[测试岗定级考试大纲](http://192.168.105.101:8090/pages/viewpage.action?pageId=8979113)

1. 第一部分 软件工程与软件测试基础知识

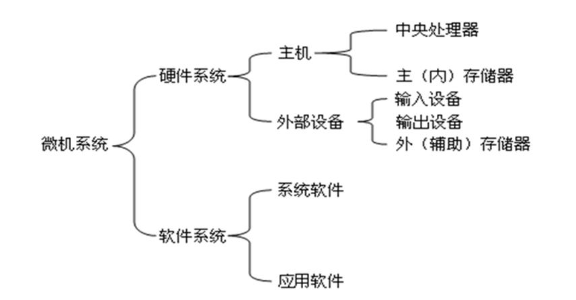
1.1. 计算机系统基础知识

* 了解计算机系统的构成

由运算器、控制器、存储器、输入/输出设备、外存储器组成。

（运算器和控制器又称为中央处理器，即CPU）

\*存储器分为内部存储器和外部存储器，内部存储器，如：只读存储器ROM（如BIOS）、随机存储器RAM（如DDR内存条） 外部存储器，如：软盘、硬盘、U盘、光盘等。



* 了解基本输入输出设备

\*输入设备：鼠标、键盘、摄像头、扫描仪、条形码阅读器、触摸屏、光笔等。

\*输出设备：打印机、显示器、绘图仪等。

* 理解二进制、十进制、十六进制数值转换

\*计算机采用二进制存储来存储数据和代码。所有的数据都必须通过某种方式转换成二进制，然后才能存入计算机。

\*二进制：数字后加B，如10010B

八进制：数字后加O，如：123O

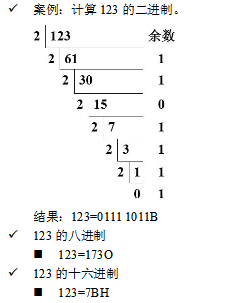
十进制：数字后加D或不加，如：10D或10

十六进制：数字后加H，如2A5EH



\*十进制转二进制：十进制数除以2，倒排余数。

（十进制转其他进制，做法一致，转几进制则除以几，然后倒排余数）

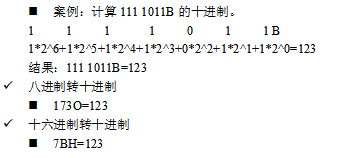


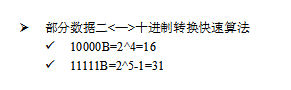
\*二进制转十进制：按权展开。（权：进制数的幂，代表每一位数字的重要性）

二进制的权：1、2、4……

十进制的权：1、10、100……

（其他进制转十进制，做法一致，按权展开）





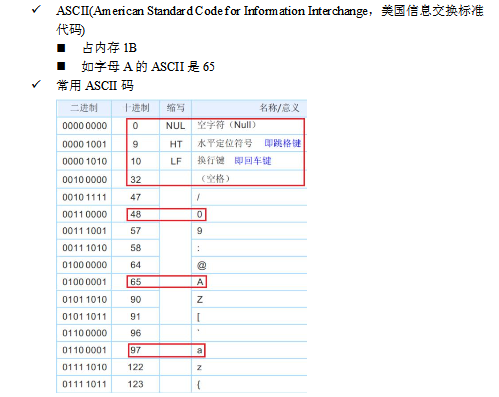
\*常用的进制权

二进制权：1、2、4、8、16、32、64

八进制权：1、8、64、512

十六进制权：1、16、256

\*常用的ASCII码



1B=8bits

1KB=1024B

1MB=1024KB

1GB=1024MB

1TB=1024GB

（生活中，1KB=1000B，如U盘）

1.2. 数据库基础知识

* 了解数据库主键、外键、索引、存储过程
* 熟练使用基本SQL语句及一种数据库客户端工具
* 熟练操作一种数据库的创建、备份、还原操作
* 理解数据库连接字符串参数配置

1.3. 信息安全知识

* 了解信息安全基本概念
* 了解国家计算机信息系统安全保护等级划分准则、等级要求
* 了解MD5单向加密、盐值概念
* 理解SQL注入原理
* 熟练掌握用户注册、登录、验证码等测试

