软件测试

第6章 Web测试

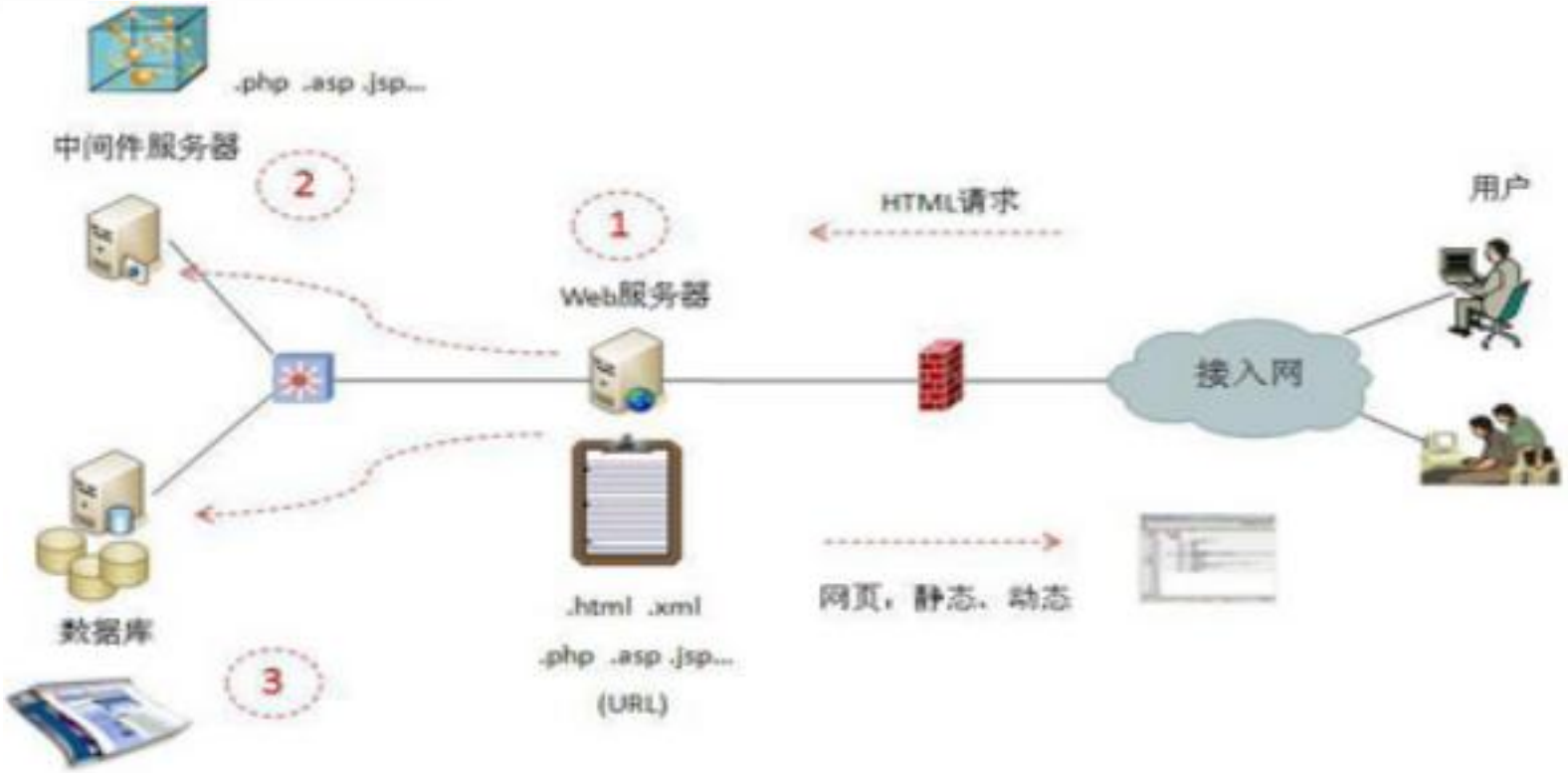
范围与目标

- 课程目标：
  - 了解Web的测试的分类
  - 掌握Web相关的测试知识
  - 掌握Web测试用例的设计方法

本章内容

- ***6.1*** WEB网站的特性
- 6.2 Web测试设计

Web应用程序原理



Web系统结构示意图

Web页面特点

新闻资讯 校情总览 机构设置 院部设置 教育教学 科学研究 学生工作 人才招聘 招生就业 合作交流 校园文化 English

UIS 学生会 学生社团 学生社区 学子之家 团苑在线 社会实践

学校召开选派“第一书记”欢送会 1 2 3 4 5

快速通道

在线邮件	网上办公	会议安排	常用电话	网站地图
图 书 馆	党务公开	校务公开	校长信箱	校友总会
成绩查询	课表查询	学历认证	学生选课	学生社区
网络服务	科大校历	科大地图	大学导航	旧版主页

科大要闻 | 校园传真 | 媒体科大 >>更多

第六届全国高校“校长杯”桥牌邀请赛在我校举行	04月16日
我校首次岗位设置管理与聘用工作启动，校长马连湘提出要求	04月13日
青科大荣获六项青岛市科学技术奖	04月13日
副校长罗公利在全省高校学生工作会议上作典型发言	04月16日
我校信息公开工作会议召开	04月14日
李镇江、魏红卫荣获首届“青岛高校教学名师”称号	04月12日

校园公告 | 校内通知 | 学术动态 >>更多

关于办理我校2011年度专业技术职务资格备案的通知	04月16日
青岛科技大学2011年度“自立自强”标兵名单公示	04月11日
关于转发山东省教育厅《关于推荐2012年孔子学院/课堂和国家	04月11日
关于认真组织学习宣传《高等学校章程制定暂行办法》的通知	04月10日
高性能聚合物及成型技术教育部工程研究中心设备招标采购	04月06日

WEB网站的特点

- 1.网络集约性
- 2.内容驱动性
- 3.持续演化性
- 4.即时性
- 5.安全性
- 6.美观性

1.网络集约性

就本质而言，一个Web网站是网络集约的。它可以驻留在网络上，并且服务于变化多样的客户群的需要。例如时下流行的门户网站或者网络游戏。它们都可以看成一个完善的大型Web应用系统，服务于各种客户群，但其本身只需要一个服务器端，用各式各样的客户端满足不同要求的客户。

