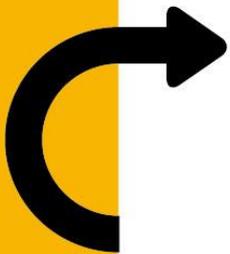


缺陷管理概述 (PDF测试岗位课程)

课程讲师：彭荣涛





第一部分 缺陷管理的定义

第二部分 缺陷管理的目标

第三部分 缺陷管理的关键属性

第四部分 缺陷管理的基本流程

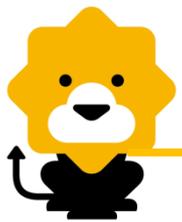
第五部分 缺陷跟踪及分析

第六部分 缺陷管理工具简介



第一部分 缺陷管理的定义

- **缺陷**：软件中的缺陷是软件开发过程中**必不可少**的“副产品”，狭义上指编程过程中造成的错误。随着测试行业的发展，更广泛来讲，是指所有影响用户正常使用或者不符合用户要求的问题
- **缺陷的特性**
 - 软件未达到软件产品需求说明书中指定的要求。
 - 软件出现了软件产品需求说明书中指明不会出现的错误。
 - 软件功能超出了软件产品需求说明书中指定的范围。
 - 软件未达到软件产品需求说明书中虽未指明但应达到的要求。
 - 测试人员认为难以理解、不易使用、运行速度缓慢或者最终用户认为不好的问题。



第一部分 缺陷管理的定义

● 软件缺陷产生的主要原因

- 软件产品本身的抽象性
- 用户、产品、开发及测试缺乏有效沟通，对需求理解不一致
- 需求描述不清楚，评审时没有发现并提出来
- 对产品需求规格说明书不够重视，投入资源较少。
- 需求变更频繁，导致需求变化不一致，前后矛盾。
- 缺乏有效沟通，可能仅部分人员能够清晰的了解产品。
- 编码过程中由于各种主观或客观的原因，引入的错误
- 版本打包时合入错误，环境配置错误

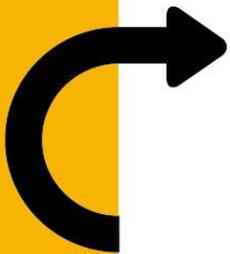


第一部分 缺陷管理的定义

- **缺陷管理：**

缺陷管理是对整个软件研发周期中发现的所有的**缺陷的跟踪、分析和**管理****。妥善的管理软件中发现的缺陷，能够提高开发和测试的测试效率和效果，从而能更好的提高软件质量。

