块，实施功能或性能测试，此处的方法即为黑盒测试。

黑盒测试工作目前是软件测试方法中的核心方法，在企业测试过程中，大多数采用

黑盒测试方法，读者在学习过程中需要重点学习此测试方法，再辅以后续的测试方法，才能很好地掌握软件测试技术。

## 2. 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，测试工程师需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，初级测试工程师难以胜任该工作。

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；代码检查还要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，代码检查时，可以根据编程规范进行检查。

以 OA 系统车辆管理添加车辆功能为例，如果对以下代码

```
function findObj(theObj, theDoc)
{
    var p, i, foundObj;
    if(!theDoc) theDoc = document;
    if( (p = theObj.indexOf("?")) > 0 && parent.frames.length)
    {
        theDoc = parent.frames[theObj.substring(p+1)].document;
        theObj = theObj.substring(0,p);
    }
    if(!(foundObj = theDoc[theObj]) && theDoc.all) foundObj = theDoc.all[theObj];
    for (i=0; !foundObj && i < theDoc.forms.length; i++)
        foundObj = theDoc.forms[i][theObj];
    for(i=0; !foundObj && theDoc.layers && i < theDoc.layers.length; i++)
        foundObj = findObj(theObj,theDoc.layers[i].document);
    if(!foundObj && document.getElementById) foundObj = document.getElementById(theObj);
    return foundObj;
}
var GetDate="";
function SelectDate(ObjName,FormatDate){
    var PostAtt = new Array;
    PostAtt[0]= FormatDate;
    PostAtt[1]= findObj(ObjName);
    GetDate=showModalDialog("../util/calendar/calendar.htm", PostAtt ,
```

```
"dialogWidth:286px;dialogHeight:221px;status:no;help:no;");  
}  
function SetDate()  
{  
    findObj(ObjName).value = GetDate;  
}
```

进行测试，验证 findObj、SetDate 等函数的功能，此类方法即为白盒测试方法。

### 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，不关注其内部逻辑设计、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。灰盒测试则综合了黑盒测试与白盒测试，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。

同样，以 OA 系统为例，如果在测试过程中，既考虑车辆管理用户需求方面的特性，如能否添加车辆、编辑车辆信息等，又从该功能的实现逻辑代码考虑，则此方法即为灰盒测试。

### 4. 静态测试

静态测试，顾名思义，静态的、不执行被测对象程序代码寻找缺陷的过程。通过阅读程序代码、文档资料等，与需求规格说明书进行比较，找出程序代码中设计不合理以及文档资料有错误的地方。

在实际研发活动中可开展同行评审活动，通过评审方式，找出文档资料、程序代码中存在的缺陷并加以修改。

以 OA 系统为例，如果针对该系统的设计文档，如概要设计文档，或系统源代码进行走读查阅，则使用的是静态测试方法。

### 5. 动态测试

动态测试即为执行被测对象程序代码，执行测试用例，检查程序运行实际结果与测试用例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。

以 OA 系统为例，搭建测试环境运行系统对其进行功能的验证测试，即为使用动态测试方法。

### 6. 手工测试

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，

使用软件产品，与用户需求规格进行比较，从而发现软件系统中的缺陷。手动测试是最传统的测试方法，也是目前大多数公司都在使用的测试形式。测试工程师设计测试用例并执行测试用例，根据实际结果与预期结果相比，记录测试结果，最终输出测试报告。手工测试，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，但手工测试方法又有一定的局限性，并且长期下去会令人觉得枯燥单调。

## 7. 自动化测试

软件行业不断发展，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 HP 的 QucikTest Professional、LoadRunner，IBM RPT、RFT 等。自动化测试是利用一些测试工具，录制业务使用流程，让工具自动运行测试过程查找缺陷，也可以编写脚本代码，设定特定的测试场景，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，而且写出结构性较好的测试脚本，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

前面概要阐述了按测试方法划分的软件测试类型，下面以测试阶段对测试类型进行划分，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

需求调研完成后，测试部门或者需求小组进行需求测试，从需求文档规范性、正确性等方面检查需求调研阶段生成的需求文档，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，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，主要采用静态测试与动态测试相结合的方法。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，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，交由配置管理员，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地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

回归测试一般发生的情况在发现缺陷后，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，可以利用这样的工具去执行测试。但一般情况下，都由测试工程师手动地执行以前的测试用例，来检查用例通过情况。

