**一、功能测试**

**1、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。

链接测试可以自动进行，现在已经有许多工具可以采用。链接测试必须在集成测试阶段完成，也就是说，在整个Web应用系统的所有页面开发完成之后进行链接测试。

**2、表单测试**

当用户给Web应用系统管理员提交信息时，就需要使用表单操作，例如用户注册、登陆、信息提交等。在这种情况下，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。

**3、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。

**4、设计语言测试**

Web设计语言版本的差异可以引起客户端或服务器端严重的问题，例如使用哪种版本的HTML等。当在分布式环境中开发时，开发人员都不在一起，这个问题就显得尤为重要。除了HTML的版本问题外，不同的脚本语言，例如Java、JavaScript、 ActiveX、VBScript或Perl等也要进行验证。

**5、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，而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。

**二、性能测试**

**1、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。

**2、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？

**3、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。

**三、可用性测试**

**1、导航测试**

导航描述了用户在一个页面内操作的方式，在不同的用户接口控制之间，例如按钮、对话框、列表和窗口等；或在不同的连接页面之间。通过考虑下列问题，可以决定一个Web应用系统是否易于导航：导航是否直观？Web系统的主要部分是否可通过主页存取？Web系统是否需要站点地图、搜索引擎或其他的导航帮助？

在一个页面上放太多的信息往往起到与预期相反的效果。Web应用系统的用户趋向于目的驱动，很快地扫描一个Web应用系统，看是否有满足自己需要的信息，如果没有，就会很快地离开。很少有用户愿意花时间去熟悉Web应用系统的结构，因此，Web应用系统导航帮助要尽可能地准确。

导航的另一个重要方面是Web应用系统的页面结构、导航、菜单、连接的风格是否一致。确保用户凭直觉就知道Web应用系统里面是否还有内容，内容在什么地方。

Web应用系统的层次一旦决定，就要着手测试用户导航功能，让最终用户参与这种测试，效果将更加明显。

**2、图形测试**

在Web应用系统中，适当的图片和动画既能起到广告宣传的作用，又能起到美化页面的功能。一个Web应用系统的图形可以包括图片、动画、边框、颜色、字体、背景、按钮等。图形测试的内容有：

（1）要确保图形有明确的用途，图片或动画不要胡乱地堆在一起，以免浪费传输时间。Web应用系统的图片尺寸要尽量地小，并且要能清楚地说明某件事情，一般都链接到某个具体的页面。

（2）验证所有页面字体的风格是否一致。

（3）背景颜色应该与字体颜色和前景颜色相搭配。

（4）图片的大小和质量也是一个很重要的因素，一般采用JPG或GIF压缩。

**3、内容测试**

内容测试用来检验Web应用系统提供信息的正确性、准确性和相关性。

信息的正确性是指信息是可靠的还是误传的。例如，在商品价格列表中，错误的价格可能引起财政问题甚至导致法律纠纷；信息的准确性是指是否有语法或拼写错误。这种测试通常使用一些文字处理软件来进行，例如使用Microsoft Word的”拼音与语法检查”功能；信息的相关性是指是否在当前页面可以找到与当前浏览信息相关的信息列表或入口，也就是一般Web站点中的所谓”相关文章列表”。

**4、整体界面测试**

整体界面是指整个Web应用系统的页面结构设计，是给用户的一个整体感。例如：当用户浏览Web应用系统时是否感到舒适，是否凭直觉就知道要找的信息在什么地方？整个Web应用系统的设计风格是否一致？

对整体界面的测试过程，其实是一个对最终用户进行调查的过程。一般Web应用系统采取在主页上做一个调查问卷的形式，来得到最终用户的反馈信息。

对所有的可用性测试来说，都需要有外部人员（与Web应用系统开发没有联系或联系很少的人员）的参与，最好是最终用户的参与。

**四、客户端兼容性测试**

**1、平台测试**

市场上有很多不同的操作系统类型，最常见的有Windows、Unix、Macintosh、Linux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。

**2、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Java，、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。

**五、安全性测试**

Web应用系统的安全性测试区域主要有：

（1）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。

（2）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。

（3）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。

（4）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。

（5）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。