

## 第二部分 测试基础

### 2.10 黑盒测试—缺陷管理



# 本节教学目标



- 掌握缺陷的定义
- 掌握缺陷的管理



1

什么是bug

2

缺陷产生的原因及修复的成本

3

缺陷报告的编写

4

缺陷的严重性和优先级

5

缺陷状态及周期性

6

缺陷工具的使用

# 什么是缺陷



- bug的由来
- IEEE对缺陷的定义
- 通俗理解缺陷
- 缺陷的别名

# 最早的bug



- 第1代计算机，由庞大而昂贵的真空管组成，计算机运行产生的光和热，吸引了一只小虫子钻进真空管里，导致整个计算机无法工作，Grace Hopper找到并取出后，恢复工作，bug这个词就流传下来

# IEEE对缺陷的定义



- 从产品内部看：

- 软件缺陷是软件产品开发或维护过程中所存在的错误、毛病等各种问题；

- 从产品外部看：

- 软件缺陷是系统所需要实现的某种功能的失效或违背。



- 满足规则之一称发生一个缺陷
  - 软件未实现产品说明书要求的功能
  - 软件出现了产品说明书指明不应该出现的错误
  - 软件实现了产品说明书未提到的功能
  - 软件未实现产品说明书虽未明确提及但应该实现的目标
  - 软件难以理解，不易使用，运行缓慢或者--最终用户会认为不好



- 缺点 (defect)
- 偏差 (variance)
- 谬误 (fault)
- 失败 (failure)
- 问题 (problem)
- 矛盾 (inconsistency)
- 错误 (error)
- 毛病 (incident)
- 异常 (anomaly)





- 软件在运行的不同时期有不同含义：
  - **Fault**（缺陷）：静态存在于软件中的问题
  - **Error**（错误）：软件运行中，运行到**fault**，触发产生的错误
  - **Failure**（失效）：**Error**传播到软件外部，使用户观测到失效的行为

Software Fault : A **static** defect in the software (i.e., defect)

Software Error : An incorrect **internal** state that is the manifestation of some faults

Software Failure : **External**, incorrect behavior with respect to the requirements or other description of the expected behavior

# Bug的三个概念

- 输入数组，求数组中数的平均值：

```
public static void CSta (int [ ] numbers)
{
    int length = numbers.length;
    double mean, sum;

    sum = 0.0;
    for (int i = 1; i < length; i++)//i=0
    {
        sum += numbers [ i ];
    }
    mean = sum / (double) length;

    System.out.println ("mean: " + mean);
}
```

- 测试输入：[0, 4, 5]

sum=0+4+5

sum=4+5

mean=3

mean=3

存在缺陷，  
但并没有  
引发错误

# Bug的三个概念

```
public static void CSta (int [ ] numbers)
{
    int length = numbers.length-1;
    double mean, sum;

    sum = 0.0;
    for (int i = 0; i < length; i++)
    {
        sum += numbers [ i ];
    }
    mean = sum / (double) length;

    System.out.println ("mean: " + mean);
}
```

• 测试输入: [3, 5, 4]

sum=3+5+4

sum=3+5

mean=4

mean=4



存在错误  
并没有引  
发失效

# Bug的三个概念



```
public static void CSta (int [ ] numbers)
{
    int length = numbers.length;
    double mean, sum;

    sum = 0.0;
    for (int i = 1; i < length; i++)//i=0
    {
        sum += numbers [ i ];
    }
    mean = sum / (double) length;

    System.out.println ("mean: " + mean);
}
```


• 输入: [3, 5, 4]

sum=3+5+4

sum=5+4

mean=4

mean=3



引发**错误**  
进而引发  
**失效**



- **Execution**(执行)
- **Infection** (感染)
- **Propagation** (传播)
- **Fault** (缺陷)
- **Error** (错误)
- **Failure** (失效)

## PIE Model

- **Execution/Reachability** : The location or locations in the program that contain the fault must be reached
- **Infection** : The state of the program must be incorrect
- **Propagation** : The infected state must propagate to cause some output of the program to be incorrect



