**WEB测试方法**

在Web工程过程中，基于Web系统的测试、确认和验收是一项重要而富有挑战性的工作。基于Web的系统测试与传统的不同，它不但需要检查和验证是否按照设计的要求运行，而且还要测试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。然而，Internet和Web媒体的不可预见性使测试基于Web的系统变得困难。因此，我们必须为测试和评估复杂的基于Web的系统研究新的方法和技术。  
本文将 web 测试分为 6 个部分：  
?  
?  
?  
（包括负载/压力测试）?  
?  
用户界面测试?  
?  
兼容性测试?  
?  
?  
?  
接口测试  
**1  
功能测试**  
 **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。链接测试可分为三个方面。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。?  
 **表单测试**  
当用户通过表单提交信息的时候，都希望表单能正常工作。  
如果使用表单来进行在线注册，要确保提交按钮能正常工作，当注册完成后应返回注册成功的消息。如果使用表单收集配送信息，应确保程序能够正确处理这些数据，最后能让顾客收到包裹。要测试这些程序，需要验证服务器能正确保存这些数据，而且后台运行的程序能正确解释和使用这些信息。  
当用户使用表单进行用户注册、登陆、信息提交等操作时，我们必须测试提交操作的完整性，以校验提交给服务器的信息的正确性。例如：用户填写的出生日期与职业是否恰当，填写的所属省份与所在城市是否匹配等。如果使用了默认值，还要检验默认值的正确性。如果表单只能接受指定的某些值，则也要进行测试。例如：只能接受某些字符，测试时可以跳过这些字符，看系统是否会报错。  
 **数据校验**  
如果系根据业务规则需要对用户输入进行校验，需要保证这些校验功能正常工作。例如，省份的字段可以用一个有效列表进行校验。在这种情况下，需要验证列表完整而且程序正确调用了该列表(例如在列表中添加一个测试值，确定系统能够接受这个测试值)。  
在测试表单时，该项测试和表单测试可能会有一些重复。  
 **cookies测试**  
Cookies通常用来存储用户信息和用户在某应用系统的操作，当一个用户使用Cookies访问了某一个应用系统时，Web服务器将发送关于用户的信息，把该信息以Cookies的形式存储在客户端计算机上，这可用来创建动态和自定义页面或者存储登陆等信息。?  
　如果Web应用系统使用了Cookies，就必须检查Cookies是否能正常工作。测试的内容可包括Cookies是否起作用，是否按预定的时间进行保存，刷新对Cookies有什么影响等。如果在 cookies 中保存了注册信息，请确认该 cookie能够正常工作而且已对这些信息已经加密。如果使用 cookie 来统计次数，需要验证次数累计正确。  
 **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，而输出错误主要是由于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。  
 **应用程序特定的功能需求**  
最重要的是，测试人员需要对应用程序特定的功能需求进行验证。尝试用户可能进行的所有操作：新增、修改、删除、查询等等。这是用户之所以使用网站的原因，一定要确认网站能像广告宣传的那样神奇。  
**2  
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。?  
 **负载压力测试?**  
在这里的负载\压力和功能测试中的不同,他是系统测试的内容,是基本功能已经通过后进行的.可以在集成测试阶段,亦可以在系统测试阶段进行。  
  
使用工具进行,虚拟一定数量的用户看一看系统的表现,是否满足定义中的指标。负载测试一般使用工具完成，loadrunner，，was，ewl，e-test等，主要的内容都是编写出，脚本中一般包括用户一般常用的功能，然后运行，得出报告。负载在各种极限情况下对产品进行测试 (如很多人同时使用该软件，或者反复运行该软件)，以检查产品的长期稳定性。例如，使用压力对web服务器进行压力测试. 本项测试可以帮助找到一些大型的问题，如死机、崩损、内存泄漏等，因为有些存在内存泄漏问题的程序，在运行一两次时可能不会出现问题，但是如果运行了成千上万次，内存泄漏得越来越多，就会导致系统崩滑。  
3  
**用户界面测试**  
界面是软件与用户交互的最直接的层，界面的好坏决定用户对软件的第一印象。而且设计良好的界面能够引导用户自己完成相应的操作，起到向导的作用。同时界面如同人的面孔，具有吸引用户的直接优势。设计合理的界面能给用户带来轻松愉悦的感受和成功的感觉，相反由于界面设计的失败，让用户有挫败感，再实用强大的功能都可能在用户的畏惧与放弃中付诸东流。目前界面的设计引起软件设计人员的重视的程度还远远不够，直到最近网页制作的兴起，才受到专家的青睐。而且设计良好的界面由于需要具有艺术美的天赋而遭拒绝。  
  
**窗口：**  
  
窗口是否基于相关的输入和菜单命令适当地打开？?  
  
