软件测试

## 简述TestLink测试管理的基本流程

1.用户设置

系统为TestLink创建一个默认的管理员账号，账号/密码为：admin/admin。

初次访问，需要建立一个新的测试项目。需要选择“增强功能”，以及活动，公开。

2.创建需求规格以及测试需求

1）创建需求规格

  主页——产品需求——新建需求规格。

2）创建需求

  需求规格——创建新产品需求——新建测试需求。

 测试需求包括：文档标识、名称、范围、需求的状态，以及覆盖需求的案例。

3.测试用例管理

1）创建用例集

   主页——测试用例——编辑测试用例——点击左侧文件夹——点击右侧“新建测试用例集”。

2）添加测试用例

  点击测试用例集——创建测试用例——新建用例——创建——创建步骤——输入数据——保存。

3）需求关联

   测试用例与系统/软件需求之间是n对n的关系。

  主页——产品需求——指派产品需求——点击左侧测试用例——点击右侧“测试需求”——指派。

4.制定测试计划（只能由admin用户进行）

1）创建测试计划

  主页——测试计划管理——测试计划管理——创建，应选择“活动，公共”。

  测试计划包括：计划名称、计划描述，及是否从已有的测试计划创建。

2）创建测试里程碑

  明确每个测试阶段的开始和截止时间，以及完成A,B,C三种优先级的比例。

  主页——测试计划管理——编辑/删除里程碑——创建——保存。

里程碑包括：名称、日期、优先级

3）构建管理

  测试计划做好后，应制定版本，如ver1.0。测试过程中发现了BUG，修改之后产生了2.0。此时应追加版本，相应的接下来未完成的测试以及降级测试都应在新的版本上完成。

   主页——测试计划管理——构建管理——创建，应选择“活动，打开”。

4）安排测试人员（定义用户/测试计划角色权限）

   主页——指派用户角色——更新。

5）设置测试用例的所用者（给测试人员分派测试任务）

    主页——测试用例集——指派执行测试用例——左侧选择测试用例集或测试用例——右侧选择被指派者——保存。

6）执行测试

  顶部菜单栏执行——测试用例执行。

6.测试结果分析

## 简述Mantis中Bug有那些状态？并对这些状态做简要说明

一、New(新建) 状态说明：

1. 测试人员发现bug，填写好bug report后通过“1”submit bug，此时bug的状态为new； 2. 处于new状态的bug，team leader认为是bug且需要修复的，通过“2”将bug分派给

相应的开发人员，此时bug的状态为assigned(分配状态)；

3. 处于new状态的bug，team leader认为测试人员的bug描述不清或者有疑问，通过“3”

将bug反馈给测试人员，并在note里注明原因，此时bug的状态为confirmed（回馈状态）；

4. 处于new状态的bug，team leader认为需要延后修复、需跟客户确认、不能修复的bug，

通过“4”将bug的状态更改为acknowledged（等待状态）。

二、Assigned(已分派) 状态说明：

5. 处于assigned状态的bug，开发人员将其修复后，通过“5”将bug的状态更改为resolved； 6. 处于assigned状态的bug，开发人员认为测试人员的bug描述不清或者有疑问，通过“6”

将bug反馈给测试人员，并在note里注明原因，此时bug的状态为confirmed； 7. 处于assigned状态的bug，开发人员认为需要延后修复、需跟客户确认、不能修复的bug，

通过“7”将bug的状态更改为acknowledged。

三、Resolved(已解决) 状态说明：

8. 处于resolved状态的bug，测试人员进行回归测试确认bug已经被修复后，通过“8”

将bug关闭，此时bug的状态为closed；

9. 处于resolved状态的bug，测试人员进行回归测试发现bug没有修复或者由此引发了其

他的bug，通过“9”将bug反馈给开发人员，并在note里注明原因，此时bug的状态为feedback。

四、Closed(已关闭) 状态说明：

10. 处于closed状态的bug，若测试人员在测试过程中还发现该问题，可以通过“10” 将

bug反馈给开发人员，并在note里注明原因，此时bug的状态为feedback。

五、Feedback(打回) 状态说明：

11. 处于feedback状态的bug，开发人员将其修复后，通过“11”将bug的状态更改为resolved； 12. 处于feedback状态的bug，开发人员认为测试人员的bug描述不清或者有疑问，通过“12”