2.内容驱动性

一般来说，Web网站不是为了某个或某些特定用户量身定做的，它们一般都拥有一个广大的服务群体，其服务的内容，往往由这些群体的要求所决定。在大多数情况下，一个Web网站的主要功能是使用HTML（超文本标记语言）javascript等语言来表示文本、图形、音频、视频内容给终端用户。

3.持续演化性

不同于传统的、按一系列规律发布进行演化的应用软件（如微软每隔1-2年发布新的Office办公软件），Web网站一般是采取持续演化的模式。对于某些Web应用而言，按小时为单位进行更新都是司空见惯的。

4.即时性

- Web网站具有其他任何软件类型中都没有的即时性，或者称为快速性。对于某些较大规模的Web网站，开发时间往往也只有几周或者几天，适度复杂的Web页面可以仅在几小时内完成。这要求开发者必须十分熟练于开发Web应用所需的压缩时间进度的规划、分析、实现以及测试方法。

5.安全性

Web网站通过网络访问，为了提高系统效率，需要限制访问终端的用户的数量。为了保护敏感内容，必须提供安全的数据传输模式。因此要求Web网站必须有一定的安全性保障。

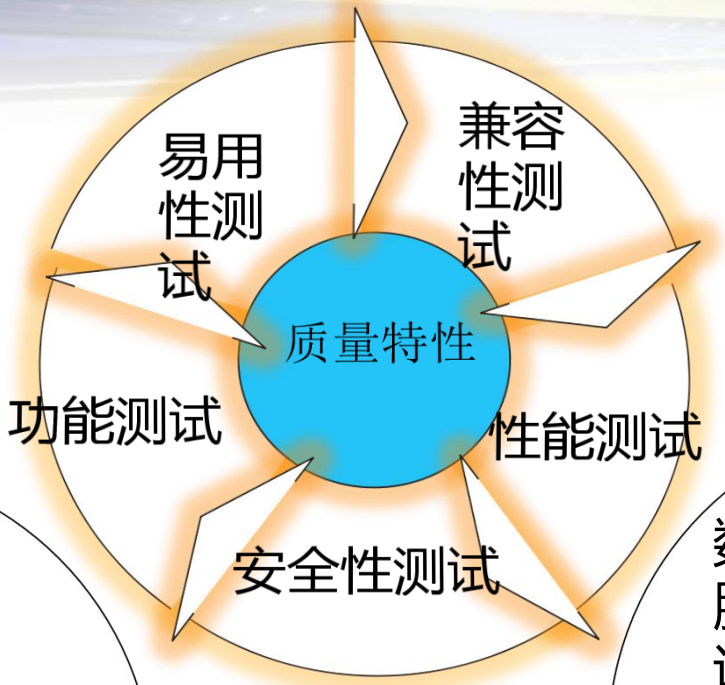
6.美观性

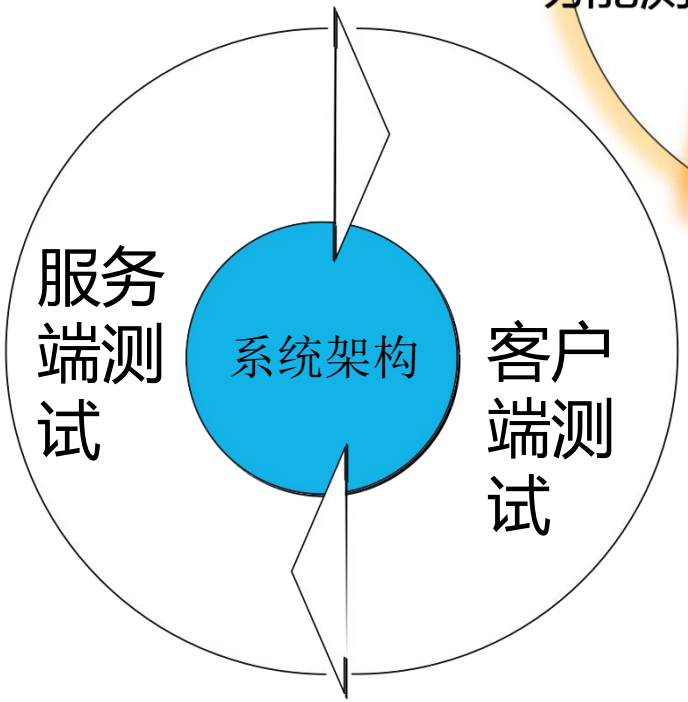
良好的观感会使一个Web网站锦上添花。在某种应用已经被市场广泛接受或者定义为标准时，美观性可能和技术在同样程度上影响该应用的成功。

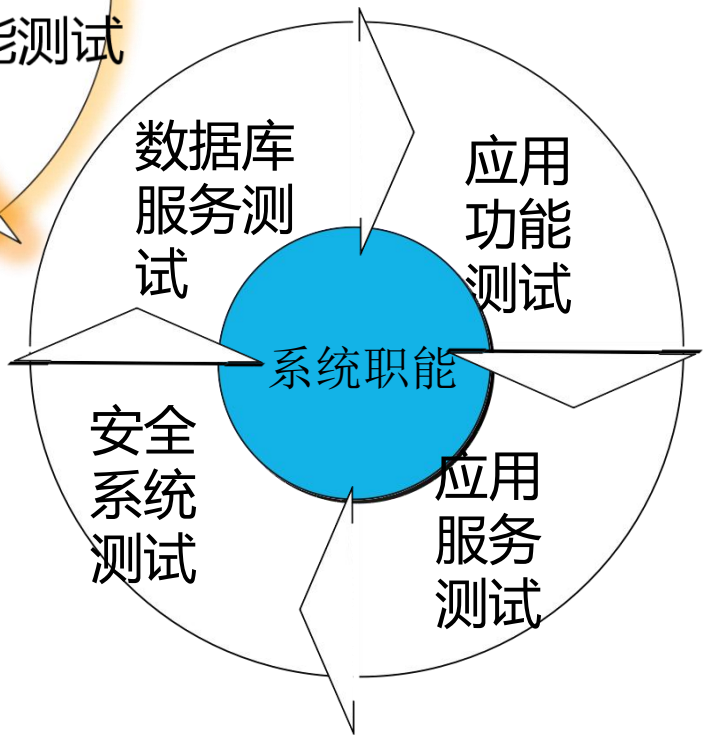
从Web特点到Web测试

图形化	美观	图形测试	内容测试
易于导航	易用	链接测试	导航测试
平台无关	兼容	平台测试	浏览器测试
分布式	资源	SSL测试	Cookies测试
动态的	更新	接口测试	数据库测试
交互的	会话	表单测试	安全性测试

