# 测试用例规范

此规范定义了测试用例的属性、级别、撰写以及执行等规范。

## 用例级别属性（Keywords）

此字段主要是用来标识用例的级别，通过对用例的级别定义，可以使整个测试过程分级式管理，从而指导测试流程的顺序、突出重点问题、选择性测试，节约了测试成本，同时也便于更好的表述和分析测试结果。也是我们定义开始测试标准、停止测试标准以及回归测试标准的一个依据。所有的用例将被定义的状态如下：

**1、Base（基础，可测用例）**

此级别为基础型，表述了能保证系统可以运行的基本条件，同时也定义了此安装包可以进行下一步测试的标准。

*如：是否有遗漏功能、安装测试、基本功能点是否没有问题、各项服务是否都是正常运行、提交制品是否相符等等。*

**2、Important （重要，可测用例）**

此级别为重要型，在完全通过了Base类型用例的测试后，首先要测试的用例，这些用例表述了系统能正常运行的条件和此版本发布必需要满足的条件。此类型的用例将符合以下标准之一：

|  |
| --- |
| * 涉及整个功能模块的用例，如果此用例不通过，将导致功能模块不可使用或功能不正常的用例。如：注册用户、注册服务等等。 * 能表明某项功能基本满足需求的用例。 * 涉及到共性问题的用例，如： 发送消息等。 * 在release Note声明解决的bug。 * 用户在正常使用中，出现频繁情况的用例。 |

**3、Normal（普通用例）**

此级别为普通型，测试用例的主体。该类型的用例主要针对在保证系统正常运行的前提下，对功能行用例进行完善，对功能性用例进行扩展。另外还有测试系统对特殊情况、异常情况、异常操作情况的处理能力，从细微找出系统漏洞，从而更加完善系统的抗压性和稳定性。

**4、Extend（扩展用例）**

此级别为建议型，该类型用例针对不影响系统正常运行的建议，力图完善系统现有功能或提出对系统功能的展望。如：界面显示信息明确、系统退信的语言规范等

## 用例书写规范

**1、Test Case Title命名规则**

用例命名时用例应该能够描述出用例的测试目的。

**2、Summary详细信息**

Summary标签，概述了用例的基本情况，应包含以下信息：

|  |
| --- |
| * 测试目的。概述此用例的测试重点和容易出现问题的地方（必写项）。 * 前提条件。描述此用例能够执行的基本条件（必写项）。 * 测试数据。描述此用例使用的测试素材，包括：输入字符串、大小等数据。 * 备注。选择填写，可在此项描述用例的测试历史，曾经出现的bug等需要表述给测试人员的信息。 |

**3、Steps设计步骤**

此标签为执行用例的主要步骤标签，详细的描述了用例执行的方法步骤和应有的结果。对此标签的撰写做以下规范：

|  |
| --- |
| * 每一步操作都对应唯一的功能点。避免没有对应功能点的无用步骤，同时也避免一个步骤对应多个功能点的情况 * 每一操作步骤的Expected Result必须填写。且只能出现描述结果的语言，不能出现描述操作的语言 |

**4、Attachments附件**

此标签为已有的用户添加附件。通过附件可以更加清楚或方便的表述用例内容，如：通过Excel文件批量添加用户、上传本地数据库文件等。

## 测试用例的执行

**1、严格按照用例的操作步骤执行**

测试过程中，必须严格按照用例设计的步骤进行操作，每个步骤都要记录测试结果并标记结果状态，不可以自行增删步骤，不可私自跳过用例不执行。

**2、比较每步操作的结果**

测试过程中，每执行一步，就应该记录此步骤所产生的结果并与预期结果进行比较。

**3、无法执行的用例应该做出标记并尽量找到解决方法**

测试过程中，遇到无法执行的用例时，比如缺少测试数据、测试环境等，必须对该用例做出相应的标记并积极寻求解决的方法。

## 测试用例的执行的结果

|  |  |
| --- | --- |
| 状态 | 说明 |
| No Run | 用例没有执行（用例的最初状态） |
| Failed | 用例执行失败 |
| Passed | 用例执行成功 |
| Blocked | 由于环境原因，无法执行的用例 |