缺陷管理是软件研发流程中不可或缺的一环。



第一部分 缺陷管理的定义

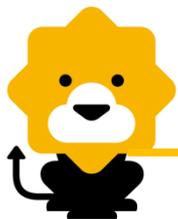
第二部分 缺陷管理的目标

第三部分 缺陷管理的关键属性

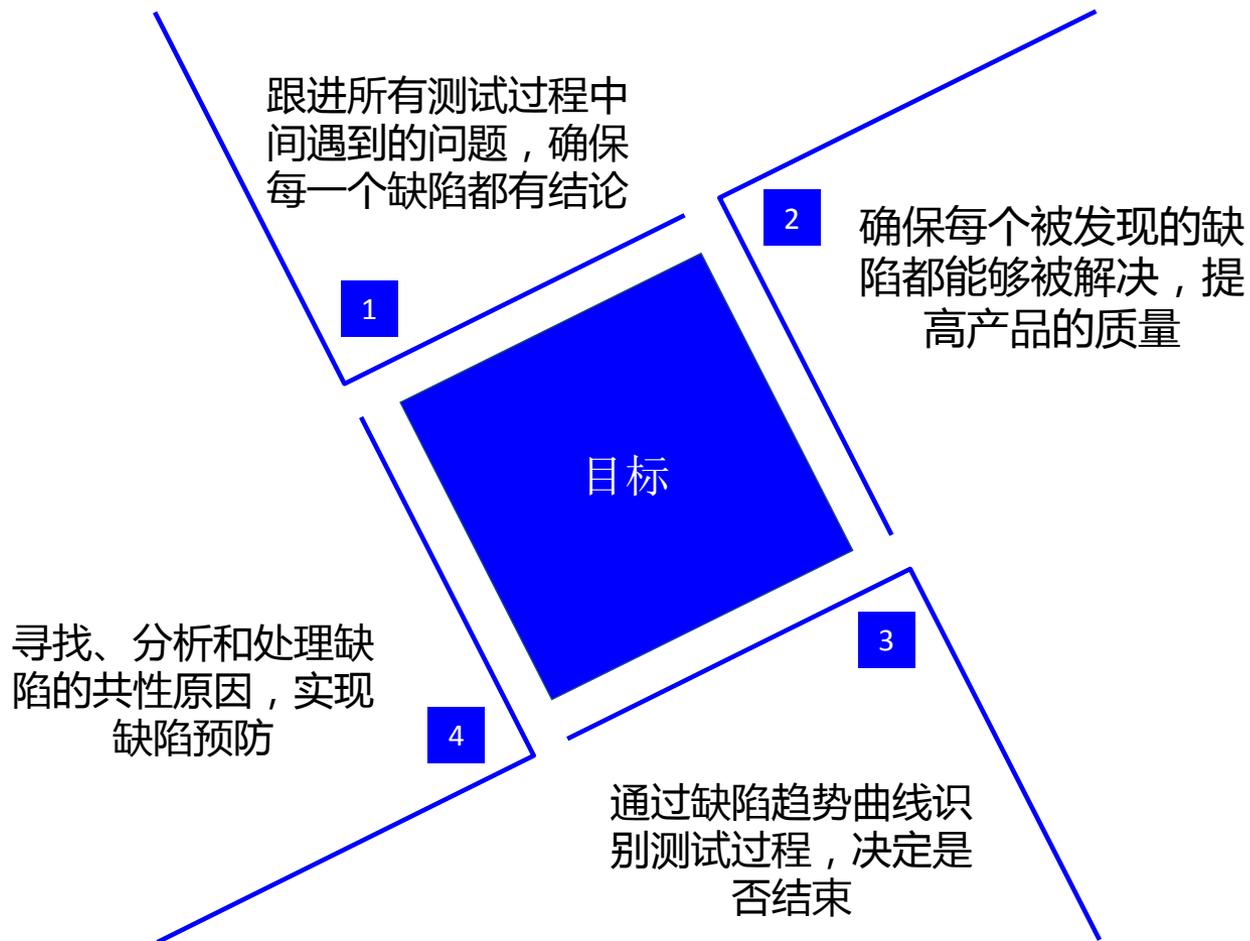
第四部分 缺陷管理的基本流程

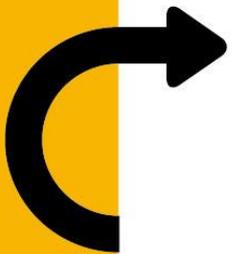
第五部分 缺陷跟踪及分析

第六部分 缺陷管理工具简介



第二部分 缺陷管理的目标





第一部分 缺陷管理的定义

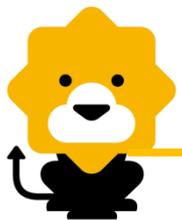
第二部分 缺陷管理的目标

第三部分 缺陷管理的关键属性

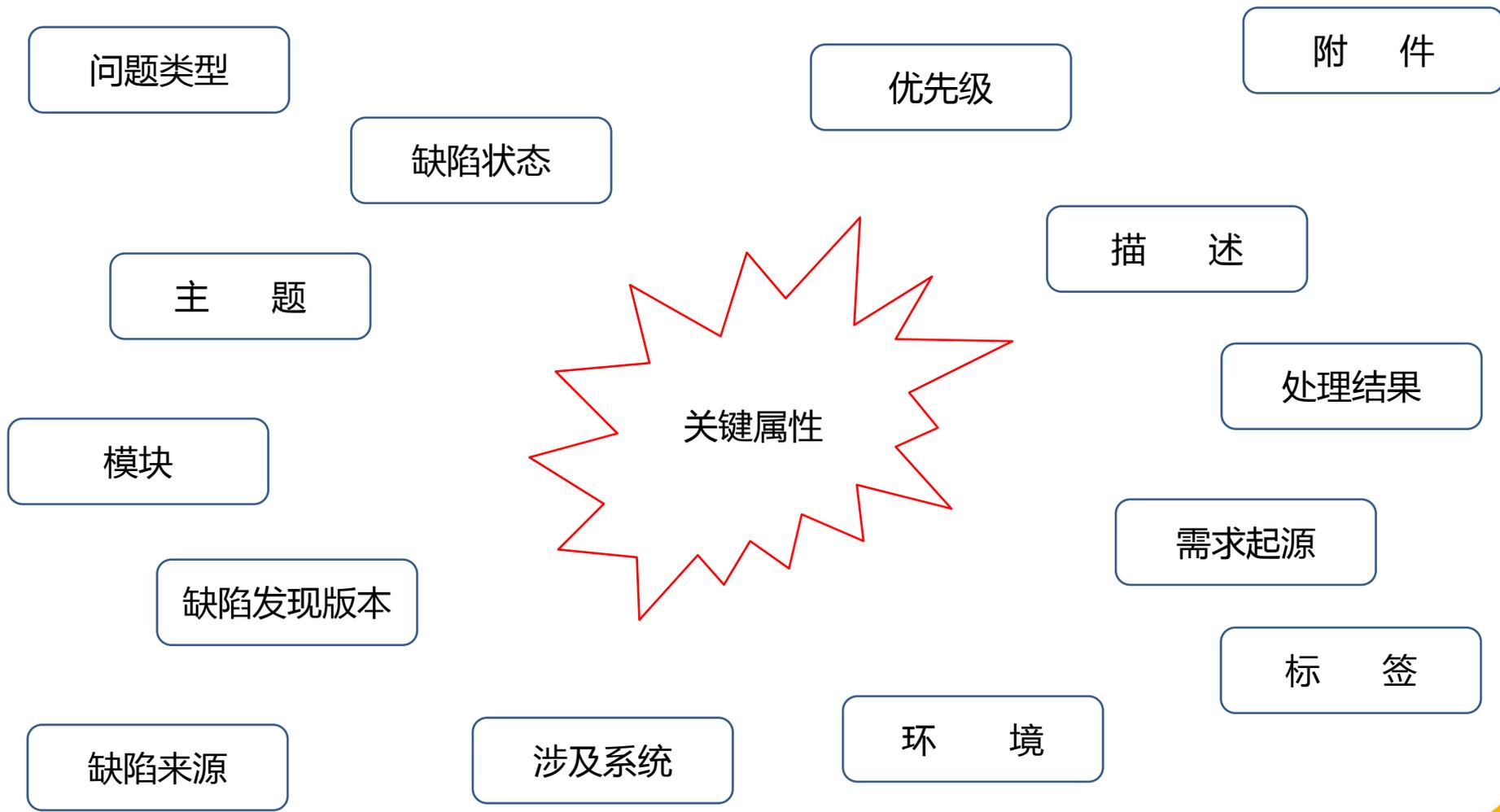
第四部分 缺陷管理的基本流程

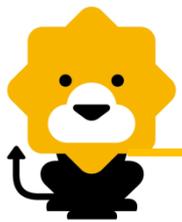
第五部分 缺陷跟踪及分析

第六部分 缺陷管理工具简介



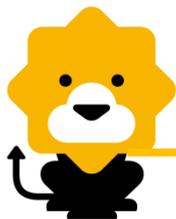
第三部分 缺陷的关键属性





第三部分 缺陷的关键属性

- **问题类型**：通过问题类型对问题进行分类。常见的分类方式有：按环境分类、按缺陷涉及因素进行分类。
 - 按环境分类：SIT-BUG、PRE-BUG、PRD-BUG。
 - 按涉及因素分类：产品需求、系统功能、用户界面（UI）、性能、系统配置等。
- **缺陷等级**：阻塞、致命、严重、一般、提示。
 - 阻塞：系统完成无法操作，导致测试工作无法继续推进。
 - 致命：系统任何一个主要功能完全失效；用户数据受到破坏等。
 - 严重：系统的主要功能部分失效；数据不能保存；系统所提供的功能或服务受到明显影响等。



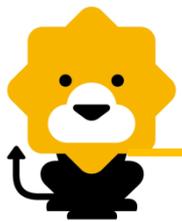