窗口能否改变大小、移动和滚动？  
窗口中的数据内容能否用鼠标、功能键、方向键和键盘访问？?  
  
当被覆盖并重新调用后，窗口能否正确地再生？?  
需要时能否使用所有窗口相关的功能？?  
所有窗口相关的功能是可操作的吗？?  
是否有相关的下拉式菜单、工具条、滚动条、对话框、按钮、图标和其他控制可为窗口使用，并适当地显示？?  
显示多个窗口时，窗口的名称是否被适当地表示？?  
活动窗口是否被适当地加亮？?  
如果使用多任务，是否所有的窗口被实时更新？?  
多次或不正确按鼠标是否会导致无法预料的副作用？?  
窗口的声音和颜色提示和窗口的操作顺序是否符合需求？?  
窗口是否正确地被关闭？?  
  
  
**4  
  
兼容性测试**  
 **平台测试?**  
市场上有很多不同的操作系统类型，最常见的有、Unix、Macintosh、等。Web应用系统的最终用户究竟使用哪一种操作系统，取决于用户系统的配置。这样，就可能会发生兼容性问题，同一个应用可能在某些操作系统下能正常运行，但在另外的操作系统下可能会运行失败。?  
　　因此，在Web系统发布之前，需要在各种操作系统下对Web系统进行兼容性测试。?  
 **浏览器测试?**  
浏览器是Web客户端最核心的构件，来自不同厂商的浏览器对，、JavaScript、 ActiveX、 plug-ins或不同的HTML规格有不同的支持。例如，ActiveX是Microsoft的产品，是为Internet Explorer而设计的，JavaScript是Netscape的产品，Java是Sun的产品等等。另外，框架和层次结构风格在不同的浏览器中也有不同的显示，甚至根本不显示。不同的浏览器对安全性和Java的设置也不一样。?  
　测试浏览器兼容性的一个方法是创建一个兼容性矩阵。在这个矩阵中，测试不同厂商、不同版本的浏览器对某些构件和设置的适应性。?  
 **分辨率测试**  
页面版式在 640x400、600x800 或 1024x768 的分辨率模式下是否显示正常? 字体是否太小以至于无法浏览? 或者是太大? 文本和图片是否对齐?  
**5  
  