1

什么是缺陷

2

缺陷产生的原因及修复的成本

3

缺陷报告的编写

4

缺陷的严重性和优先级

5

缺陷状态及周期性

6

缺陷工具的使用



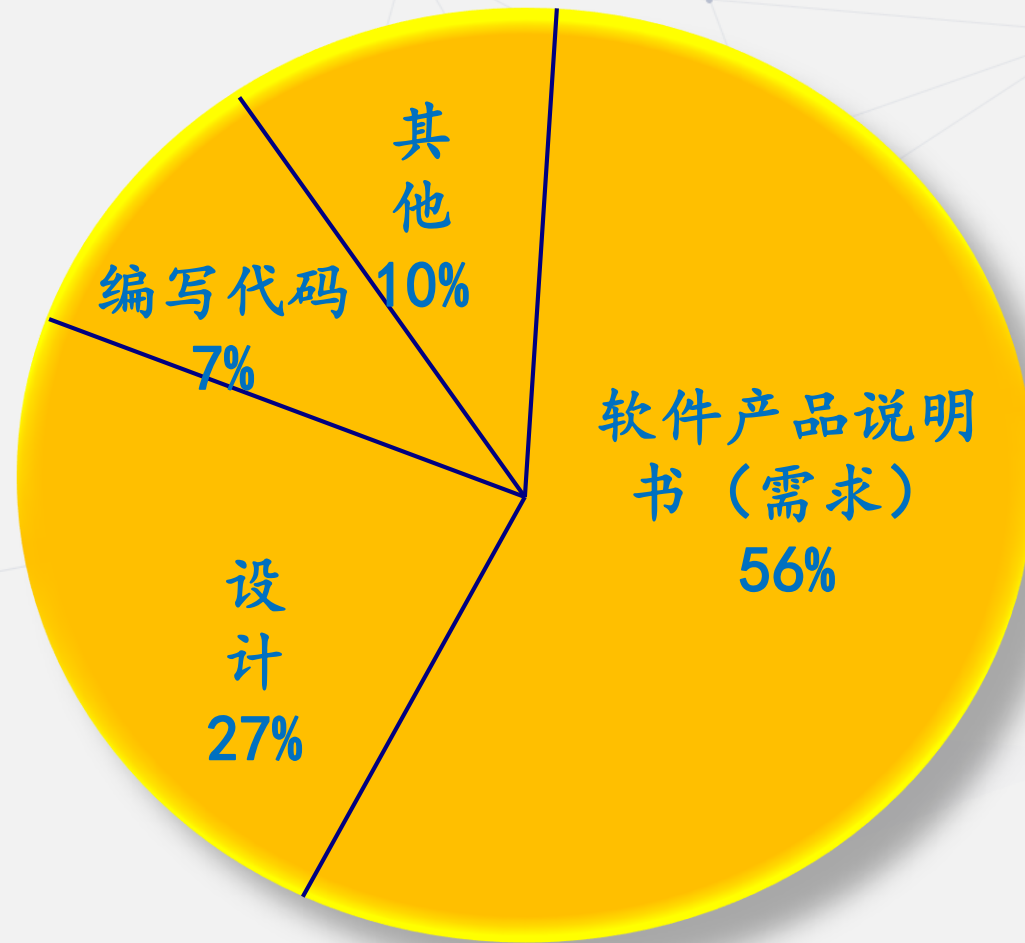
# 缺陷产生的原因



- 人与人的交流不够或没有充分的文档资料
- 项目没有被很好地理解；计划不周，最终导致进度拖延
- 软件可靠性缺少度量的标准，质量无法保证
- 软件难以维护、不易升级
- 等等



