纽劢软件缺陷管理

改记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本号 | 变更说明 | 修改人 |
| 2019-11-06 | V1.0 | 初始版本 | 吕莹莹 |
|  |  |  |  |
|  |  |  |  |

批准者

此文档需要以下人员批准

|  |  |  |
| --- | --- | --- |
| 姓名 | 职务 | 备注 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

分发者

此文档组需要发给以下部门或单位人员

|  |  |  |
| --- | --- | --- |
| 项目角色 | 姓名|邮箱|分机 | 备注 |
| 项目经理 |  |  |
| 产品经理 |  |  |
| 架构设计 |  |  |
| 研发负责 |  |  |
| 测试负责 |  |  |

[纽劢软件缺陷管理目录 1](#_Toc24127925)

[1. 综述 3](#_Toc24127926)

[1.1 流程目的 3](#_Toc24127927)

[1.2 流程范围 3](#_Toc24127928)

[1.3 流程开始 3](#_Toc24127929)

[1.4 流程结束 3](#_Toc24127930)

[2. 缺陷定义 4](#_Toc24127931)

[2.1缺陷状态 4](#_Toc24127932)

[2.1.1 Jira缺陷流程图 4](#_Toc24127933)

[2.1.2 JIRA缺陷状态描述 4](#_Toc24127934)

[2.2故障解决结果 5](#_Toc24127935)

[2.3 JIRA状态与解决结果 6](#_Toc24127936)

[3. 缺陷严重程度分类及定义 7](#_Toc24127937)

[4. 缺陷分析 8](#_Toc24127938)

[4.1.1 缺陷引入阶段 8](#_Toc24127939)

[4.1.2 缺陷根源分析 8](#_Toc24127940)

## 综述

缺陷不仅仅指软件的Bug，还包括需求、设计上的问题等；通常使用缺陷管理系统管理软件开发过程中所发现的缺陷。NullMax使用JIRA工具管理所有软件缺陷。

## 流程目的

缺陷管理流程的目的是规范在缺陷管理过程中各方的职责和任务，提高缺陷管理的效率。同时，成功识别和解决测试发现的缺陷从而提高产品质量，在测试过程中发生的所有的缺陷有需要进行记录、分析、并查找根本原因，利用缺陷的相关度量指标进行统计分析，预测下一阶段的缺陷发生趋势。缺陷记录和相应的缺陷报告为项目和产品提供主要的管理信息，也能对于缺陷的解决方案进行优先级排序。