第三部分 缺陷的关键属性

● 缺陷等级

- 一般：系统的次要功能没有完全实现，但不影响用户的正常使用。如提示信息不准确或用户界面差、操作时间长等。
- 提示：使操作者不方便或遇到麻烦，但它不影响功能的操作和执行，如个别不影响理解的错别字、排布不整齐等。

● 缺陷状态

- 已提交：创建问题，并将问题提交给相关的责任人。
- 开发安排中：开发已接受问题，并安排相应的人员处理该问题。
- 开发解决：开发已解决问题，待测试人员验证。
- 重新打开：测试人员验证问题未解决，重新打开问题。
- 已关闭：测试人员验证通过后，关闭问题。



第三部分 缺陷的关键属性

● 缺陷来源

- 需求说明书：需求说明书的错误和不清楚引起的问题。
- 设计文档：设计文档描述不清楚，和需求文档不一致导致的问题。
- 系统集成接口：系统各模块不匹配，开发组之间缺乏协调引起的缺陷。
- 数据库：由于数据库字典、数据库中的错误引起的缺陷。
- 程序代码：纯粹是编写代码引起的缺陷。

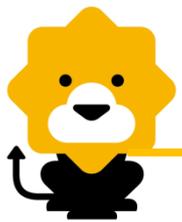
● 缺陷起源

- 需求阶段：在需求阶段发现的缺陷。
- 架构阶段：在架构设计时发现的系统缺陷。
- 设计阶段：在程序设计时发现的系统缺陷。
- 编码阶段：在编码阶段发现的系统缺陷。
- 测试阶段：在测试阶段发现的系统缺陷。



第三部分 缺陷的关键属性

- **缺陷描述**：为便于处理人员能够快速定位问题，应尽可能详细的描述缺陷的情况。
 - **现象**：清晰的描述清楚权限的表象或问题，建议直接截图。
 - **操作步骤**：描述该问题重现的步骤，以便处理人员可以方便的定位问题。
 - **测试数据**：提供该缺陷所使用的测试数据。
 - **预计结果**：按照系统需求的正确结果。
 - **实际结果**：不符合系统需求的实际输出结果。
- **附件**：有助于处理人员定位和解决问题所需的信息。例如：界面截图、日志文件等。



