

# Web 应用测试

## 1, Web 应用测试特性:

- 首先, 为什么要进行 web 应用测试?

因为在开发过程中会出现各种 bug 或者设计方面的不合理因素。

- 包括如下过程中可能发生的几点:

HTML 页面之间的交互(断开链接或孤立链接)

运行在服务器端的应用程序(例如 CGI 脚本、业务规则、与其他应用的集成、数据库连接、日志应用程序、动态页面生成器、asp、jsp)

各种服务器和浏览器(异构系统)

不一致性软件与非收敛技术

信息安全(脆弱性)

网络问题(延迟、协议等)

遗留系统(数据不一致、缺少事务)

性能(建设能力不足)

维护与保障

可用性(很难找到有用的信息或说明)

可伸缩性(而不是随着体积的增加而扩展)

- 那么我们面对这些问题该如何解决:

正确并且大量的测试, 不停地犯错并且修改。

## 2, Web 应用测试过程:

测试过程流程一般如下:

功能测试、内容测试和评审、Web 页面测试、导航测试、接口测试、配置测试、安全测试、性能测试。

这里只选几个测试模块进行重点分析, 基于篇幅限制, 不用将全部内容进行叙述。

- 本次测试由于项目只是校园项目的原因, 所以测试人群选的范围比较小, 而且我们的网站做的是体育直播网站, 所以面向群体还是年轻人。

- 首先进行测试的当然是体育爱好者, 由于体育爱好者有着丰富的使用直播网站看比赛的经验, 所以可以更加容易看出来不足以及优点, 其次就是普通的直播观看者, 这类的用户不会太过苛刻, 对他们来说, 舒适度是最重要的, 只要达到直播网站的平均水平就可以, 最后是很少使用直播网站的用户, 这类用户一般不会对直播网站上花费太多时间, 而是利用碎片化时间进行观看, 那么对这类用户来说, 重要的就是是否简洁方便。

将用户以使用时间进行优先级划分, 这样比较容易全面的测试。

- 不同的用户使用不同的测试方案, 重度用户就要用长时间的比赛进行测试, 以观察用户的反馈, 而轻度用户则可以给他观看短视频。



### 3，功能测试：

- 链接测试：可以进行测试链接是否可以访问，使用不同的设备以及不同的网络
- 交互测试：交互测试则是根据交互设计，查看注册，登陆，搜索，弹幕等操作是否有什么错误，以及交互过程是否流畅，没有卡顿。

### 4，内容测试：

- 语法错误：这个阶段就是查看网页中是否有一些显而易见的拼写错误等小问题
- 语义错误：语义错误则是要查看网页中的内容是否有着明显的逻辑错误，导致用户理解有误，要记住尽量简单易懂。
- 组织/结构错误：这个阶段就是检查网页设计层面中布局以及配色字体是否美观以及和谐，需要与整体环境一致。还有各个界面的布局是否符合用户的习惯，应该尽量做到和其他网站功能大致类似，但要有自己特色。

### 5，Web 页面测试：

- 字体：样式一致性字体清晰度，识别斜体和截线字体的困难度，一篇文档中多种字体造成的视觉混淆以及目标平台上字体的可用性
- 颜色：背景颜色的适宜性，前景颜色的适宜性，字体颜色的适宜性。精细、互补颜色的选择通常比饱和、反差色更悦目
- 边缘：命令按钮的三维效果对用户是有效的可视化提示，对非交互元素三维效果的使用会被混淆
- 图像：大图像可能增加装载时间，可视化提示和设计细节同背景是否能区分背景的适宜性，标签清晰度，按钮清晰度，图片尺寸的适宜性
- 框架：浏览器能否正确显示，显示设置和浏览器类型影响框架的显示情况，后退按钮经常有意想不到的结果
- 表格：网格（表格中的表格）减慢 HTML 的装载速度，外观是否由于显示设置和浏览器类型而导致不正确的范围和重叠，测试应该包括所有浏览器、显示设置和浏览器窗口大小

### 6，性能测试：

#### • 测试评估阶段

在这个阶段，我们要评估被测的产品是否要进行性能测试，并且对目前的服务器环境进行粗估，服务的性能是否满足条件。

首先要明确只要涉及到准备上线的服务端产品，就需要进行性能测试。其次如果产品需求中明确提到了性能指标，那也必须要做性能测试。

测试人员在进行性能测试前，需要根据当前的收集到的各种信息，预先做性能的评估，收集的内容主要包括带宽、请求包大小、并发用户数和当前 web 服务

的带宽等。

- 测试准备阶段

在这个阶段，我们要了解以下内容：

- a. 服务器的架构是什么样的，例如：web 服务器是什么？是如何配置的？数据库用的是什麼？服务用的是什麼语言编写的？
  - b. 服务端功能的内部逻辑实现；
  - c. 服务端与数据库是如何交互的，例如：数据库的表结构是什么样的？服务端功能是怎样操作数据库的？
  - d. 服务端与客户端之间是如何进行交互的，即接口定义；
- 通过收集以上信息，测试人员整理出服务器端各模块之间的交互图，客户端与服务端之间的交互图以及服务端内部功能逻辑实现的流程图。
- e. 该服务上线后的用户量预估是多少，如果无法评估出用户量，那么可以通过设计测试执行的场景得出这个值；
  - f. 上线要部署到多少台机器上，每台机器的负载均衡是如何设计的，每台机器的配置什么样的，网络环境是什么样的。
  - g. 了解测试环境与线上环境的不同，例如网络环境、硬件配置等
  - h. 制定测试执行的策略，是需要验证需求中的指标能否达到，还是评估系统的最大处理能力。
  - i. 沟通上线的指标

通过收集以上信息，确定性能测试用例该如何设计，如何设计性能测试用例执行的场景，以及上线指标的评估。

- 测试设计阶段

根据测试人员通过之前整理的交互图和流程图，设计相应的性能测试用例。性能测试用例主要分为预期目标用户测试，用户并发测试，疲劳强度与大数量测试，网络性能测试，服务器性能测试，具体编写的测试用例要更具实际情况进行裁减。

用例编写的步骤大致分为：

- a. 通过脚本模拟单一用户是如何使用这个 web 服务的。这里模拟的可以是用户使用 web 服务的某一个动作或某几个动作，某一个功能或几个功能，也可以是使用 web 服务的整个过程。
- b. 根据客户端的实际情况和服务器端的策略，通过将脚本中可变的数据进行参数化，来模拟多个用户的操作。
- c. 验证参数化后脚本功能的正确性。
- d. 添加检查点
- e. 设计脚本执行的策略，如每个功能的执行次数，各个功能的执行顺序等

- 测试执行阶段

根据客户端的产品行为设计 web 服务的测试执行场景及测试执行的过程，即测试执行期间发生的事儿。通过监控程序收集 web 服务的性能数据和 web 服务所在系统的性能数据。

- 测试分析阶段

将收集到的数据制成图表，查看各指标的性能变化曲线，结合之前确定的上线指标，对各项数据进行分析，已确定是否继续对 web 服务进行测试，结果是否达到了期望值。

- 测试验证阶段

在开发针对发现的性能问题进行修复后，要再执行性能测试的用例对问题进行验证。这里需要关注的是开发在解决问题的同时可能无意中修改了某些功能，所以在验证性能的同时，也要关注原有功能是否受到了影响。