1.4. 软件工程知识

* 理解敏捷测试流程
* 理解Scrum、UserStory、backlog、Kanban概念
* 了解数据流图、程序流程图、系统流程图文件编制符号及约定

1.5. 软件开发风险基础知识

* 理解风险管理
* 理解风险防范及应对

2. 第二部分 软件测试应用技术

2.1. 软件测试基本概念

* 理解软件质量与软件测试

\*软件质量：软件特性的总和，软件满足规定或潜在用户需求的能力。（从软件在内部、外部和使用中的表现来衡量）

\*软件测试：在规定的条件下对程序进行操作，以发现错误，对软件质量进行评估。

\*区别：软件测试是软件工程不可获缺的重要环节之一，是软件质量保证的重要手段。

它们是软件质量工程的两个不同层面的工作。

质量保证的重要工作通过防预、检查与改进来保证软件质量，主要着眼于软件开发活动中的过程、步骤和产物，而不是对软件进行剖析找出问题或评估。

测试关心的不是过程的活动，而是对过程的产物以及开发出的软件进行剖析，如对开发文档和源代码进行走查，运行软件以找出问题，报告质量。

* 理解软件测试定义

在规定的条件下对程序进行操作，以发现错误，对软件质量进行评估。

* 理解软件测试目的

发现软件缺陷与错误，并尽可能以最少的人力、物力和时间找出更多的软件中所潜在的各种错误和缺陷，通过修正来提高软件质量。对软件质量进行度量和评估。

测试是为了发现错误而执行程序的过程；

测试是为了证明程序有错，而不是证明程序无错；

一个好的测试用例在于能发现至今未发现的错误；

一个成功的测试是发现了至今未发现的错误的测试。

* 理解软件测试原则

\*所有的软件测试都应追溯到用户需求；

\*应当尽早地和不断地进行软件测试；（应尽早介入，从需求分析开始）

\*完全测试是不可能的，测试需求终止；

\*测试无法显示软件潜在的缺陷；（测试只能证明软件存在错误而不能证明软件没有错误）

\*充分注意测试中的群集现象；（二八原则：软件测试中存在Pareto原则：80%的缺陷发现在20%的模块中。）

\*程序员应避免检查自己的程序；

\*尽量避免测试的随意性；（应制定测试计划，分清测试重点，并严格执行测试计划）

\*测试用例应当定期修订和评审，增加新的或不同的测试用例帮助发现更多的缺陷，应包含合理的输入条件和不合理的输入条件；（Pesticide Paradox杀虫剂悖论）

\*妥善保存测试计划、测试用例、出错统计和最终分析报告，为维护提供方便。

* 理解软件测试对象

\*软件包括程序、数据和文档，所以软件测试不仅仅是对程序进行测试，它贯穿于整个软件生命周期中。

需求分析、概要设计、详细设计及程序编码等各阶段所得到的文档，包括需求规格说明、概要设计规格说明、详细设计规格说明以及源程序，都是软件测试的对象；编码结束后，对各个程序模块进行单元测试；在模块集成后，进行集成测试；在集成测试后，需要检测与证实软件是否满足需求规格说明书的要求，进行确认测试；最后将整个程序模块集成为软件系统，安装在运行环境下，进行系统测试。