# Bug的提交和管理规范

## BUG的属性

|  |  |
| --- | --- |
| **Column** | **Description** |
| Version | Bug被发现时的软件版本号 |
| Component | 提交bug时，bug对应的模块选择列表 |
| Platform | 测试时使用的硬件平台，可根据bug的实际情况选择 |
| OS | 测试时使用的操作系统版本，可根据bug的实际情况选择 |
| Severity | Bug的严重级别，根据计算出的RPN值来确定bug严重度 |
| Priority | Bug解决的优先级别，P1至P5逐渐减弱，根据Bug的严重度来确定 |
| Initial State | 初始状态，测试人员此处为可选状态Unconfirmed和New |
| Assigned to | 确定bug指派的开发人员，这里统一要用被指派人员登陆bugzilla用的邮箱 |
| CC | 可抄送给多人，将该bug的状态通知给相关人员，可自行指定 |
| Estimated Hours | 解决此bug需要的时间长度 |
| Deadline | 为此bug预设的解决期限 |
| URL | Bug的定位（可选） |
| Re-producible | Bug的复现率，即bug的出现频率 |
| Summary | Bug概述，简要描述bug |
| Description | Bug的详述，包括测试环境，操作步骤，执行完后实际结果和预期结果 |
| Attachment | 如需要对bug做其它说明，可添加bug相关附件 |
| Depends on | 表示当前bug与哪个bug有依赖关系 |
| Blocks | 此bug影响了，阻碍了其他bug的修改进程 |

## BUG的状态属性

|  |  |
| --- | --- |
| **Status** | **Description** |
| New | 新建Bug |
| Assigned | 接受New的Bug，并分配给指定的开发人员 |
| Fixed | Bug已修复 |
| Closed | 已修复Bug的关闭 |
| Duplicate | Bug重复 |
| Wontfix | 描述的问题将不会被修复 |
| Invalid | 描述的问题不是一个bug (输入错误后，通过此项来取消) |
| Reopen | Bug重新开放 |
| Verified | Bug经验证无误后置为Verified |
| Worksforme | 无法重现该bug，如有更多信息出现，请重新分配，现将其归档 |
| Later | 描述的问题将不会在产品的这个版本中解决 |

## BUG的等级属性

Bug的严重度根据计算出来的RPN值来确定。

故障等级分类

|  |  |
| --- | --- |
| **Type** | **Description** |
| 严重度（S） | 指潜在的故障出现时对用户造成的影响程度，评估分值为1-5分。严重度影响越大，分值越高。 |
| 难检度（D） | 指用户在正常使用过程中对产品功能或者菜单使用的频繁度,评估分值为1-5分, 在正常使用中越容易被用户发现，分值越高。 |
| 频度（O） | 指该问题可能发生的频率.评估分值为1-5分，出现频率越高，分值越高。 |
| 故障风险指标（RPN） | 由故障严重度（S）、难检度（D）和频度（O）组成，故障严重等级分值为故障严重度、难检度和频度的乘积(RPN=S\*O\*D) 。 |