Web测试框架分类


易用性测试
兼容性测试
质量特性
功能测试
性能测试
安全性测试


服务端测试
系统架构
客户端测试


数据库服务测试
应用功能测试
系统职能
安全系统测试
应用服务测试

Web测试框架示意图

功能测试

- 数据库测试
- 设计语言
- 状态保持：Session、Cache、Cookies
- 表单与数据校验
- 链接测试

测试用例

兼容性测试

- 网络环境
- WEB服务器
- 分辨率
- 打印机
- 浏览器
- 操作系统

测试环境

性能测试

- 压力测试
- 负载测试
- 链接速度测试

性能指标

易用性测试

- 整体界面测试
- 内容测试
- 图形测试
- 导航测试

用户体验

安全性测试

- 防DOS攻击
- 日志文件
- 登录
- SSL
- 目录设置

专家专业评估

6.2 Web测试用例设计

- Web功能性测试用例设计
- Web性能测试用例设计
- Web易用性测试用例设计
- Web兼容性测试用例设计
- Web安全性测试用例设计

1、Web功能性用例分类：

- 链接测试
- 表单与数据校验测试
- 状态保存测试
  - Session
  - Cache
  - Cookies
- 数据库操作测试

链接测试

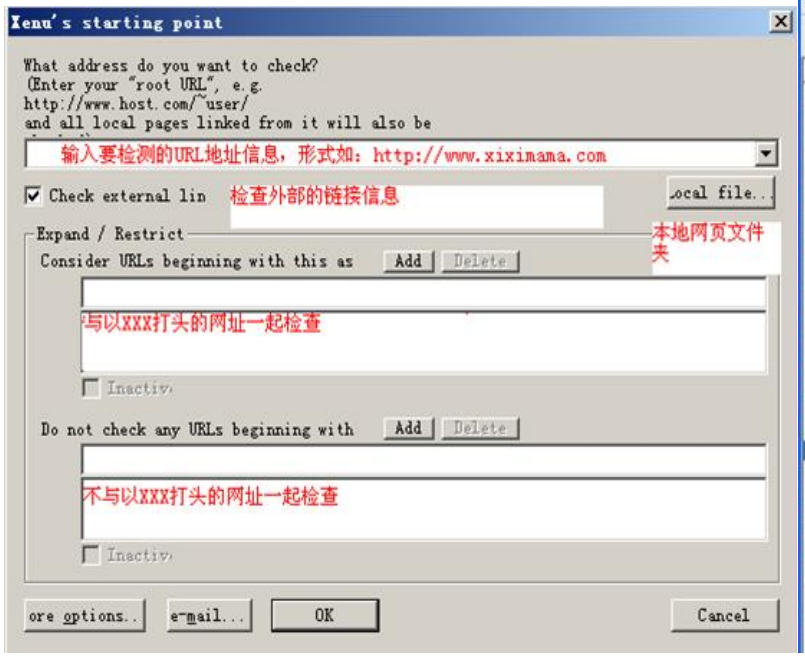
- 链接是Web应用系统的一个主要特征，它是在页面之间切换和指导用户去一些不知道地址的页面的主要手段。
  - 按链接的表现形式分：文字、图像、图标、按钮等
  - 按链接的编写方式分：静态链接、动态生成的链接、自动跳转的链接等
  - 按链接的类型分：HTTP、FTP、news、Gopher等
  - 按链接的地址所在分：内部链接、外部链接等
  - 按链接的打开方式分：在框架内打开、刷新页面、新开窗口、新开模式窗口等

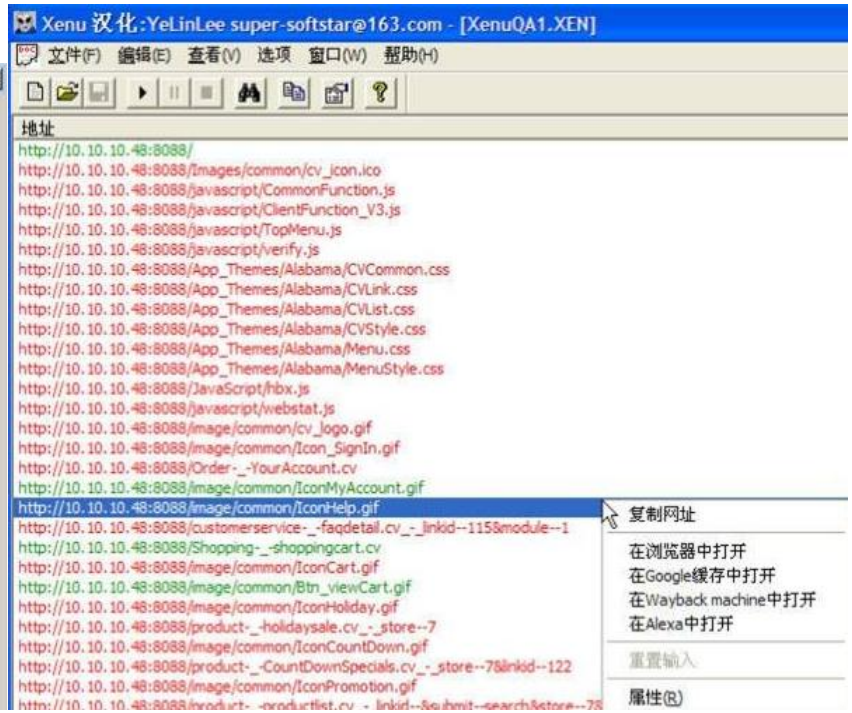
链接测试用例设计

- **用例设计思想：**
- 首先，测试所有链接是否按指示的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，所谓孤立页面是指没有链接指向该页面，只有知道正确的URL地址才能访问。
  - 链接的显示
  - 链接跳转的结果
  - 链接访问的页面是否存在
  - 是否有孤立的页面存在
- **自动化的链接检测工具**
  - Xenu Link Sleuth
  - HTML Link Validator
  - Web Link Validator

