N-Gitea

基于Gitea的通知扩展

测试缺陷报告

[v1.0]

北京航空航天大学计算机学院

二〇二〇年五月

文档修改记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本号 | 日期 | 修改章节 | 修改说明 | 修改人 | 审核人 |
| 1.0 | 2020.5.11 |  | 初稿 | 麦梓健 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# 1 简介

### 1.1 缺陷定义

对于N-Gitea的缺陷定义为以下几种情况：

1. 软件未实现需求和规格要求的功能
2. 软件出现了需求规格说明书指明不应当出现的错误
3. 软件实现了需求规格说明书未提及的功能
4. 软件未实现需求和规格说明书未明确提及单应当实现的内容
5. 测试用例执行的时候出现与预期结果不符合的现象

### 1.2 缺陷原因

对于N-Gitea出现的缺陷而言，可能会是由于以下几个方面引起的，需要对其进行分析：

1. 需求规格说明书撰写不规范，或者不合格，没有考虑可能出现的问题，或者没有考虑到要求实现功能之间、与原生功能之间的不兼容问题。
2. 设计问题，可能是模块设计的时候，耦合度太高出现问题、可扩展性不够导致后续扩展出现问题、高层模块依赖了底层细节实现等等。
3. 实际代码问题，业务逻辑有漏洞、没有考虑某处的异常处理、
4. 其他问题，测试步骤不够规范没有严格按照测试步骤、部署环境不兼容强行部署导致使用出现问题、网络环境不够稳定导致测试出现问题等等。

### 1.3 缺陷修复选择

并不是所有的缺陷都值得修复，因为有些缺陷可能会因为各种各样的原因让我们选择不去修复它们：

1. 时间不够，开发时间不够可能是最大的问题，由于项目测试完成到展示时间之间时间比较短暂，而且缺陷的修复所需时间过长，会让我们选择暂时搁置这项缺陷。
2. 不算真正的软件缺陷，不符合我们上述对缺陷的定义。
3. 修复的风险太大，修复该缺陷的方案大概率会导致其他更为严重问题的出现

### 1.4 缺陷的处理流程

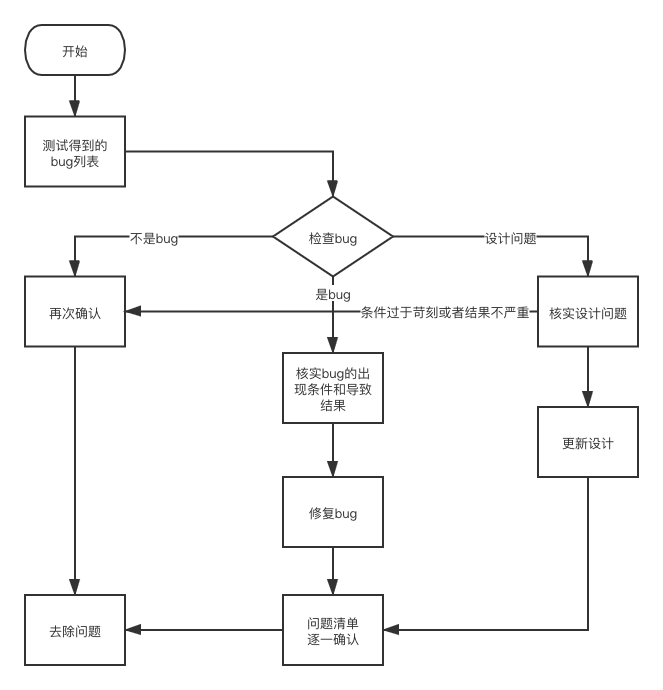


图 1 缺陷的处理流程

如图1所示，我们需要对测试得到的bug列表进行处理：

1. 检查bug，判断bug的类型
   1. 是代码bug，核实bug的条件、结果
   2. 不是bug，让测试组再次确实
   3. 设计问题，更新设计
2. 对问题清单一一确认
3. 去除问题

# 2 缺陷描述

### 2.1 缺陷报告表格说明

表格1为缺陷报告模版

* 缺陷ID：独属于缺陷报告的ID序列
* 缺陷命名：需要在命名中体现缺陷的特点，有较高辨识度
* 缺陷类型：设计问题，代码问题，还是其他
* 缺陷状态：该缺陷的活动状态，活动或者已修复
* 缺陷所属模块：与测试需求说明书保持一致
* 缺陷所属测试用例ID：与发现缺陷时，所执行的测试用例ID保持一致
* 测试环境信息：这一块儿需要尽可能详细
  + 部署的系统环境、版本
  + 使用的开发语言、版本
  + 其他参与测试的软件的版本
* 测试步骤：尽量与测试用例中说明的保持一致，但要与实际相符
* 预期结果：预期结果为按照测试步骤进行测试，没有缺陷后应当得到的输出
* 实际结果：实际得到的输出
* 缺陷处理结果：搁置不修复，已修复，正在修复

表格 1 缺陷报告模版

|  |  |  |  |
| --- | --- | --- | --- |
| 项目组名称 | N-Gitea | | |
| 缺陷ID |  | 缺陷命名 |  |
| 缺陷提交时间 |  | 缺陷类型 |  |
| 缺陷状态 |  | 提交人 |  |
| 缺陷所属模块 |  | 缺陷所属测试用例ID |  |
| 测试环境信息 |  | 缺陷历史版本 |  |
| 简要描述 |  | | |
| 测试步骤 |  | | |
| 预期结果 |  | | |
| 实际结果 |  | | |
| 缺陷处理结果 |  | | |
| 备注 |  | | |