严重度（S）

|  |  |  |
| --- | --- | --- |
| 等级 | 描 述（举例） | 分 值 |
| A | •测试执行系统的主要功能时，会直接导致系统死机、反复重启、蓝屏、挂起或是程序非法退出； | 5 |
| •系统的主要功能点没有实现； |
| •主要模块或主要功能不满足用户需求或设计上的要求； |
| •系统存在一定的安全问题，会缺陷导致重要数据丢失或损坏。 |
| B | •测试执行系统的次要功能时，会导致系统死机、蓝屏、挂起或是程序非法退出； | 4 |
| •系统的次要功能点没有实现； |
| •对于主要功能的执行结果与预期结果差别较大，或是计算结果不正确； |
| •系统的易用性不好，导致用户可能不能正常完成系统的主要功能操作； |
| •系统执行过程过于缓慢； |
| •系统占用过大的系统资源，或是占用资源后不能正常释放； |
| •主要交互界面有明显的错别字或描述错误 |
| C1 | •系统实际执行过程与预期结果有差异，但不严重； | 3 |
| •非正常操作或输入会导致系统出错，或执行结果不正确； |
| •系统运行过程中偶尔（出现概率<5%）有出错提示或导致系统运行不正常； |
| C2 | •系统交互性不好，对于用户可能造成难于操作、学习和理解； | 2 |
| •在用户经常使用的环境中，交互界面不美观，影响系统品质； |
| •交互界面、程序或帮助文档中文档或文字描述问题，造成用户难于理解。 |
| D | •系统实际执行过程与预期结果有较小的差异； | 1 |
| •系统不能处理用户可能使用的极端条件下的操作； |
| •交互界面、程序或帮助文档中文档或文字描述存在问题，但影响不大。 |

难检度（D）

|  |  |  |
| --- | --- | --- |
| 故障发生的可能性 | 描 述 | 分 值 |
| 每次再现 | 100% | 5 |
| 经常再现 | 大于20% | 4 |
| 很少再现1 | 小于20%，大于10% | 3 |
| 很少再现2 | 小于10% | 2 |
| 出现一次 | 在整个测试工作（一轮）中只出现一次 | 1 |

频度（O）

|  |  |  |
| --- | --- | --- |
| 用户使用频率 | 描 述 | 分 值 |
| 特别经常使用 | 用户在正常使用中，特别经常使用的菜单及功能,使用频率81-100%。 | 5 |
| 经常使用 | 用户在正常使用中，经常使用的菜单及功能，使用频率51-80%。 | 4 |
| 不常用 | 用户在正常使用中，不常用的菜单及功能，使用频率31-50%。 | 3 |
| 很少用 | 用户在正常使用中，很少用的菜单及功能，使用频率10-30%。 | 2 |
| 几乎不用 | 用户在正常使用中，基本上不会使用的菜单及功能，使用频率10%以下。 | 1 |

BUG的等级属性

|  |  |
| --- | --- |
| **Level** | **Description** |
| Blocker(非常严重) | 阻碍了项目开发或测试的继续进行（RPN=125） |
| Critical（严重） | 冲突，数据丢失和严重的内存泄露问题（100=<RPN<=124） |
| Major（问题较大） | 严重的功能缺陷（64<=RPN<=99） |
| Normal（一般问题） | 一般的功能缺陷（32<=RPN<=63） |
| Minor（问题较小） | 较小的功能缺陷（16<=RPN<=31） |
| Trivial（微小） | 拼写，对齐类的错误（8<=RPN<=15） |
| Enhancement（有待提高） | 需要改进的（1<=RPN<=7） |

## BUG解决优先级别的属性

|  |  |
| --- | --- |
| **Level** | **Description** |
| P1 | 严重问题，必须马上解决，例如系统安装包问题，相关重要功能未实现 |
| P2 | 重大问题，尽快解决，例如导致整个模块不可用的问题 |
| P3 | 一般问题，不影响系统基本功能，但影响用户正常使用，如果时间允许应该修改 |
| P4 | 微小的，轻微的问题 |
| P5 | 建议，可以不解决或以后解决 |

## BUG解决状态定义

|  |  |
| --- | --- |
| **Status** | **Description** |
| New | 新填写的bug的初始状态 |
| Closed | Bug的最终状态。产品发布前，对全部bug进行验证后，确认bug不需要继续追踪，则标记此状态 |
| Verified | Bug经验证无误后置为verified |
| Open | 当测试人员验证已修改状态的bug仍存在，将其status修改为Reopen时，做此标记 |