将bug反馈给测试人员，并在note里注明原因，此时bug的状态为confirmed； 13. 处于feedback状态的bug，team leader或者开发人员认为该bug需要重新分派，通过“13”

将bug重新分派给相应的开发人员，此时bug的状态为assigned；

14. 处于feedback状态的bug，开发人员认为需要延后修复、需要跟客户确认、不能修复的

bug，通过“14”将bug的状态更改为acknowledged。

六、Confirmed(已确认) 状态说明：

15. 处于confirmed状态的bug，测试人员对于开发人员的疑问作出回应，通过“15”将bug

的状态更改为feedback，并在note里注明原因；

16. 处于confirmed状态的bug，测试人员认为该bug有争议，需上一级领导确认的，通过

“16”将bug的状态更改为acknowledged；

17. 处于confirmed状态的bug，测试人员认为该bug不需要修复，通过“17”将bug关闭，

此时bug的状态为closed。

七、Acknowledged(公认) 状态说明：

18. 处于acknowledged状态的bug，team leader或者测试人员认为该bug必须修复，通过“18”

重新将bug分派给相应的开发人员，此时bug的状态为assigned；

19. 处于acknowledged状态的bug，team leader或者开发人员发现该bug已经修复了，通过

“19” 将bug的状态更改为resolved；

20. 处于acknowledged状态的bug，测试人员认为该bug不需要修复，通过“20”将bug

关闭，此时bug的状态为closed

## 简述你所知道的Android自动化测试工具

 Robotium 安卓测试工具

MonkeyRunner 安卓应用测试  
Ronaorex 安卓测试应用工具

Appium安卓自动化框架

UI Automator 安卓测试自动化

## 敏捷测试与普通测试有什么区别

1.项目相当于开发与测试并行，项目整体时间较快。

2.模块提交较快，测试时较有压迫感。

3.工作任务划分清晰，工作效率较高。

4.项目规划要合理，不然测试时会出现复测的现象，加大工作量。

　5.发现问题需跟紧，项目中人员都比较忙，问题很容易被遗忘。

　6.耗时、或较难解决对项目影响不大的问题一般会遗留到下个阶段解决。

　7.发现BUG能够很快的解决，对相关的模块的测试影响比较小。

　8.版本更换比较勤，影响到测试的速度。

　9.要多与开发沟通。

　10.要注意版本的更新情况。

　11.测试人员几乎要参加整个项目组的所有会议。

## 什么是性能测试、负载测试、压力测试

性能测试：

性能测试是通过自动化的测试工具模拟多种正常、峰值以及异常负载条件来对系统的各项性能指标进行测试。负载测试和压力测试都属于性能测试，两者可以结合进行。通过负载测试，确定在各种工作负载下系统的性能，目标是测试当负载逐渐增加时，系统各项性能指标的变化情况。压力测试是通过确定一个系统的瓶颈或者不能接受的性能点，来获得系统能提供的最大服务级别的测试。

负载测试：

负载测试（Load testing），通过测试系统在资源超负荷情况下的表现，以发现设计上的错误或验证系统的负载能力。在这种测试中，将使测试对象承担不同的工作量，以评测和评估测试对象在不同工作量条件下的性能行为，以及持续正常运行的能力。

负载测试的目标是确定并确保系统在超出最大预期工作量的情况下仍能正常运行。此外，负载测试还要评估性能特征。例如，响应时间、事务处理速率和其他与时间相关的方面。

压力测试：

压力测试（Stress Test），也称为强度测试、负载测试。压力测试是模拟实际应用的软硬件环境及用户使用过程的系统负荷，长时间或超大负荷地运行测试软件，来测试被测系统的性能、可靠性、稳定性等。

## 简要说明WEB测试应注意那些问题

web 测试大致可分为 6 个部分：  
用户界面测试 ：用户界面测试要注意是否有使用说明、站点地图和导航条，还要关注内容、颜色/背景、图片表格等。  
功能测试：功能测试要关注链接、信息交互、数据校验等。  
接口测试：接口测试关注服务器接口、外部接口、错误处理等。  
兼容性测试：兼容性测试要关注操作系统、浏览器、Modem/连接速率、硬件设备等的兼容性。  
负载/压力测试：要关注瞬间访问高峰、每个用户传送大量数据、长时间的使用等。  
安全测试：要关注目录设置、登录、日志文件等。