第三部分 缺陷的关键属性

● 解决结果

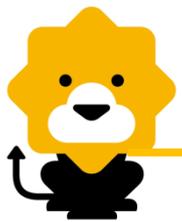
- 未解决：缺陷已提交，尚未确认解决结果。
- 已解决：处理人员解决问题后，修改状态为已解决，待测试人员验证。
- 问题遗留暂不处理：问题存在但对项目影响较少，由项目组评估，确认暂时不处理，待后期优化。
- 无法复现：处理人员接受问题后，无法重新问题，导致问题无法解决，则状态置为无法复现，此类问题在项目报告中，应作为风险提出。
- 非问题：处理人员认为当前描述场景未非问题，则测试需和产品经理确认该问题后关闭。
- 需求变更：一般在需求不一致的情况下产生，因要求产品提交需求变更后再次关闭问题。
- 已验证关闭：对已解决的问题进行验证，验证通过后关闭。



第三部分 缺陷的关键属性

- **解决结果**

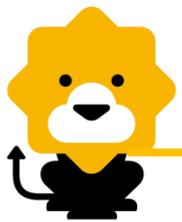
- 注：解决问题时，处理人员在解决后修改状态时，必须注明：解决结果、缺陷原因、开发定位（定位思路和方法；问题原因；修改内容）。其中修改内容必须明确清晰，已确保测试经理能明确所需验证的场景。



第三部分 缺陷的关键属性

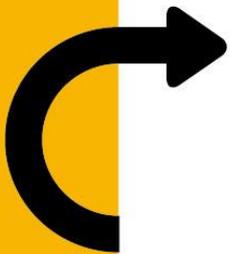
● 有效的缺陷描述特征

- 单一性：每一个报告只针对一个软件缺陷
- 完整统一：提供完整的缺陷描述信息，描述软件缺陷毕竟的细节。规法统一、通俗易懂、简单易行。
- 可以再现：不要忽视或者省略任何一项操作步骤，特别是关键操作一定要描述清楚，确保开发人员按照所描述的步骤可以再现缺陷
- 客观：针对缺陷客观描述，不评价程序好坏，不掺杂主观思想
- 特定条件：注明缺陷发生的特定条件，比如测试环境、工具、数据
- 解决建议：有经验的测试人员可以对缺陷修复提出建议。（可选）



第三部分 缺陷的关键属性

- **开发人员无法修改或拒绝修改的缺陷**
 - 开发人员无法重现或者现象难以捕捉
 - 没有明确的报告说明重复缺陷的步骤
 - 开发人员无法读懂的缺陷报告
 - 缺陷描述有主观针对性和指向性