安全测试**  
主要是测试系统在没有授权的情况下，内部或者外部用户对系统进行攻击或者恶意破坏时如何进行处理，是否仍能保证数据的安全。测试人员可以学习一些黑客技术，来对系统进行攻击。  
 **登录**  
有些站点需要用户进行登录，以验证他们的身份。这样对用户是方便的，他们不需要每次都输入个人资料。你需要验证系统阻止非法的用户名/口令登录，而能够通过有效登录。用户登录是否有次数限制? 是否限制从某些 IP 地址登录? 如果允许登录失败的次数为3，你在第三次登录的时候输入正确的用户名和口令，能通过验证吗? 口令选择有规则限制吗?  
是否可以不登陆而直接浏览某个页面？  
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。  
**6  
接口测试**  
数据一般通过接口输入和输出，所以接口测试是的第一步。每个接口可能有多个输入参数，每个参数有“典型值”、“边界值”、“异常值”之分，所以输入的组合数可能并不少。根据接口的定义，可以推断某种输入应当产生什么样的输出。输出包括函数的返回值和输出参数。如果实际输出与期望的输出不一致，那么说明程序有错误。  
 **服务器接口**  
第一个需要测试的接口是浏览器与服务器的接口。测试人员提交事务，然后查看服务器记录，并验证在浏览器上看到的正好是服务器上发生的。测试人员还可以查询数据库，确认事务数据已正确保存。  
 **外部接口**  
有些 web 系统有外部接口。例如，网上商店可能要实时验证信用卡数据以减少欺诈行为的发生。测试的时候，要使用 web 接口发送一些事务数据，分别对有效信用卡、无效信用卡和被盗信用卡进行验证。  
 **错误处理**  
最容易被测试人员忽略的地方是接口错误处理。通常我们试图确认系统能够处理所有错误，但却无法预期系统所有可能的错误。尝试在处理过程中中断事务，看看会发生什么情况？订单是否完成？尝试中断用户到服务器的网络连接。尝试中断 web 服务器到信用卡验证服务器的连接。在这些情况下，系统能否正确处理这些错误？是否已对信用卡进行收费？如果用户自己中断事务处理，在订单已保存而用户没有返回网站确认的时候，需要由客户代表致电用户进行订单确认。  
**7  
测试点**  
 **文本框的测试**  
测试方法：  
　a,输入正常的字母或数字。  
　b,输入已存在的文件的名称；  
　c,输入超长字符。例如在“名称”框中输入超过允许边界个数的字符，假设最多255个字符，尝试输入 256个字符，检查程序能否正确处理；  
　d,输入默认值，空白，空格；  
　e,若只允许输入字母，尝试输入数字；反之；尝试输入字母；  
　f,利用复制，粘贴等操作强制输入程序不允许的输入数据；  
　g,输入特殊字符集，例如，NUL及\n等；  
　h,输入超过文本框长度的字符或文本，检查所输入的内容是否正常显示；  
　i,输入不符合格式的数据，检查程序是否正常校验，如，程序要求输入年月日格式为yy/mm/dd，实际输入yyyy/mm/dd，程序应该给出错误提示  
 **命令按钮测试**

测试方法：

　a,点击按钮正确响应操作。如，单击确定，正确执行操作；单击取消，退出窗口；  
　b,对非法的输入或操作给出足够的提示说明，如，输入月工作天数为32时，单击“确定”后系统应提示:天数不能大于31；  
　c,对可能造成数据无法恢复的操作必须给出确认信息，给用户放弃选择的机会；

**单选按钮的测试**

测试方法：

　a,一组单选按钮不能同时选中，只能选中一个。  
　  
b,逐一执行每个单选按钮的功能。分别选择了“男”“女”后，保存到数据库的数据应该相应的分别为“男”“女”；  
　?  
c,一组执行同一功能的单选按钮在初始状态时必须有一个被默认选中，不能同时为空；

**组合列表框的测试**

测试方法：

　a,条目内容正确，其详细条目内容可以根据需求说明确定；  
　  
b,逐一执行列表框中每个条目的功能；  
　  
c,检查能否向组合列表框输入数据；

**复选框的测试**

测试方法：

　a,多个复选框可以被同时选中；  
　b,多个复选框可以被部分选中；  
　c,多个复选框可以都不被选中；  
　d,逐一执行每个复选框的功能；

**列表框控件的测试**

测试方法：

　a,条目内容正确；同组合列表框类似，根据需求说明书确定列表的各项内容正确，没有丢失或错误；  
　b,列表框的内容较多时要使用滚动条；  
　c,列表框允许多选时，要分别检查shift选中条目，按ctrl选中条目和直接用鼠标选中多项条目的情况；

**滚动条控件的测试**

要注意一下几点：

　a,滚动条的长度根据显示信息的长度或宽度及时变换，这样有利于用户了解显示信息的位置和百分比，如，word中浏览100页文档，浏览到50页时，滚动条位置应处于中间；  
　b,拖动滚动条，检查屏幕刷新情况，并查看是否有乱码；  
　c,单击滚动条；  
　d,用滚轮控制滚动条；  
　e,滚动条的上下按钮。

**各种控件在窗体中混和使用时的测试**

　a,控件间的相互作用；  
　b,tab键的顺序，一般是从上到下，从左到右；  
　c,热键的使用，逐一测试；  
　d,enter键和esc键的使用；

在测试中，应遵循由简入繁的原则，先进行单个控件功能的测试，确保实现无误后，再进行多个控件的功能组合的测试。