## 软件的本地化测试包括那些内容？软件的国际化测试包括那些内容

软件本地化测试：

* 功能性测试，所有基本功能、安装、升级等测试；
* 翻译测试，包括语言完整性、术语准确性等的检查；
* 可用性测试，包括用户界面、度量衡和时区等；
* 兼容性调试，包括硬件兼容性、版本兼容性等测试；
* 文化、宗教、喜好等适用性测试
* 手册验证，包括联机文件、在线帮助、PDF文件等测试

软件国际化测试：

软件国际化特征功能测试：

软件国际化特征功能测试描述如何测试国际化软件，以保证软件在任何区域性或区域设置中都能正常运行，并且特征功能都能正确地实现。

本地化能力测试：

本地化能力测试描述测试软件是否可以被方便的本地化，测试软件是否可以轻松地将程序的用户界面翻译成任何目标语言，而不需要重新设计或修改代码。

本地化测试：

本地化测试描述如何测试本地化的软件，检查特定目标区域性或区域设置的产品本地化的质量，包括本地化功能测试、语言质量测试和用户界面测试等。

## 跨浏览器测试有那些注意事项

最好同时用不同的浏览器打开一个测试站点，进行逐一比较测试。跨浏览器测试时，要注意同一个浏览器不同版本的变化。

## Web安全测试通常要考虑那些测试点

1、问题：没有被验证的输入

　　测试方法：

　　数据类型(字符串，整型，实数，等)

　　允许的字符集

　　最小和最大的长度

　　是否允许空输入

　　参数是否是必须的

　　重复是否允许

　　数值范围

　　特定的值(枚举型)

　　特定的模式(正则表达式)

　　2、问题：有问题的访问控制

　　测试方法：

　　主要用于需要验证用户身份以及权限的页面，复制该页面的url地址，关闭该页面以后，查看是否可以直接进入该复制好的地址

　　3、错误的认证和会话管理

　　4、缓冲区溢出

　　没有加密关键数据

　　在页面输入密码，页面显示的是 \*\*\*\*\*, 右键，查看源文件就可以看见刚才输入的密码，

　　5、拒绝服务

　　分析：领测认为攻击者可以从一个主机产生足够多的流量来耗尽狠多应用程序，最终使程序陷入瘫痪。需要做负载均衡来对付。

　　6、不安全的配置管理

　　分析：Config中的链接字符串以及用户信息，邮件，数据存储信息都需要加以保护

　　程序员应该作的： 配置所有的安全机制，关掉所有不使用的服务，设置角色权限帐号，使用日志和警报。

　　分析：用户使用缓冲区溢出来破坏web应用程序的栈，通过发送特别编写的代码到web程序中，攻击者可以让web应用程序来执行任意代码。

7、注入式漏洞

8，不恰当的异常处理

分析：程序在抛出异常的时候给出了比较详细的内部错误信息，暴露了不应该显示的执行细节，网站存在潜在漏洞，

9，不安全的存储

分析：帐号列表：系统不应该允许用户浏览到网站所有的帐号，如果必须要一个用户列表，推荐使用某种形式的假名（屏幕名）来指向实际的帐号。

浏览器缓存：认证和会话数据不应该作为GET的一部分来发送，应该使用POST，

10 问题：跨站脚本（XSS）

分析：攻击者使用跨站脚本来发送恶意代码给没有发觉的用户，窃取他机器上的任意资料

测试方法：

• HTML标签：<…>…</…>

• 转义字符：&(&)；<(<)；>(>)； (空格) ；

• 脚本语言：

<script language=‘javascript’>

…Alert(‘’)

</script>

• 特殊字符：‘ ’ < > /

• 最小和最大的长度

• 是否允许空输入

## 测试人员在软件开发过程中的任务是什么

1、寻找Bug；

2、避免软件开发过程中的缺陷；

3、衡量软件的品质；

4、关注用户的需求。

总的目标是：确保软件的质量。