第一部分 缺陷管理的定义

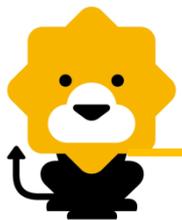
第二部分 缺陷管理的目标

第三部分 缺陷管理的关键属性

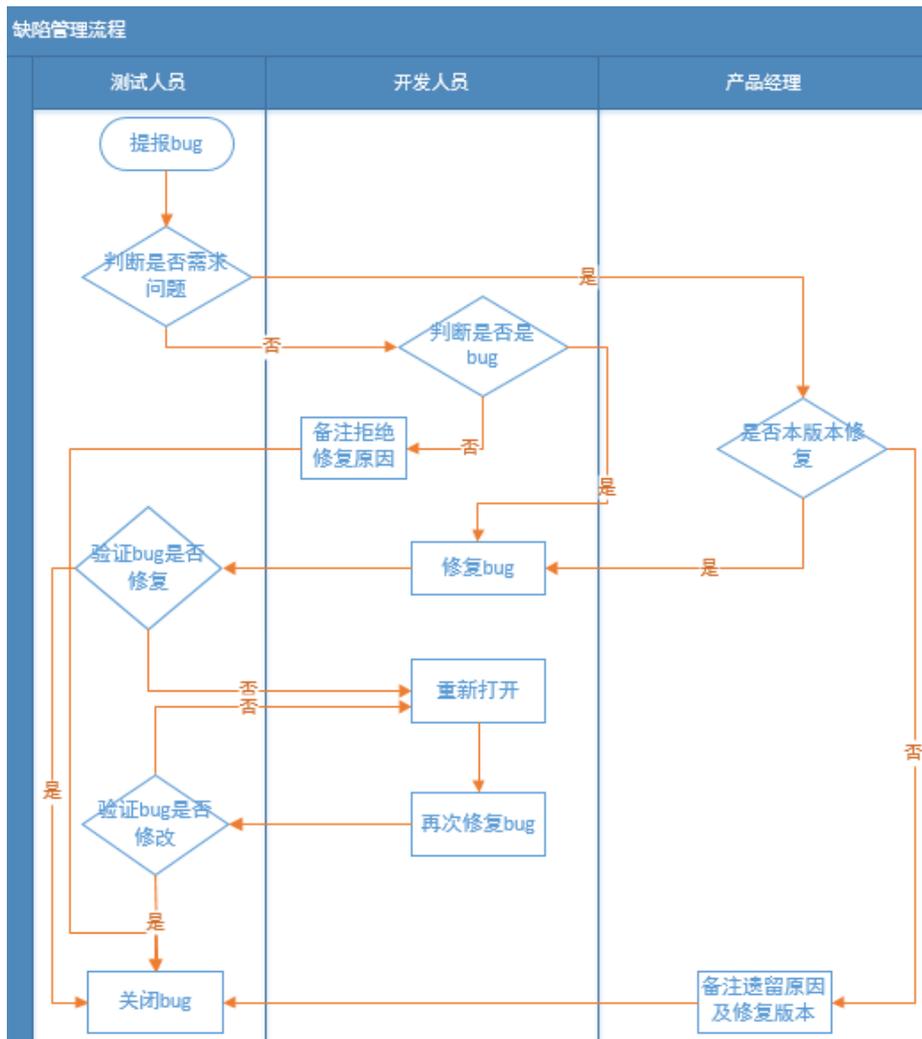
第四部分 缺陷管理的基本流程

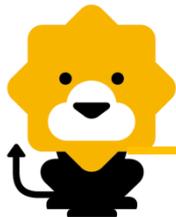
第五部分 缺陷跟踪及分析

第六部分 缺陷管理工具简介



第四部分 缺陷管理的基本流程





第四部分 缺陷管理的基本流程

- **简易流程：**

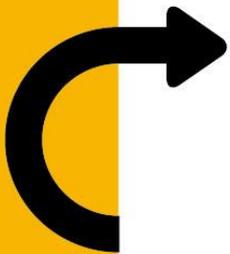
测试人员---提交--开发人员----修改---测试人员---（回归闭环）。

注：一般适用于项目或迭代前期测试人员介入，方便问题。即时修改，快速开发。

- **复杂流程：**

测试人员---提交—测试经理---审核---开发经理----分配---开发人员---修复---复核人员----开发经理----审核----测试经理---分配----测试人员---（回归闭环）。

注：一般适用于后期整个项目完成后正式的系统测试，可以方便项目经理审核统计缺陷数据，实时监控版本缺陷情况。



第一部分 缺陷管理的定义

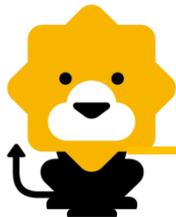
第二部分 缺陷管理的关键属性

第三部分 缺陷管理的目标

第四部分 缺陷管理的基本流程

第五部分 缺陷跟踪及分析

第六部分 缺陷管理工具简介



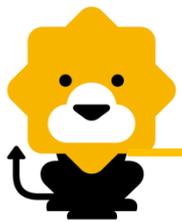
第五部分 缺陷跟踪及分析

- **缺陷分析的定义**

- 测试将软件开发、运行过程中产生的缺陷进行必要的收集，对缺陷的信息进行分类和汇总统计，有针对性的提出遏制缺陷发生的措施，降低缺陷的数量。

- **缺陷分析的目的**

- I. 收集系统缺陷，评估开发质量，为开发部门的考核提供依据。
- II. 分析缺陷分布，找出系统的薄弱环节及导致缺陷的根本原因。
- III. 分析缺陷趋势，评估软件质量及测试过程的效率。
- IV. 跟踪缺陷状态，检验测试及开发人员工作情况，并且检验软件产品是否满足交付条件。
- V. 总结软件缺陷，改进软件研发过程

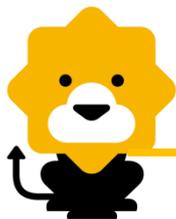


第五部分 缺陷跟踪及分析

- 收集系统缺陷，评估开发质量，为开发部门的考核提供依据
- 收集系统缺陷，对于人员、功能模块、缺陷等级等的缺陷分布情况进行分析，可从部门、人员、功能模块等维度对开发质量进行评估。
- 为确保测试进度，可根据日常评估情况，针对性的对冒烟测试用例进行验证。

模块	提示	严重	一般	阻塞	总计
菜单权限				1	1
发票抬头				2	2
会员				1	1
会员创建			1		1
会员地址	1		4	27	32
可卖标识					1
权限				1	1
政企欠款				3	3
综合订单老			1		1
综合订单新	1		2	19	22
综合订单新 (UI)				5	5
综合订单新 (换异型)			1		1
综合订单新 (券)				1	1
综合订单新 (调回)				2	2
综合订单新 (延保)				8	8
总计	2		9	72	83