回归测试可以发现在产品发布前未能发现的问题，比如时钟的延迟、软件的性能问题等。

## 1.4 软件测试流程

在学习了软件测试的基本概念后，接着介绍一般公司里软件测试的流程。以 OA 系统为例，一般公司的软件测试工作流程如图 1-1 所示。

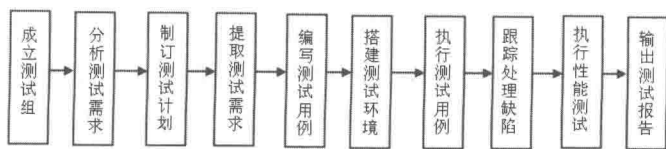


图 1-1 软件测试工作流程图

### 1.4.1 成立测试组

当需测试的项目分配下来后，该项目的负责人向测试部门提出测试申请，通过测试经理的审批后，由测试经理指派测试组长与测试工程师，成立项目测试组，负责该项目的测试工作。

根据项目团队的组织流程，OA 系统的负责人张三向测试部门经理李四申请实施测试活动，此时，李四接受测试任务后，制定王五、赵六、田七为测试工程师，负责 OA 系统的实际测试活动。

### 1.4.2 分析测试需求

测试经理任命测试组长，测试组长需提前熟悉被测对象的需求，从总体上掌握项目的进展情况。通过仔细地阅读项目的相关文档（比如项目的进度计划、测试要求等）后，测试组长需安排下一步工作。

### 1.4.3 制订测试计划

测试组长在详细了解项目信息后，根据项目需求、项目进度计划表制定当前项目的测试计划，并以此测试计划来指导测试组开展对应的测试工作。测试计划中需说明每个测试工件输出的时间点、测试资源、测试方法、测试风险规避、测试停测标准等。

### 1.4.4 提取测试需求

测试组长制订好了测试计划后，项目组进行评审。评审通过后，项目测试组即可按照此测试计划开展工作。测试组员根据测试组长的任务分配，进行项目用户需求规格说明书的阅读，甚至开展需求测试工作。需求阅读理解完成后，进行测试需求的提取，也就是列出被测对象需测试的点，这项工作可以利用 TestDirector 等测试管理工具开展。

本阶段的工作，因某些项目周期及要求不同，可能不做要求。

### 1.4.5 编写测试用例

测试需求提取完毕，经过测试组的评审通过后，测试组员可以进行测试用例的设计，这些工作都是在测试计划中规定的时间内完成。比如测试计划中规定“2008-12-20~2008-12-30 完成系统测试用例设计及评审”，那么就必须在这个时间段内完成被测对象的测试用例设计。测试用例的设计一般使用 Word、Excel 等样式，也可使用 TestDirector、TestLink 等工具进行管理。

测试用例设计工作在某些企业中因项目周期及要求不同，可能不开展，直接进行测试活动。

## 1.4.6 搭建测试环境

测试用例设计工作完成后，如果项目开发组告知测试组长可以开展测试的时候，测试组长可从配置管理员处提取测试版本，根据开发组提供的被测对象测试环境搭建单进行测试环境的搭建。测试环境搭建需要测试工程师掌握基本的硬件、软件知识。