## 流程范围

本流程的适用范围包括功能测试、集成测试、整车测试和Demo展示，适用于所有参与产品质量人员。

## 流程开始

整个软件研发生命周期内任何测试活动开始，测试执行人员识别出缺陷或即成上一版本不修复缺陷，本流程即开始。

## 流程结束

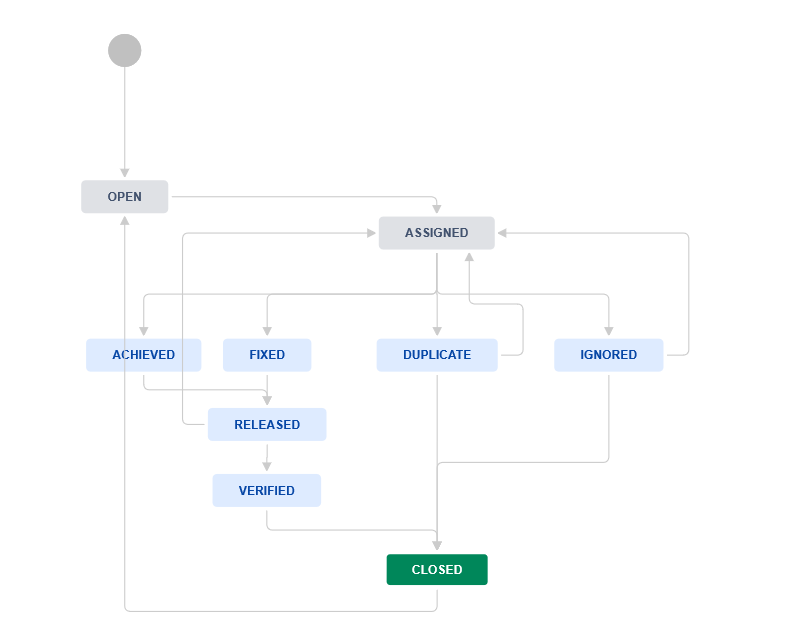
缺陷的JIRA状态为已关闭，即为缺陷在缺陷生命周期中结束。

缺陷的JIRA状态为已关闭，但解决结果为问题遗留暂不修复，视为缺陷在本项目流程中结束，但是在产品维度未终结，后续的相关项目中该缺陷需要被重新打开进行修复、验证。

## 缺陷定义

### 2.1缺陷状态

### Jira缺陷流程图



### JIRA缺陷状态描述

目前公司的缺陷状态分为以下十种：

|  |  |  |
| --- | --- | --- |
| 序号 | 状态 | 描述 |
| 1 | 已提交Open | 新建故障，已经提交 |
| 2 | 开发安排中Assigned | 指派故障给具体解决人员 |
| 3 | 已完成Achieved | 故障已经在最新版本解决 |
| 4 | 已修复Fixed | 故障修改代码已经提交 |
| 5 | 已发布Released | 故障代码已经进入主分支 |
| 6 | 已验证Verified | 故障在新的版本验证通过 |
| 7 | 重复Duplicate | 故障与其它已知故障重复 |
| 8 | 忽略Ignored | 非故障或者本问题本周期内不解决 |
| 9 | 关闭Closed | 缺陷在本生命周期内结束 |
| 10 | 重新打开Reopen | 复测方案不通过，重新打开缺陷 |

### 2.2故障解决结果

缺陷生命周期内，故障应该有相应的处理结果。

|  |  |  |
| --- | --- | --- |
| 序号 | 状态 | 描述 |
| 1 | 未解决 | 新发现或者重新打开的缺陷，还未被理 |
| 2 | 问题遗留且暂不修复 | 方案变更等，本版本内不解决 |
| 3 | 重复问题 | 重复的缺陷，关联到主故障 |
| 4 | 无法复现 | 现有LOG不足以分析且故障在最近版本中不能复现 |
| 5 | 非软件问题 | 确认非问题 |
| 6 | 需求变更 | 确定是需求变更 |
| 7 | 描述不完整 | 缺陷内的信息不充分，不能支持开发分析 |
| 8 | 已解决 | 已经被缺陷关系人处理修复的缺陷 |
| 9 | 已经验证通过 | 缺陷报告人验证开发已经修复缺陷，并关闭 |

在解决结果被设置为以上选项时，需要确认相关的依据和证据已经被提交到JIRA中，才允许关闭缺陷：

1. 设置为“问题遗留且暂不修复”的缺陷，必须是在JIRA中留下技术委员会或者相应技术负责人同意遗留的沟通记录截图。最后由测试相关人员确认，允许在本项目中关闭缺陷。
2. 设置为“问题重复”的缺陷，必须在备注中填写被重复的缺陷单号，并由相应测试人员确认后允许关闭；
3. 设置为“无法复现”的缺陷，开发人员必须和测试人员充分沟通，由测试负责人员确认后允许关闭。
4. 设置为“非软件问题”的缺陷，必须在备注中说明非问题的详细原因，可以添加附件说明。由相关人员确认后允许关闭。
5. 设置为“需求变更”的缺陷，必须在备注中填写需求变更的JIRA号，并由相关人员确定后关闭。
6. 设置为“描述不完整”，开发人员必须和测试人员充分沟通后关闭。

### 2.3 JIRA状态与解决结果

#### 2.3.1缺陷状态切换

所有缺陷都需要有一个最终状态，包括需求或者方案变更等。主要状态的切换以及相应的处理意见请参看下表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 已提交 | 开发安排中 | 重新打开 | 开发解决 | 已关闭 |
| 未解决 | √ |  | √ |  |  |
| 问题遗留暂不修复 |  |  |  | √ | √ |
| 重复问题 |  |  |  | √ | √ |
| 无法复现 |  |  |  | √ | √ |
| 非问题 |  |  |  | √ | √ |
| 需求变更 |  |  |  | √ | √ |
| 描述不完整 |  |  |  | √ |  |
| 已解决 |  |  |  | √ |  |
| 已验证通过 |  |  |  |  | √ |