经办人	提示	严重	一般	阻塞	总计
陈梅				1	13
彭汪				2	1
张汪			1		1
张汪		1	3		25
张汪				3	3
张汪				6	6
张汪				1	1
张汪		1	4		22
总计		2	9		72

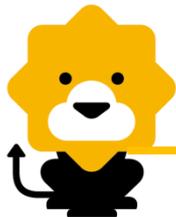


第五部分 缺陷跟踪及分析

- 分析缺陷分布，找出系统的薄弱环节及导致缺陷的根本原因
 - 从模块、缺陷等级、缺陷来源的分布情况，了解在哪些模块比较容易出问题？一般是什么原因导致缺陷的出现？

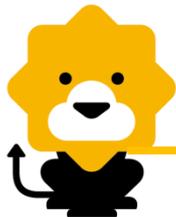
问题原因	提示	严重	一般	阻塞	总计
场景增加	2	2	11		15
非问题			2		2
系统老BUG			1		1
新需求		6	4		10
修改问题后引入					
预期遗漏					
合并版本					
需求变更					
总计	2	9	7		18

模块	提示	严重	一般	阻塞	总计	
菜单权限				1	1	
发票抬头				2	2	
会员				1	1	
会员创建			1		1	
会员地址	1		4	27	32	
可卖标识					1	1
权限				1	1	
政企欠款				3	3	
综合订单老			1	1	2	
综合订单新	1		2	19	23	
综合订单新 (UI)				5	5	
综合订单新 (换异型)			1	1	2	
综合订单新 (券)				1	1	
综合订单新 (调回)				2	2	
综合订单新 (延保)				8	8	
总计	2		9	72	85	



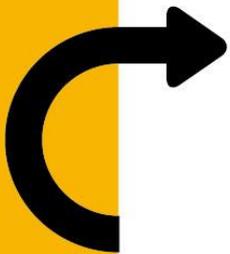
第五部分 缺陷跟踪及分析

- **分析缺陷趋势，评估软件质量及测试过程的效率**
 - 通过对模块缺陷趋势的分析，对产品的质量和稳定性进行评估。
 - 通过每轮测试的缺陷趋势，分析测试过程中存在的问题，及对测试效率影响的因素。
 - 通过对版本缺陷趋势的分析，评估项目是否具备上线条件。



第五部分 缺陷跟踪及分析

- **跟踪缺陷状态，检验测试及开发人员工作情况，并且检验软件产品是否满足交付条件**
 - 所有缺陷必须都要有解决结果（已验证关闭、遗留暂不修复、非问题等）
 - 对于阻塞问题，开发必须第一时间解决；对于一般问题，开发也必须要求在2小时内解决；非特殊的问题不允许出现拖到第二天解决的情况。
 - 测试人员应跟进问题的处理情况，当问题未及时处理时，理应询问处理人员当前进度，预判风险，当存在风险时应及时的反馈给测试经理/测试组长/部门负责人。



第一部分 缺陷管理的定义

第二部分 缺陷管理的关键属性

第三部分 缺陷管理的目标

第四部分 缺陷管理的基本流程

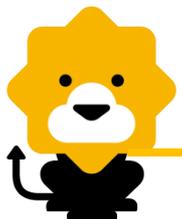
第五部分 缺陷跟踪及分析

第六部分 缺陷管理工具简介



第六部分 缺陷管理工具简介





第六部分 缺陷管理工具简介

创建问题

创建问题

创建问题

配置字段

经办人  自动

分配给我

环境

例如操作系统,软件环境或者相关的硬件。

缺陷发现版本

输入字符来获得可能匹配的列表,或点击下拉列表选择

标签

输入内容来查找或创建标签,或点击下拉列表中选择一个建议的标签。

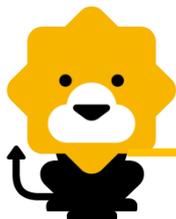
涉及系统

输入内容来查找或创建标签,或点击下拉列表中选择一个建议的标签。

创建另一个

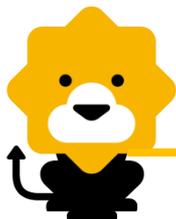
创建

取消



第六部分 缺陷管理工具简介

The screenshot displays a web-based defect management tool interface. At the top, there is a navigation bar with the title "916版本优化" (916 Version Optimization) and a search bar containing the filter "project = POSOPT AND labels = 916版本优化". Below the search bar, a list of issues is shown, with the first issue selected: "POSOPT-1836 开单界面展示的年限延保价格与生成延保订单取的延保价格不一致" (POSOPT-1836: Inconsistent price for extended warranty shown on the order form vs. the price taken from the generated order). The details for this issue are visible, including its priority (中优先级 - Medium), status (无 - None), and tags (916版本优化 - 916 Version Optimization). The interface also shows a sidebar with navigation options like "新建过滤器" (New Filter) and "搜索过滤器" (Search Filter), and a bottom section with "All Criteria" and "Export" options.