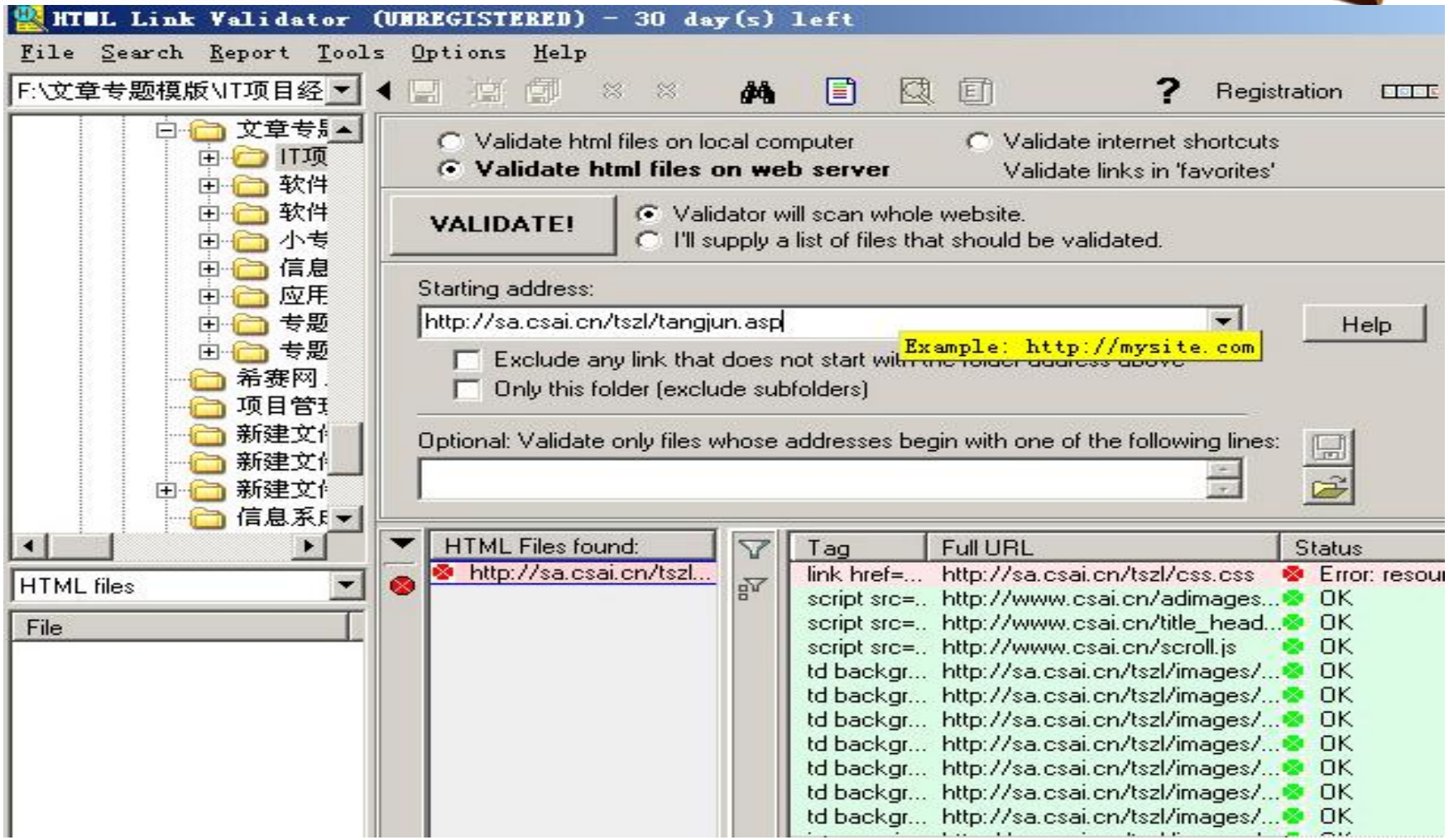
链接测试工具Xenu Link Sleuth

- 你可以打开一个本地网页文件来检查它的链接，也可以输入任何网址来检查。它可以分别列出网站的活链接以及死链接，连转向链接它都分析得一清二楚；它支持多线程，可以检查结果存储成文本文件或网页文件。Xenu无需安装，支持asp、do、jsp等结尾的网页，同时能够生成html格式的测试报告。





链接测试工具HTML Link Validator



表单与数据校验

- 表单，在网页中经常使用，主要负责数据采集的功能，比如你可以采集访问者的名字和E_mail地址、调查表、留言簿等等。
- 一个表单有三个基本组成部分：
  - 表单标签：这里面包含了处理表单数据所用CGI程序的URL以及数据提交到服务器的方法。（＜form＞＜/form＞）
  - 表单域：包含了文本框、密码框、隐藏域、多行文本框、复选框、单选框、下拉选择框和文件上传框、网格Grid等。
  - 表单按钮：包括提交按钮、复位按钮和一般按钮；用于将数据传送到服务器上的CGI脚本或者取消输入，还可以用表单按钮来控制其他定义了处理脚本的处理工作。

表单与数据校验测试用例设计

- **用例设计思想：**
- 应尽量利用测试用例设计的方法：边界值分析、等价类划分等
  - 显示：加载、缺省值、快捷方式、提示信息等
  - 单个控件：文本框、下拉列表、文件上传、日期框、检查框 、滚动条、网格、ActiveX控件等
  - 组合控件：几个下拉框、日期范围等
- 工具：QTP，Winrunner,Robotform

Session测试及用例设计

- 含义：指一类用来在客户端与服务器端之间保持状态的解决方案。

  Session，中文经常翻译为会话，其本来的含义是指有始有终的一系列动作/消息，比如打电话是从拿起电话拨号到挂断电话这中间的一系列过程可以称之为一个Session