3. 软件测试类型

按开发阶段划分为：单元测试、集成测试、系统测试、确认测试、验收测试。

按测试实施组织划分为：开发方测试（а测试）、用户测试（β测试）、第三方测试。

按测试技术划分为：白盒测试、黑盒测试、灰盒测试。或者静态测试和动态测试。

* 理解单元测试、集成测试、系统测试

\*单元测试：又称模块测试，是针对软件设计的最小单位，即对程序模块进行正确性检验的测试工作。

目的在于检查每个程序单元能否正确实现详细设计说明中的模块功能、性能、接口和设计约束等要求，以发现各模块内部可能存在的各种错误。

单元测试需要从程序的内部结构出发设计测试用例。

多个模块可以平行地独立进行单元测试。

\*集成测试：也叫组装测试，在单元测试基础上，将所有程序模块进行有序的、递增的测试。

集成测试是检查程序单元或部件的接口关系，逐步集成为符合概要设计要求的程序部件或整个系统。

软件集成过程是一个持续的过程，会形成多个临时版本。

每个版本提交时都要进行冒烟测试，即对程序的主要功能进行测试。冒烟测试也叫版本验证测试、提交测试。

\*系统测试：是为了验证和确认系统是否达到其原始目标，而对集成的硬件和软件系统进行的测试。

系统测试是在真实或模拟系统运行的环境下，检查完整的程序系统能否和系统正确配置、连接，并满足用户需求。

* 理解确认测试、验收测试

\*确认测试：通过检测和提供客观证据，证实软件是否满足特定预期用途的需求。

（是检测与证实软件是否满足软件需求说明书中规定的要求）

\*验收测试：按照项目任务书或合同、供需双方约定的验收依据文档进行的对整个系统的测试与评审，决定是否接收或拒收系统。

* 理解开发方测试、用户测试、第三方测试

\*开发方测试：也叫验证测试或а测试，在软件开发环境下，由开发者检测与证实软件的实现是否满足软件设计说明或软件需求说明的要求。

\*用户测试：也叫β测试，在用户的应用环境下，由用户通过运行和使用软件，来检测与核实软件实现是否符合自己预期的要求。（一般指用户的使用性测试而不是验收测试）

\*第三方测试：也称独立测试，介于软件开发方与用户方之间的测试组织的测试。一般在模拟用户真实应用环境下，进行软件确认测试。

* 理解动态测试、静态测试

\*静态测试：指不运行程序，通过人工对程序和文档进行分析与检查，包括：走查、符号执行、需求确认等。（实际上是对软件的需求说明书、设计说明书、程序源码等进行非运行的检查）

\*动态测试：通过人工或使用工具运行程序进行检查、分析程序的执行状态和程序的外部表现。

* 理解白盒测试、黑盒测试、灰盒测试

\*白盒测试：又称结构测试或逻辑驱动测试，通过对程序内部结构的分析、检测来寻找问题。

\*黑盒测试：又称功能测试，通过测试来检测每个功能是否都能正常使用。通过软件的外部表现来发现其缺陷和错误，完全不考虑程序内部结构和处理过程。

（在程序界面处进行测试，只是检查样序是否按照需求规格说明书的规定正常实现，但如果外部特性本身有问题或规格说明的规定有误，用黑盒测试则发现不了）

黑盒测试注重于测试软件的功能需求，主要试图发现以下几类错误： 功能不正确或遗漏；界面错误；数据库访问错误；性能错误；初始化和终止错误等。

黑盒测试只有采用穷举输入测试，才能查出程序中所有的错误。但完全测试是不可能的，所以要针对性地进行测试，通过制定测试计划并严格执行测试计划来有效进行。

\*灰盒测试：介于白盒和黑盒之间的测试。主要关注输出对于输入的正确性，同时也关注内部表现，但不像白盒测试那样详细完整。它结合了白盒和黑盒测试的要素。

注：白盒、黑盒和灰盒测试，在实现测试方法上既包括了动态测试，也包括了动态测试。

软件测试贯穿整个软件生命周期。一般开发文档和源程序可以应用单元测试应用走查的方法；单元测试可应用白盒测试方法；集成测试可应用近似灰盒测试方法；系统测试和确认测试可应用黑盒测试方法。

3.1. 开发与运行阶段的测试

* 理解单元测试
* 理解集成测试
* 理解系统（确认）测试
* 理解验收测试

3.2. 测试用例设计方法

**3.2.1. 白盒测试设计**

* 了解白盒测试基本技术
* 了解白盒测试方法

**3.2.2. 黑盒测试用例设计**

* 理解测试用例设计方法

\*测试用例：就是将软件测试的行为活动，作一个科学化的组织归纳。 （即，测试用例就是设计一个情况，软件程序在这种情况下，必须能够正常运行并且达到程序所设计的执行结果。如果在此情况下不能正常运行且问题会重复发生，则表示有缺陷，需将此问题标注出来反馈给开发人员修改，版本更新后，需要利用同一个用例来测试之前的问题，确保问题已被修复）