第六部分 缺陷管理工具简介

分配给我的
NSF综合服务
未解决问题

未解决问题

+ 添加小工具 版式布局 工具

2维过滤器统计: 916版本优化	2维过滤器统计: NSF-未解决问题
经办人	标签

小工具目录 获取更多小工具 · 创建自己的小工具

所有 (51)

- Charts (13)
- JIRA (47)
- Other (4)
- Wallboard (20)

马上添加

2维过滤器统计
被 Atlassian
显示指定过滤器返回的结果
<http://jira.cnsuning.com/rest/gadgets/1.0/g/com.atlassian.jira.gadgets.two-dime...>

Agile Classic Card View
被 Atlassian
Display a specific issue or all the issues in a version.
<http://jira.cnsuning.com/rest/gadgets/1.0/g/com.pyxis.greenhopper.jira.greenho...>

Agile Classic Cumulative Flow Chart
被 Atlassian
View the cumulative flow chart for a project (Wallboard capable)
<http://jira.cnsuning.com/rest/gadgets/1.0/g/com.pyxis.greenhopper.jira.greenho...>

Agile Classic Days Remaining
被 Atlassian

关闭

版式布局

选择面板布局

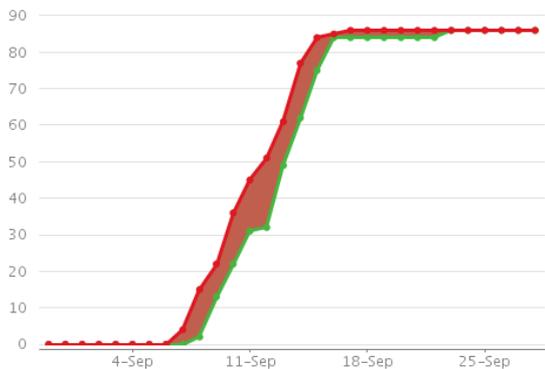
关闭



第六部分 缺陷管理工具简介

● 创建和解决问题对比图

创建与解决的问题: 916版本优化



说明：

- 红色曲线表示随日期增加所提交的bug数累计分布，绿色线条表示随日期增加所解决的bug数累计分布

分析：

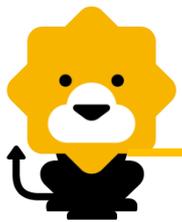
- Bug累计数随日期的增加还在持续的快速增长，并且红色曲线斜率多处区域大于 45° ，说明产品仍存在较多缺陷，质量并没有稳定下来。
- 两条曲线斜率多处区域均大于 45° ，说明测试和开发的效率都还是不错的
- 后期平滑并且两线相交，说明系统基本区域问题并都得到解决，具备上线条件。

其他情况：

- 情况一：两条曲线之间的间距越来越小，且红色曲线的斜率趋于平缓
- 分析：质量越来越稳定，且可以预见两条曲线有交织的可能性，可以考虑关闭项目测试。
- 情况二：两条曲线之间的间距越来越大，且红色曲线斜率并没有放缓趋势
- 分析：产品质量比较差，需要及时做出修改和调整，使产品质量相对稳定下来。