## BUG填写规范

登录bugzilla系统后，点击“Enter a new bug report”或点击“New”选择产品名称后进入新bug填写页面。

**1、必填项**

填写一个Bug时，以下项为必填项：Summary、Severity、Version、Component、Priority、Re-producible、Platform、OS。

|  |
| --- |
| * Summary摘要, 概要：需要包括测试模块，测试操作，输入数据，产生错误，要求写的很精练并且要一眼能看出来是什么问题，以方便开发或者bug统计时方便查看。 * Severity严重性：该项设置时需要根据该规范提到的要求设置 * Version：指发现bug时的项目的版本号 * Component：指发现bug的所属模块 * Priority：解决bug的优先级，该项的设置需要考虑bug对应的需要，以重要需要用户使用频率高的需要不出问题，不影响系统运行为标准考虑指定合适的级别 * Re-producible：指bug出现的频率 * Platform、OS：指发现该bug所用平台及操作系统版本 |

**2、BUG中Description说明**

当填写Bug时，Description要尽量详细，以避免写的含糊不清导致反复交流填写时要注意以下几点：

|  |
| --- |
| * 位置：首先应说明操作进行的位置，通常是系统中的某一模块。另外是具体的出错位置，可能是某一字段、某一页面... * 操作：详细的、有次序的、每一步的操作步骤，包括输入的数据 * 现象：具体的错误描述，包括界面显示、错误信息 * 日志：当用例发生错误时，程序运行的日志。当日志显示Exception时，必须加上日志。 * 附件：该有附件的尽量粘贴附件或者当错误发生时拷贝屏幕作为附件上传，这样开发才会更直接分析问题。 |

## BUG跟踪

**1、bug状态改变后必须要有必要的说明**

Bug在产品开发周期中，记录了产品的各种生命状态，有效的描述对产品的维护及修改很有帮助，测试人员在修改bug状态时必须添加必要的说明，可在comment中添加，当发现bug状态被开发人员改动而没有相应说明的，应追加说明

填写规范：开发人员fix bug时应简述产生问题的原因，尽量填写什么版本解决以及修改时间。测试人员close bug时必须填写close的时间及版本

**2、收到邮件信息后，应该尽快跟踪bug**

Bug的某些属性值发生改变时，bug管理工具（如Bugzillia）会主动发送邮件通知相关人员。如bug新建时，会通知“Assigned to”项指定的人员，可为项目组长或负责该模块的开发人员，并会抄送给相关人员，通过bug的“CC”项来指定。开发人员收到邮件通知后，要对该bug做出适当处理操作，若为自己负责的bug，要将该bug更改状态置为Assigned，表示接受，解决之后将该bug置为Fixed状态；若该bug有其他问题，可根据情况置为WONTFIX或INVALID。经开发人员修改bug的属性值以后，Bugzilla会主动通知相关测试人员，此时测试人员理应第一时间跟踪此bug并做出适当的处理操作（验证、添加说明等），High以上级别必须要修复，WONTFIX 的bug是不是可以应该WONTFIX。Invalid 要确认是否无效等，不是，也要说明情况，置为相应的状态。

**3、验证bug必须保持条件的统一性**

Bug在已经修复或等待重现时，在验证及重现的过程中必须保持测试条件的统一性，尽量使用发现问题时的数据、测试环境及测试人员

应该要及时补充用例，如果自动化能实现，最好把bug用例加入自动化，以防止以后更多版本时，在没有该bug的测试要求时，再重复出现已经出现过的bug，另外，回归验证bug时不应该单单是回归bug本身，应该发散型回归测试和bug相关及接近的功能

**4、开展****下一轮测试前，关于bug，必须完成的工作**

所有Bug没有用例对应的，要补充完用例，查看bug状态，不能有new reopen等未处理的状态，如果有则需要开发进行处理后才进行下一轮测试。