- **用例设计思想：**
  - 登录后的权限
  - 注销后的再次登录
  - Session超时
  - 一终端多用户和多终端一用户等

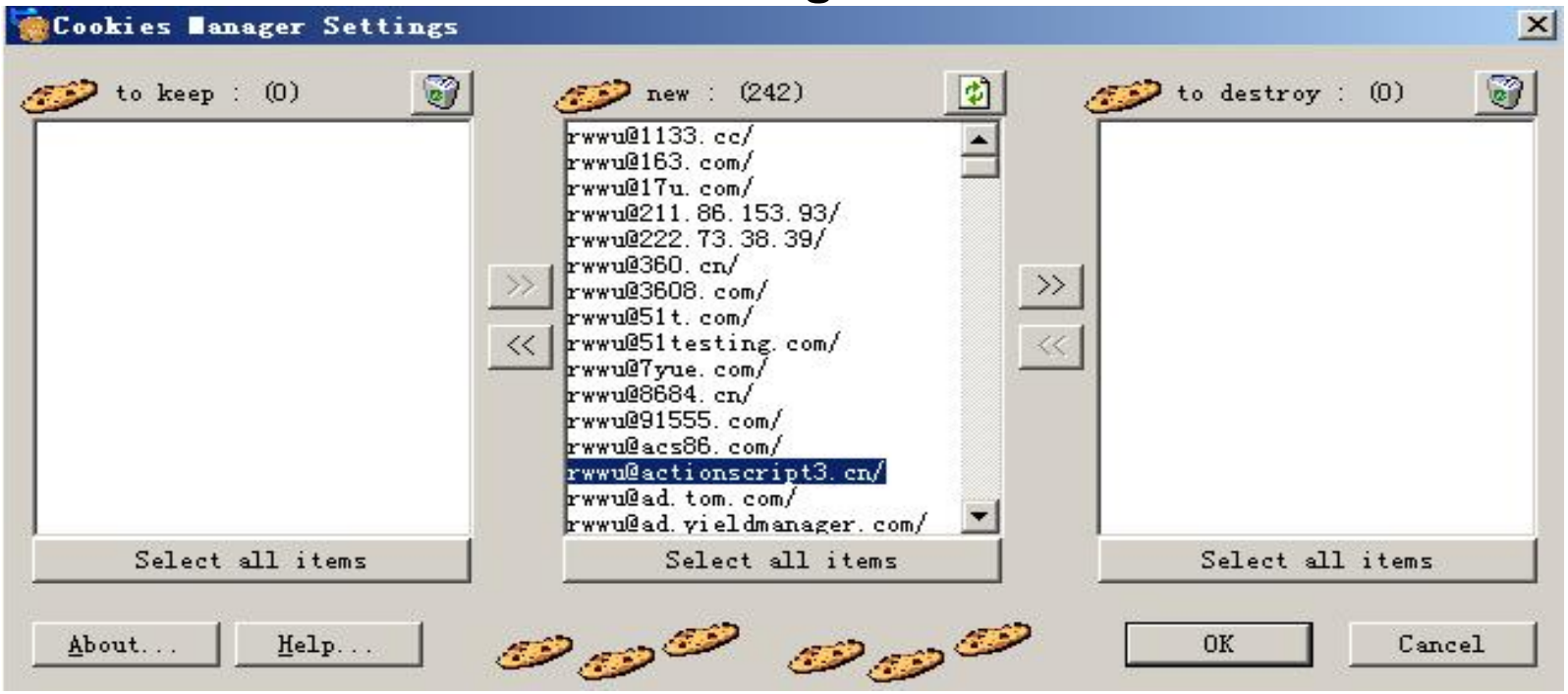
Cookies测试及用例设计

- 含义： 一种能够让网站服务器把少量数据储存到客户端的硬盘或内存，或是从客户端的硬盘读取数据的一种技术。
- 作用：用于自动登录

- **用例设计思想：**
  - Cookies的加密
  - 自动登录
  - 失效时间
  - 更改密码等

Cookies Manager

- Cookie是存在于您硬盘里的小文件，只要是您浏览过的网站，大都会留下这样的文件在您的电脑里头，当您再次光临该网站时，该网站就会立刻辨认您的身份，加快您进入的速度。而有些网站甚至可以很聪明的进入之前所浏览的网页中，充分做到个人化的服务。因为它记录了您的一些资料，可以用**Cookies Manager**帮您管理Cookie。



Cookie测试用例设计

编号	SEC_Web_ SESSION_02
测试用例名称	Cookie存储方式测试
测试目的	某些Web应用将SesssionId放到了URL中进行传输，攻击者能够诱使被攻击者访问特定的资源，例如图片。在被攻击者查看资源时获取该SessionID（在HTTP协议中Referer标题头中携带了来源地址），从而导致身份盗用。
用例级别	1
测试条件	1. 已知Web网站地址 2. Web业务运行正常 3. Web业务存在登陆认证模块 4. 已知正确的用户名、口令
执行步骤	1. 登录系统。 2. 请求不同的业务应用 3. 观察URL。
预期结果	URL中没有携带Session ID信息（可能是sid,JSESSIONID等形式）。
备注	
测试结果	

Cache测试及用例设计

- 含义：Cache即高速缓冲存储器(Cache Memory)，用来保存浏览过页面的一种机制。
  - 在WEB应用中，缓存机制也是相当重要的。比如你打开IE，第一次打开是很慢的，但是关闭后马上再打开就快很多，这是因为这时数据还没被系统“请”出内存，系统从内存中直接取得数据自然快了
  - Cache的作用就是缓存浏览过的页面,图片等,比如刚刚看过A页面了,现在看的B页面,如果还想看A页面,直接点后退,A页面就可以从Cache中装入,而不用再连接网络下载了。
- **用例设计思想：**
  - 缓存是否起作用
  - 数据修改后是否能及时刷新
  - 缓存失效时间是否正确

数据库测试

- 在Web应用技术中，数据库起着重要的作用，数据库为Web应用系统的管理、运行、查询和实现用户对数据存储的请求等提供空间。在Web应用中，最常用的数据库类型是关系型数据库，可以使用SQL对信息进行处理。
- 在使用了数据库的Web应用系统中，一般情况下，可能发生两种错误，分别是数据一致性错误和输出错误。
  - 数据一致性错误主要是由于用户提交的表单信息不正确而造成的，
  - 输出错误主要是由于网络速度或程序设计问题等引起的，针对这两种情况，可分别进行测试。