\*设计测试用例的目的：为了能将软件测试行为转换为可管理的模式。

（测试用例可将测试行为具体量化，以便进一步让管理阶层掌握所需要的测试过程）

\*使用测试用例的好处：可避免盲目测试并提高测试效率；

令软件测试的实施重点突出、目的明确；

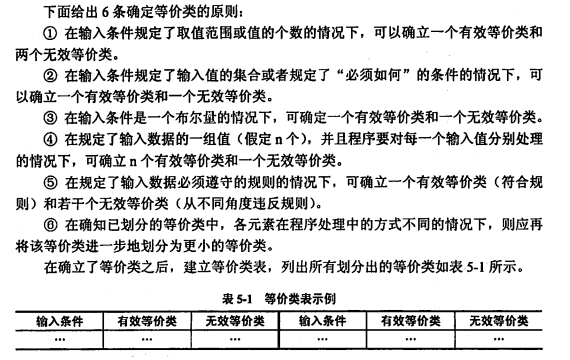
版本更新后只需修正少部分的用例便可展开测试工作，降低工作强度，缩短项目周期；

功能模块的通用化和复用化使软件易于开发，测试用例的通用化和复用化使软件测试易于开展，并随着测试用例的不断精化，其效率也不断攀升。

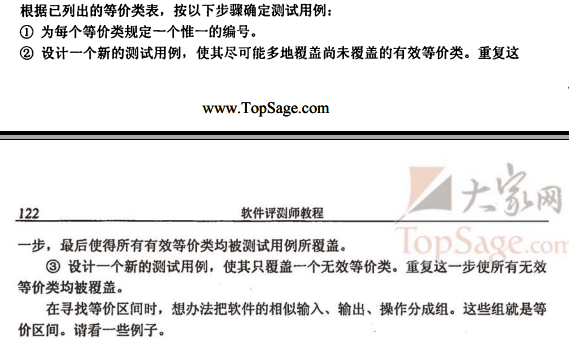
\*黑盒测试用例设计方法包括：等价类划分法、错误推测法、因果图法、判定表驱动法、 正交试验设计法、功能图法等。