#### 2.3.2遗留缺陷定期处理

各部门研发负责人有责任参与和推动遗留缺陷清理工作。测试负责人需要定期统计JIRA中对应的系统遗留缺陷清单，并发送给各部门研发负责人，各负责人需要评估因方案修改而遗留的缺陷是否需要继续解决，如果暂不解决，需要备注原因方案变更，并及时关闭。

## 缺陷严重程度分类及定义

缺陷的定义一般可以分为3-5级不等。公司统一分为四级：P3-一般、P2-严重、P1-失效、P0-致命。

|  |  |  |
| --- | --- | --- |
| 严重程度 | 定义描述 | 响应时间要求 |
| 致命 | * 系统无法启动，系统崩溃、刹车失灵等致命错误。 * 系统不减速，有碰撞风险 * 模块识别或者控制异常，有安全危险 | 要求立即解决，4小时内必须有反馈。当天解决。如现场demo问题，缩短解决时间，以产品要求为主。 |
| 失效 | * 系统关键性能严重不达标，引起系统挂死或影响系统运行。 * 关键业务逻辑错误。 * 内存溢出。 * 自动驾驶系统中断或非法退出，且无法通过重启恢复。 * 系统死锁。 * 平台与其它模块内部通信接口不通。 | 要求当日必须有反馈。紧急问题需要24小时内必须解决。 |
| 严重 | * 系统中单元模块或者功能有缺失或错误。 * 问题不影响其它模块正常运行。 * 可以有替代办法。   例如：   1. 业务逻辑错误；功能接口错误。 2. 数据处理有偏差，但可以刹停。 3. 功能模块反应时间超出正常合理时间范围，但可控。 4. 重要日志信息记录不正确或应记录而未记录。 5. 界面校验或者提示信息有异常处理。 | 要求48小时内必须有反馈。紧急问题需要72小时内必须解决。 |
| 一般 | * 基本不影响系统的运行和功能实现，但是存在于标准、定义、用户体验不一致。   例如：   1. 辅助说明描述不清楚，提示信息不清晰或者错误。 2. 长时间操作未给客户提示。 3. 某些查询、存储等实时性要求不高的功能性错误。 4. 日志信息不够完整或者清晰。 5. 一般性数据处理错误，一般性程序错误。  * 功能增强与改进，或者建议优化的项，并非系统缺陷。  1. 提示说明不规范。 2. 界面优化。   3、功能易用性建议。 | 无要求 |

## 缺陷分析

对于实车测试和Demo过程中解决的故障需要进行故障分析，并给出解决方案，增加相应的测试用例用以确保发现过的问题以后不会再出现在验收阶段。

### 缺陷引入阶段

|  |  |  |
| --- | --- | --- |
| 序号 | 引入阶段 | 简要描述 |
| 1 | 需求阶段 | 在业务需求和研发需求分析中进行评审，以挖掘需求拆解过程中隐含缺陷。 |
| 2 | 设计阶段 | 产品设计阶段，通过评审、同行评审、checklist等手段，提前发现设计中可能存在的缺陷。 |
| 3 | 开发阶段 | 通过单元测试、代码评审等方式进行动态或静态测试。 |
| 4 | 测试阶段 | 除正常的功能场景测试外，进行缺陷挖掘，从需求覆盖，鲁棒性测试以及探索性测试等 |

### 缺陷根源分析

缺陷根源可用于对测试过程中所发现的缺陷进行根因分析，识别根本原因，并按照度量结果采取有针对性的解决方法和改进措施。

测试过程中产生的缺陷根源和改进措施对应表：

|  |  |  |
| --- | --- | --- |
| 序号 | 根源 | 改进措施 |
| 1 | 需求错误 | 需求评审 |
| 2 | 设计错误 | 产品设计评审 |
| 3 | 编码错误 | 增加单元测试用例或代码评审等动态或静态测试。 |
| 4 | 测试环境错误 | 需求分析增加checklist |
| 5 | 部署错误 | 增加或修改部署指导文档 |
| 6 | 整车测试问题 | 增加软件测试用例 |