数据库测试用例设计

用例设计思想：

- 增加、修改、查询数据，注意会造成字段约束、默认值、重复数据等问题
- 注意删除数据的关联情况
- 数据的并发访问冲突
- 数据库的压力测试
- 数据库的备份恢复问题

小贴士：最好使用真实数据测试

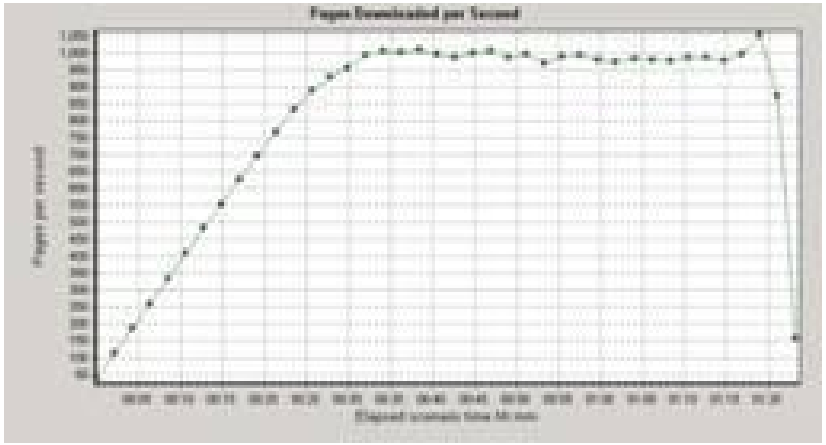
2、Web性能测试

- 连接速度测试
  - 用户连接到Web应用系统的速度根据上网方式的变化而变化，他们或许是电话拨号，或是宽带上网。当下载一个程序时，用户可以等较长的时间，但如果仅仅访问一个页面就不会这样。如果Web系统响应时间太长（例如超过5秒钟），用户就会因没有耐心等待而离开。
  - 另外，有些页面有超时的限制，如果响应速度太慢，用户可能还没来得及浏览内容，就需要重新登陆了。而且，连接速度太慢，还可能引起数据丢失，使用户得不到真实的页面。

负载测试

❖ 负载测试

- 负载测试是为了测量Web系统在某一负载级别上的性能，以保证Web系统在需求范围内能正常工作。
- 负载级别可以是某个时刻同时访问Web系统的用户数量，也可以是在线数据处理的数量。例如：Web应用系统能允许多少个用户同时在线？如果超过了这个数量，会出现什么现象？Web应用系统能否处理大量用户对同一个页面的请求？



压力测试

❖压力测试

- 负载测试应该安排在Web系统发布以后，在实际的网络环境中进行测试。因为一个企业内部员工，特别是项目组人员总是有限的，而一个Web系统能同时处理的请求数量将远远超出这个限度，所以，只有放在Internet上，接受负载测试，其结果才是正确可信的。
- 进行压力测试是指实际破坏一个Web应用系统，测试系统的反映。压力测试是测试系统的限制和故障恢复能力，也就是测试Web应用系统会不会崩溃，在什么情况下会崩溃。黑客常常提供错误的数据负载，直到Web应用系统崩溃，接着当系统重新启动时获得存取权。
- 压力测试的区域包括表单、登陆和其他信息传输页面等

性能测试用例设计

功能	系统支持多个用户并发访问
目的	测试多用户访问系统时，系统的处理能力
方法	模拟多个用户在不同客户端访问系统，然后进行并发访问系统的操作

并发用户数与事务执行情况

并发用户数	事务平均响应时间	事务最大响应时间	事务成功率	平均流量
10				
50				
100				
…				

3、Web易用性测试

- “易用性”是一个衡量标准，用来衡量使用一个产品完成指定任务的难易程度。
- “易用性Usability（又被译为可用性）”这个词在软件开发中表现为这样一种方式，即把用户而非系统置于开发过程的中心。这种被称为“以用户为中心进行设计”的概念，是指从设计过程的开端便把用户所关注的东西包含于其中，并规定用户应该是任何设计决定中最重要的因素。

易用性测试用例设计

- 导航---我可以很容易找到在哪
- 帮助和支持---当我需要时我能得到帮助
- 工作流支持---我可以按照自己的方式完成
- 错误处理---错误很难产生,并容易修正
- 一致性---我不需要学习新的技巧
- 反馈信息---我知道系统在做什么
- 功能性---系统能作我期望的工作
- 控制---系统交互在我的控制中
- 视觉清晰---如果有疑问,它就不应该出现
- 语言---我能了解我所读到的

4、兼容性测试

- 操作系统
- 浏览器
- 网络环境
- 分辨率
- 打印机

兼容性用例设计：操作系统

- 测试方法：根据需求中关于所支持的操作系统进行测试；如果没有，可以考虑一下按各种操作系统的市场占有率的多少来选择常用的操作系统进行测试

用例设计思想：

- 不同平台下web页面版式是否显示正常
- 相关控件或者脚本是否能正常的安装执行
- 网页程序是否能正常使用等

- 常用操作系统：Windows2000、XP、2003、Vista、Linux、Unix等，还应考虑各种不同语言版本的差别

兼容性测试用例设计：浏览器

用例设计思想：

- 网页脚本是否可以正常执行
- ActiveX控件是否正常运行
- HTML页面是否正常的显示
- 媒体文件是否可以直接播放

- 常见浏览器：IE6、IE7、Firefox、傲游和世界之窗


小贴士：浏览器的按钮（后退、刷新按钮）对功能有何影响

兼容性用例设计：网络环境

用例设计思想：

- 各种网络环境对页面显示、业务逻辑、数据存储、对话视频等的影响
- 网络速度对web性能的影响
- 防火墙打开和关闭
- 杀毒软件禁用
- 防木马软件

- 网速：56k、128k、1M、10M、100M等

兼容性用例设计：分辨率

用例设计思想：

- 显示是否正常
- 字体是否太小或者太大
- 文本和图片是否对齐
- 在窗口模式下拉伸或缩放是否有影响
- 是否影响图片的质量和像素多少

- 普通屏：640×480、800×600、1024×768、1280×1024、1600×1200等
- 宽屏：1280×720、1440×900、1680×1050等