\*等价类划分法：把程序的输入域划分成若干部分，然后从每个部分中选取少数代表性数据作为测试用例。

步骤：1）必须在分析需求规格说明的基础上划分等价类，列出等价类表，如：有效等价类和无效等价类；



2）确定测试用例



（完全不考虑程序内部结构，只根据需求规格说明书，仔细分析和推敲，然后把说明书中对输入的要求和输出的要求区别开来并加以分解）

\*边界值分析法：相对输入等价类和输出等价类而言，稍高于其边界值及稍低于其边界值的一些特定情况。

1）边界条件

2）

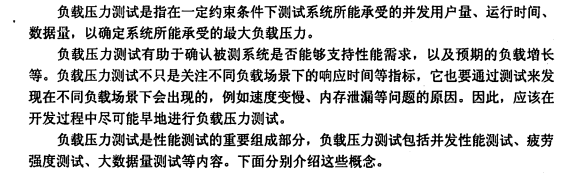
* 理解测试用例的编写

**3.2.3. 测试方法选择的策略**

* 理解黑盒测试方法选择策略
* 了解白盒测试方法选择策略

**3.2.4. 负载压力测试**

* 理解性能/压力测试基本概念



* 理解性能/压力测试解决方案

\*并发性能测试

\*疲劳强度测试

\*大数据量测试

* 理解性能/压力测试指标分析

\*交易处理性能指标

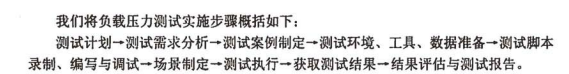
\*服务器操作系统资源监控

\*数据库资源监控

\*web服务器监控

\*中间件服务器监控

* 了解性能/压力测试实施



3.3. Web应用测试

* 理解链接测试

链接是web应用系统的一个主要特征，它是在页面之间切换和指导用户去一些不知道地址的页面的主要手段。

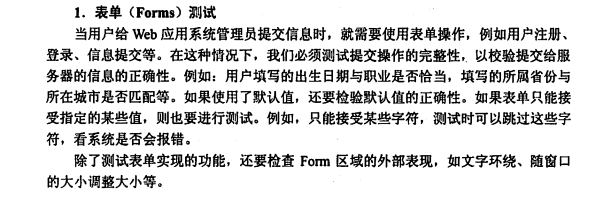
\*分为三个方面：首先，测试所有链接是否按指示那样确实链接到了该链接的页面；

其次，测试所链接的页面是否存在；

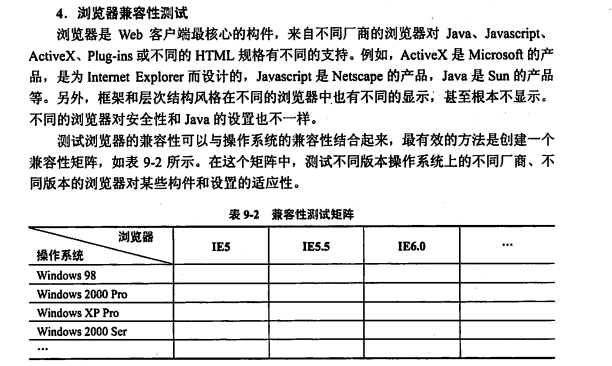
最后，保证web应用系统上没有孤立的页面，即没有正确链接指向该页面。

测试时，可逐一检查链接的有效性、可达性、正确性等。

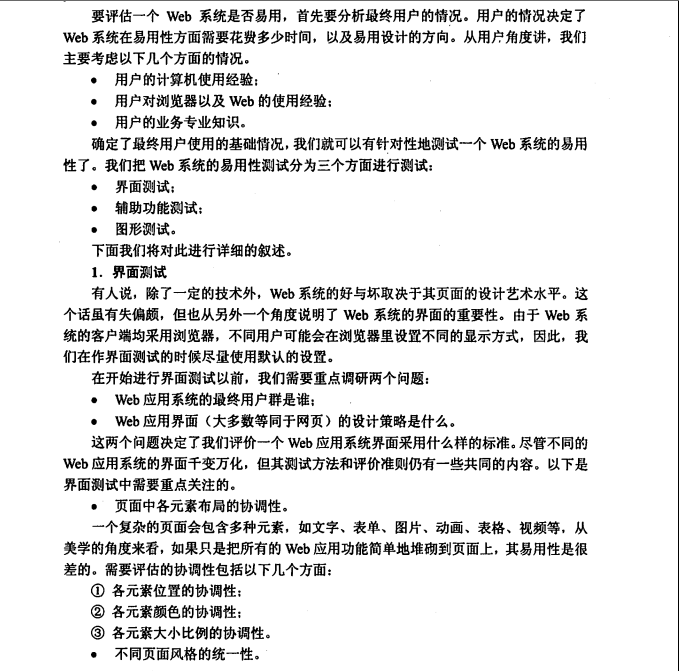
* 理解表单测试