随着用户需求的不断加大，项目运行环境往往非常复杂，并且搭建成本极高，以大型网站系统架构为例，如图 1-2 所示。

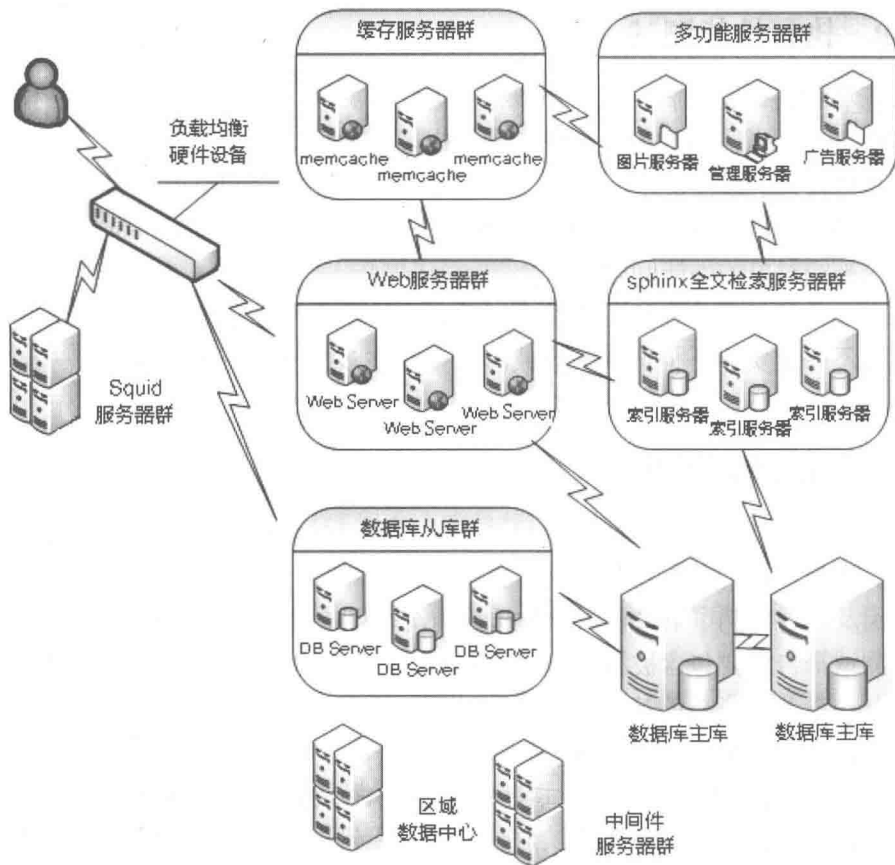


图 1-2 大型 Web 系统架构

从用户角度来看，该服务器的架构非常复杂，从测试人员角度来看，同样站在用户角度，也不需要掌握其复杂架构，测试环境一般都由开发人员搭建，所以该环节的工作测试人员不一定实施。

## 1.4.7 执行测试用例

测试环境搭建完成后，测试组员将进行测试用例的执行。根据前期设计并评审通过的测试用例，测试组员进行各个功能模块的测试。在执行测试用例的过程中，如果发现遗漏或者不完善的测试用例，需及时做更新，并用文档记录变更历史。用例执行过程中如果发现了 Bug，则需按照部门或者项目组的 Bug 提交规范，利用一些 Bug 管理工具提交 Bug。常用的 Bug 管理工具有 Bugzilla、TestTrack、Mantis、TestDirector 等。

### 1.4.8 跟踪处理缺陷

大多数公司都有自己的 Bug 管理流程规范，项目组成员需根据这个流程规范开展日常的 Bug 处理工作。在缺陷处理阶段，大多要经过 4 次、甚至更多的迭代过程，多次进行回归测试，直到在规定的时间内达到测试计划中所定义的停测标准为止。

在这个阶段，主要使用黑盒测试方法开展工作，以被测对象的需求规格说明为依据，重点关注被测对象的界面与功能表现。

### 1.4.9 执行性能测试

一般在功能测试完成后，还需开展相应的性能测试工作。与功能测试一样，在测试之前，需要进行测试需求的分析，性能指标提取、用例设计、脚本录制、优化、执行、分析等一系列过程。通过使用一些自动化工具进行性能测试是目前性能测试的主要手段，常用的性能测试工具有 WAS、QALoad、WebLoad、LoadRunner、Robot 等。性能测试阶段主要解决被测对象的性能问题。

目前大部分项目软件在执行功能测试后，可能不进行性能测试，所以本过程在实际项目测试时不一定实施，但面向大众或涉及多用户多并发的业务系统时，一定会开展性能测试活动。

### 1.4.10 输出测试报告

功能测试、性能测试都完成后，测试组长需要对被测对象做一个全面的总结，以数据为依据，衡量被测对象的质量状况，并提交测试结果报告给项目组，从而帮助项目经理、开发组及其他部门了解被测对象的质量情况，以决定下一步的工作计划。

功能测试报告主要包含被测对象的缺陷修复率、Bug 状态统计、Bug 分布等，性能测试报告主要包含测试指标的达标情况及测试部的质量评价等。当然，也可以出一份整体的测试报告，包含功能、性能测试结果。

综上所述，测试需求分析、测试计划制订、执行测试、跟踪处理缺陷、编写测试报告等测试活动，在任何项目中都会实施，而测试用例设计、测试环境搭建、性能测试等活动则可能根据项目需求不一样不一定实施。

## 1.5 OA 系统分析

通过上面几部分的介绍，已经了解了软件测试的基本概念，软件测试工作的常用流程等。从本节起，正式进入本书的实战部分，以实际的项目实例介绍软件测试工作。

现在软件行业中有很多业务类型，大多数公司招聘时都需要测试工程师具备丰富的项目经验，那么这些项目经验怎么来呢？

这里介绍一个常用的方法。对于软件测试初学者，一个比较好的方法是利用网络下载一些程序源代码，根据这些资料中配备的环境配置说明，自己练习部署、源代码阅读、业务理解等，如果在环境配置、程序应用过程中出现问题的话，可以通过网络查找相关