兼容性用例设计：打印机

用例设计思想：

- 文字、表格、图片等是否打印正常
- 没有安装打印机时是否正确
- 专用打印机的效果、文本位置
- 忽略背景的打印是否正常
- 不同操作系统、不同分辨率下打印

组合测试

- 600x800 的分辨率在MAC 机上可能不错，但是在IBM 兼容机上却很难看。
- 在IBM 机器上使用Netscape 能正常显示，但却无法使用Lynx 来浏览。
- 有些内部应用程序，开发部门可能在系统需求中声明不支持某些系统而只支持一些那些已设置的系统。
- 理想的情况，系统能在所有机器上运行

浏览器与平台组合测试

- 测试矩阵

	WIN07	WIN2003	WIN2000	WIN ME	WIN XP	MAC9
IE6						
IE7						
IE8						
Firefox						
MSN Explorer8						
Netscape 7						

5、Web安全性测试

- 美国东部时间1999年6月29日7点36分，电脑黑客使用了一种为人熟知的软件攻击了美国陆军的主要站点，而且在长达9个小时的时间里面没有人发现。
- 2000年2月7日到9日这三天中，包括雅虎公司、亚马逊书店、有线电视新闻网等美国各大公司的网站都受到来历不明的电子攻击；之后，微软的3家网站也受到类似的攻击……

Web安全测试

技术概述

- 就攻击技术本质而言，它利用的工具是SQL的语法，针对的是应用程序开发者编程中的漏洞，当攻击者能操作数据，向应用程序中插入一些SQL语句时，SQL Injection攻击就发生了。
- 实际上，SQL Injection攻击是存在于常见的多连接的应用程序中的一种漏洞，攻击者通过在应用程序预先定义好的SQL语句结尾加上额外的SQL语句元素，欺骗数据库服务器执行非授权的任意查询，篡改和命令执行。

安全风险

- 就风险而言，SQL Injection攻击也是位居前列，和缓冲区溢出漏洞相比，其优势在于能够轻易的绕过防火墙直接访问数据库，甚至能够获得数据库所在的服务器的系统权限。
- 在Web应用漏洞中，SQL Injection 漏洞的风险要高过其他所有的漏洞。

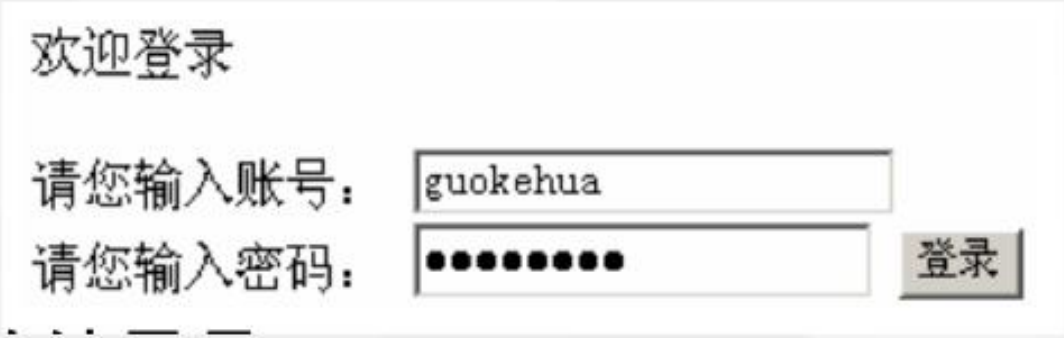
Web安全测试

- 终止式SQL注入：

攻击者注入一段包含注释符的SQL语句，将原来的语句的一部分注释，注释掉的部分语句不会被执行。

SQL注入

有一个登录页面，输入用户的账号、密码，查询数据库，进行登录，为了将问题简化，我们仅仅将其SQL打印出来供大家分析。在文本框内输入查询信息，提交，能够到达loginResult.jsp显示登录结果。假设此时用户名密码为：guokehua,guokehua



数据库执行的sql语句如下：

SELECT * FROM USERS WHERE USERNAME='guokehua' AND PASSWDORD='guokehua'

按sql执行的逻辑，这样执行，如果存在结果集，表示登录成功，如果不存在结果集，登录不成功。

SQL注入

上面那个登录功能，如果用户名密码在接收后，没有做任何处理，就进入sql语句，那会产生什么危害？

如果此时用户名输入aa‘ OR 1=1 --，密码为aa

该程序中，SQL语句为：

SELECT * FROM USERS WHERE USERNAME='aa' OR 1=1 --' AND PASSWORD='aa'

其中，--表示注释，因此，真正运行的SQL语句是：

SELECT * FROM USERS WHERE USERNAME='aa' OR 1=1

此处，‘1=1’永真，所以该语句将返回USERS表中的所有记录,此时网站受到了SQL注入的攻击。

SQL注入

- 注释符应用：

url为：http://host/newid=1,2,3

原sql语句

SELECT * FROM news WHERE id IN (1,2,3)

标红处存在sql注入

把url改为：http://host/newid=1) and 1=1- -

sql语句为:

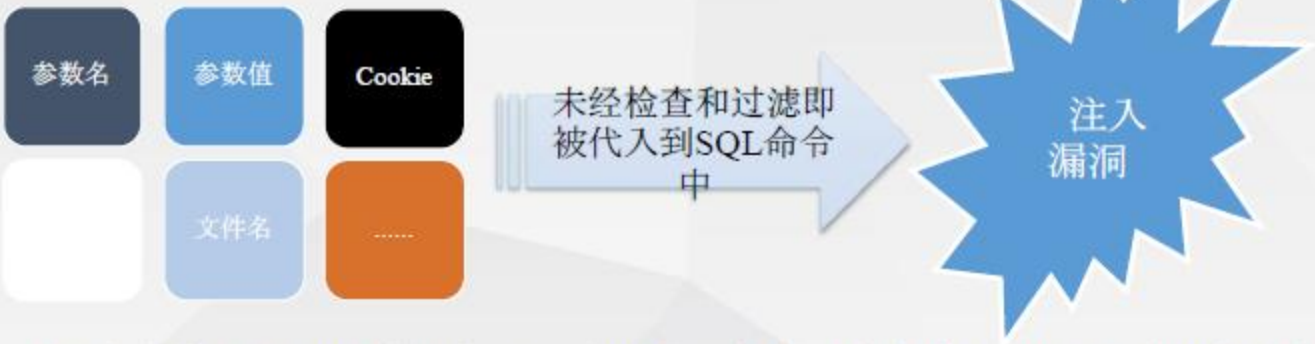
SELECT * FROM news WHERE id IN (1) and 1=1- -)