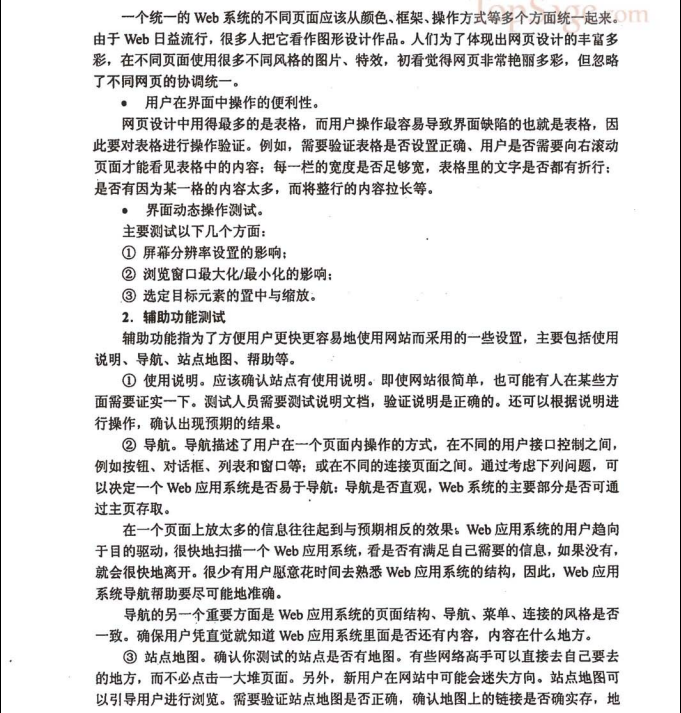


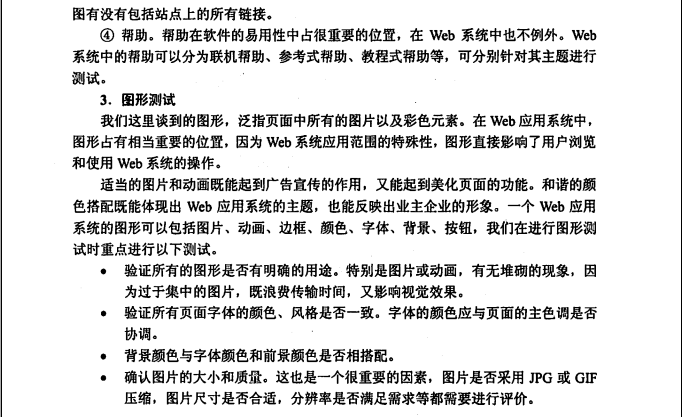
* 理解浏览器兼容性测试



* 理解其他测试（界面输入、风格、布局、式样、颜色搭配、操作体验、提示信息、热键、文字内容、国际化、导航……）



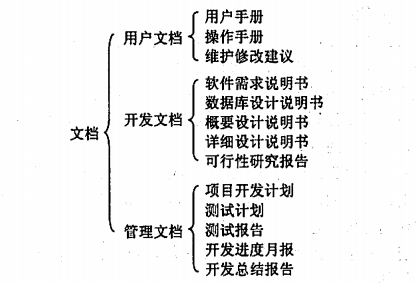


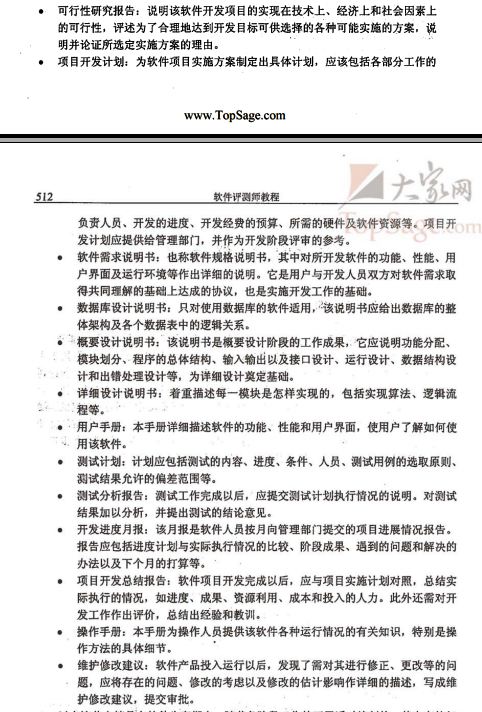


3.4. 文档测试

* 理解文档测试的范围

\*软件产品由可运行的程序、数据和文档组成。文档是软件的一个重要组成部分。

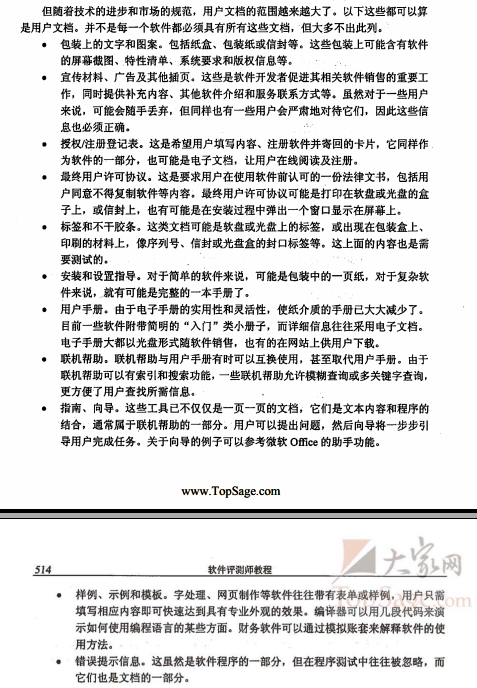




* 理解用户文档的内容

当用户文档仅包含一个Readme文件时，只需对此进行拼写检查，确认涉及到的技术准确无误，最多对Readme文件进行病毒扫描，确保其不带病毒即可。

一般用户文档内容包括：包装上的文字图案；宣传材料、广告及其他插页；授权/注册登记表；最终用户许可协议；标签和不干胶条；安装和设置指导；用户手册；联机帮助；指南、向导；样例、示例和模板；错误提示信息。



用户文档的作用：改善易安装性、提高软件的易用性、改善软件可靠性、促进销路、降低技术支持的费用。

* 理解用户文档测试的要点

\*读者群

\*术语

\*正确性

\*完整性

\*一致性

\*易用性

\*图表与界面截图

\*样例和示例

\*语言

\*印刷与包装

* 理解用户手册的测试

\*准确按照手册的描述使用程序

\*尝试每一条建议

\*检查每条陈述

\*查找容易误导用户的内容

* 理解在线帮助的测试

很大程序上与用户手册测试相同，但帮助并不只是用户手册的电子版，还应包含以下几点补充说明：

\*准确性

\*帮助是文档编写和程序编写的结合

\*帮助索引

\*超链接

\*链接的意义

\*帮助的风格

考试题型：选择题、填空题、案例简答题。

其中：第一部分 软件工程与软件测试基础 占比30%。

第二部分 软件测试应用技术 占比70%。