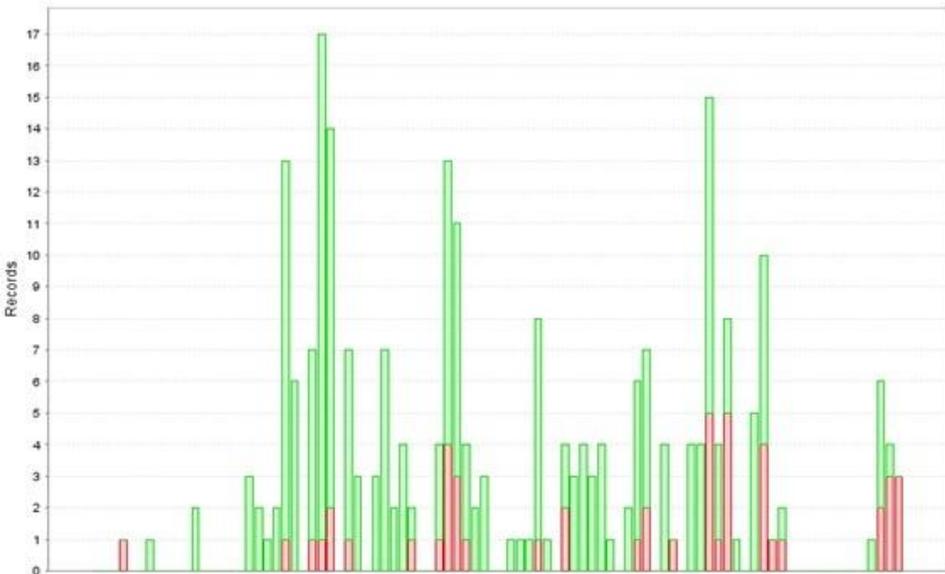
情况三：两条曲线之间间距稳定，但是曲线斜率趋于平缓

分析：开发遇到了技术挑战，效率开始降低。由于模块不能及时发布，同时也影响了测试效率。



第六部分 缺陷管理工具简介

● 最新问题图

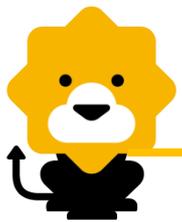


说明：

- 柱子显示每天提交的问题数量；绿色部分表示已解决的问题；红色部分表示未解决的问题。

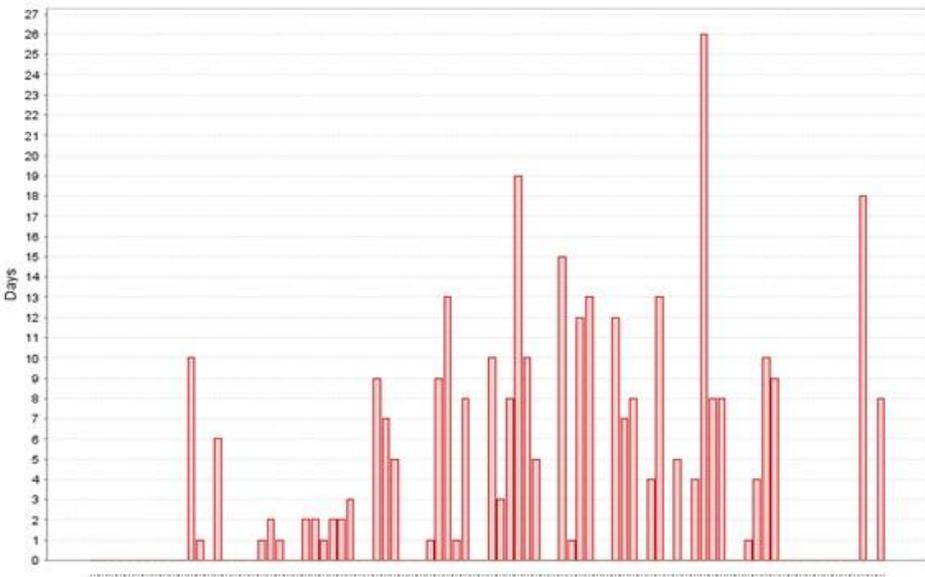
分析：

- 如果某一天的提交的bug数量非常多，说明这一天提交的测试版本中可能是添加了某个新的功能点，且该功能点处于不稳定状态；还一种可能就是开发的某一处的修改带来了连锁反应，将其它稳定之处也连带改的不稳定了，从而注入了新的bug（在实际的工作中，确实遇到过这样的问题，开发为了修改一个bug，将起先版本中稳定的功能点也改坏了）。



第六部分 缺陷管理工具简介

● 问题解决时间

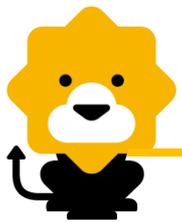


说明：

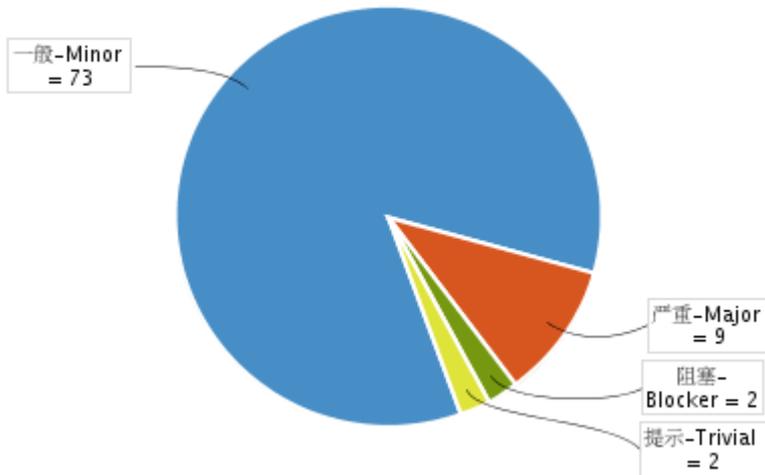
- 此图显示了修复bug所花费的平均天数，横坐标代表bug关闭的日期

分析：

- 绝大部分的bug修复花费了较长的天数，说明此项目对于开发团队可能是全新的领域，诸多问题对于他们来说都是非常挑战；如果此种情况不存在，那么开发的效率可能存在问题，可能是资源受限，如人力不足等。



● 问题解决时间



说明：

- 缺陷等级：阻塞>严重>一般>提示

分析：

- 当阻塞问题较多时，说明转测版本较差，并且有可能未进行冒烟测试。
- 当阻塞问题和严重问题占比较高时，说明质量控制需加强。

Thanks!