根据注入的语法可以分为

- Boolean-based blind SQL injection(布尔型注入)
  Select * from table where id = 1 and 1=1
- Error-based SQL injection(报错型注入)
  Select * from table where id = 1 or updatexml(1,concat(0x21,database()),1)
- Union query SQL injection(可联合查询注入)
  Select name from user where id = -1 union select user()
- Stack queries SQL injection(可多语句查询注入)
  SELECT * FROM user WHERE id=1 ; DELETE FROM user
- Time-base blind SQL injection(基于时间的注入)
  Select * from table where id = 1 and sleep(3)
  UPDATE `user` SET name = 'test' WHERE id = 1 and SLEEP(3)

SQL注入

- 哪些地方可能存在注入漏洞？



- 所有的输入只要和数据库进行交互的，都有可能触发SQL注入常见的包括：
  1.Get参数触发SQL注入
  2.POST参数触发SQL注入
  3.Cookie触发SQL注入
  4.其他参与sql执行的输入都有可能进行SQL注入
- 最普遍的注入漏洞是由于参数值过滤不严导致的。
- Cookie注入漏洞普遍存在于ASP的程序中。
- 参数名、目录名、文件名等注入漏洞通常存在于有网站路由的程序中。

安全性测试用例设计

- **Web应用系统的安全性测试区域主要有：**
  - 现在的Web应用系统基本采用先注册，后登陆的方式。因此，必须测试有效和无效的用户名和密码，要注意到是否大小写敏感，可以试多少次的限制，是否可以不登陆而直接浏览某个页面等。
  - Web应用系统是否有超时的限制，也就是说，用户登陆后在一定时间内（例如15分钟）没有点击任何页面，是否需要重新登陆才能正常使用。
  - 为了保证Web应用系统的安全性，日志文件是至关重要的。需要测试相关信息是否写进了日志文件、是否可追踪。
  - 当使用了安全套接字时，还要测试加密是否正确，检查信息的完整性。
  - 服务器端的脚本常常构成安全漏洞，这些漏洞又常常被黑客利用。所以，还要测试没有经过授权，就不能在服务器端放置和编辑脚本的问题。 (跨站式脚本)

安全性用例设计：登录

用例设计思想

- 正常和异常的用户名密码登录
- SQL注入式攻击（如：mm‘ or ’2‘>’1 ）
- 猜解密码的测试
- 不同权限用户登录

小贴士：安全性测试并不能最终证明应用程序是安全的。

http://www.cnblogs.com/coderzh/category/151315.html

强口令规则

Web应用安全开发规范中的强口令策略：

规则.1：口令长度的取值范围为：0-32 个字符；口令的最短长度和最长长度可配置；口令的最短长度建议默认为6个字符。

规则.2：口令中至少需要包括一个大写字母（A-Z）、一个小写字母（a-z）、一个数字字符（0-9）；口令是否包含特殊字符要求可以配置。

规则.3：口令中允许同一字符连续出现的最大次数可配置，取值范围：0-9，当取值为 0 时，表示无限制，建议默认为 3。

规则.4：口令须设置有效期，最短有效期的取值范围：0-9999 分钟，当取值为0时，表示不做限制，建议默认：5 分钟；最长有效期的取值范围：0-999 天，当取值为 0 时，表示口令永久有效，建议默认：90 天。

规则.5：在口令到期前，当用户登录时系统须进行提示，提前提示的天数可配置，取值范围：1-99 天，建议默认：7 天。

强口令规则

规则.6：口令到达最长有效期后，用户再次登录成功但在进入系统前，系统强制更改口令，直至更改成功。

规则.7：口令历史记录数可配置，取值范围为：0-30；建议默认：3个。

规则.8：管理员/操作员/最终用户修改自己的口令时，必须提供旧口令。

规则.9：初始口令为系统提供的默认口令、或者是由管理员设定时，则在用户/操作员使用初始口令成功登录后，要强制用户/操作员更改初始口令，直至更改成功。

规则.10：口令不能以明文的形式在界面上显示。

规则.11：口令不能以明文的形式保存，须加密保存；口令与用户名关联加密，即加密前的数据不仅包括口令，还包括用户名。

规则.12：只有当用户通过认证之后才可以修改口令。

规则.13：修改口令的帐号只能从服务器端的会话信息中获取，而不能由客户端指定。

强口令测试

编号	SEC_Web_ AUTHEN_08
用例名称	强口令策略测试
测试目的	本测试为检查目标系统是否存在强口令策略。
用例级别	2
测试条件	1. 已知Web网站地址 2. Web业务运行正常 3. Web业务存在帐号管理 4. 已知正常用户的用户、口令 5. 存在口令修改页面
执行步骤	1. 使用正确的用户、口令登陆Web业务系统 2. 打开口令修改页面 3. 在新口令输入框中输入字母加数字的5位字符（如ab123）作为密码并提交，如果未提示“口令长度过短”等诸如此类的信息，说明存在弱点，完成测试。 4. 在新口令输入框中输入6位数字（如123456）作为密码并提交，如果未提示 “口令字符需要大小写”等诸如此类的信息，说明存在弱点，完成测试。 5. 观察结果
预期结果	目标系统存在满足上述步骤的较严格的口令复杂度策略。
备注	上面只是举例说明口令复杂度的测试，实际上强口令策略还包括口令有效期、历史口令等，这些都要测试。对于一些Web应用（如移动网上客服系统）密码只能是数字组成，则不强制要求强口令。
测试结果	

课程回顾

- 从功能、性能、可用性、客户端兼容性、安全性等方面讨论了基于Web的系统测试方法。
- 基于Web的系统测试与传统的软件测试既有相同之处，也有不同的地方，对软件测试提出了新的挑战。
- 基于Web的系统测试不但需要检查和验证是否按照设计的要求运行，而且还要评价系统在不同用户的浏览器端的显示是否合适。
- 重要的是，还要从最终用户的角度进行安全性和可用性测试

Q & A