# 软件缺陷--构成

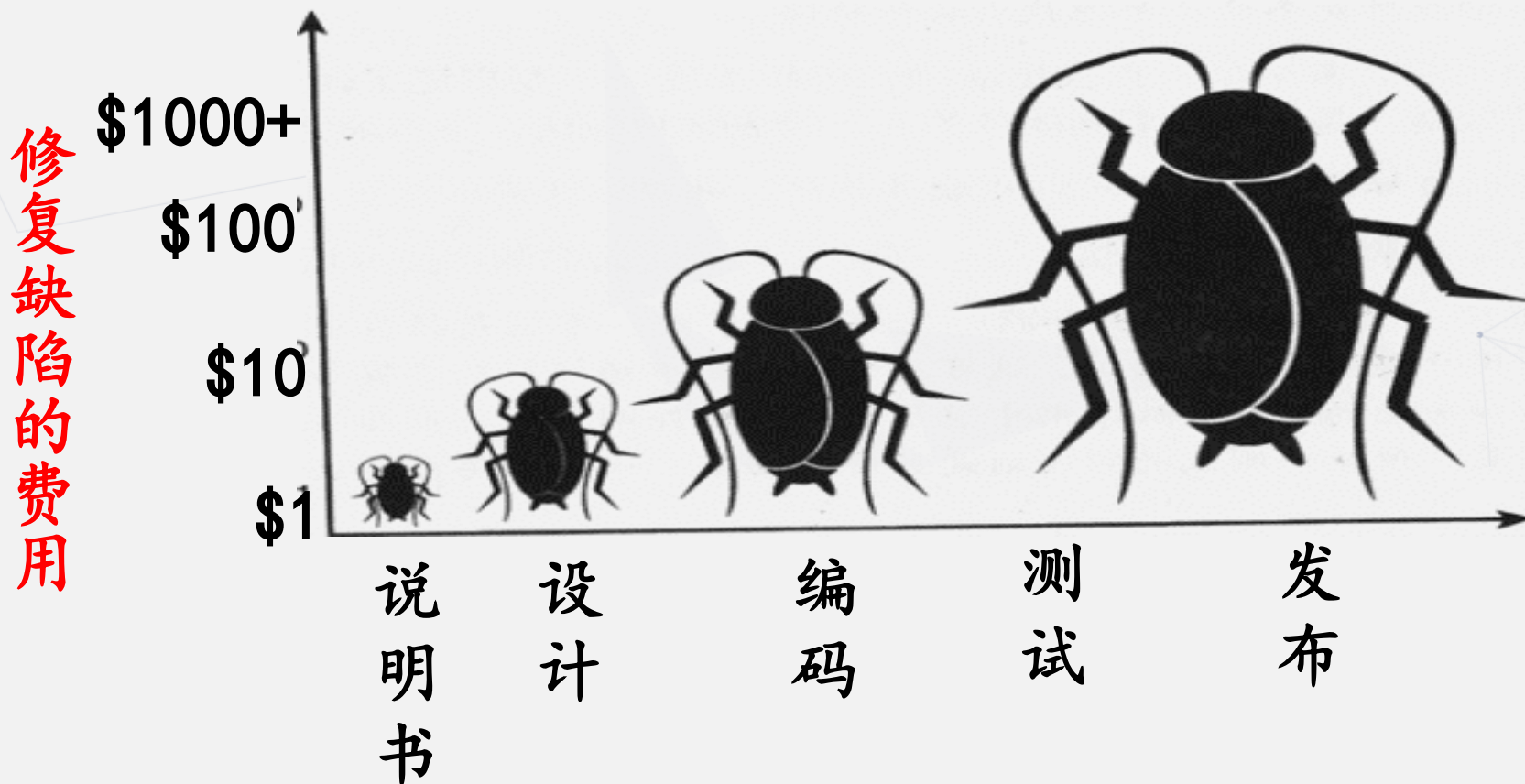


缺陷分布图



# 软件缺陷——成本

- 软件在从需求、设计、编码、测试一直到交付用户公开使用后的过程中，都有可能产生和发现缺陷
- 随着整个开发过程的时间推移，更正缺陷或修复问题的费用呈几何级数增长





- 沟通是前提
- 需求评审很重要
- 设计评审不可少
- 文档更新要及时
- 开发测试要思考



1

什么是缺陷

2

缺陷产生的原因及修复的成本

3

缺陷报告的编写

4

缺陷的严重性和优先级

5

缺陷状态及周期性

6

缺陷工具的使用



- 为什么编写缺陷报告
- 如何提交缺陷报告
- 怎样写缺陷报告

# 为什么写缺陷报告



- 缺陷报告的用途是什么？
  - 记录缺陷
  - 缺陷分类（为解决缺陷分配资源）
  - 缺陷跟踪

# 如何提交bug?

## 空中接龙模式

口头描述  
国内测试管理规范程度低的小企业使用

### 缺点:

不易追踪  
沟通理解易出错  
打乱开发思路

## 流水记帐模式

直接记录, 内容可以记录成  
word, excel 等格式

### 缺点:

反映bug延迟  
延误bug修改时间  
不易管理

## 系统管理模式

使用专业工具,

### 优点:

及时有效修复bug  
可标识、追踪缺陷  
测试员: 直接提交  
程序员: 直接查找

# 怎样编写缺陷报告

- 保证重现缺陷
  - 判断一个缺陷报告撰写好坏的简单方法：让**非缺陷报告撰写者**（技术人员）依据缺陷报告重现缺陷，如果能**简单、迅速的重现**缺陷，表明缺陷报告较好



# 怎样编写缺陷报告

- 分析故障——使用**最少步骤**重现缺陷
  - 减少开发人员重现缺陷的时间
  - 使开发人员更准确的定位缺陷





# 怎样编写缺陷报告

- 包含所有重现缺陷的**必要步骤**
  - 测试人员假定常用的操作步骤开发人员不一定熟悉，省略了必要的步骤常常造成开发人员无法重现缺陷。



# 怎样编写缺陷报告



- 举例：使用“记事本”仅保存“联通”二字后再打开该文件，出现乱码。
  - 描述步骤：
    - 点击“开始”->“程序”->“附件”->“记事本”打开记事本软件；
    - 仅输入“联通”二字后，点击“文件”->“保存”；
    - 在打开的“另存为”对话框中保存文件，退出（文件名、保存位置任意）；
    - 打开保存的文件，出现乱码，不是“联通”二字。

# 怎样编写缺陷报告



- 值得注意的细节
  - 报告小缺陷
  - 报告随机缺陷
  - 不要夸大缺陷



# 怎样编写缺陷报告小结



- 保证重现缺陷
- 分析故障——使用最少步骤复现故障
- 包含所有重现缺陷的必要步骤
- 方便阅读
- 尽量简单—— 一个缺陷一个报告



1

什么是缺陷

2

缺陷产生的原因及修复的成本

3

缺陷报告的编写

4

缺陷的严重性和优先级

5

缺陷状态及周期性

6

缺陷工具的使用



- 缺陷实例

- 雪梨教育平台不能提交作业
- 雪梨教育不能关注粉丝

先解决哪个问题？

# 缺陷严重级别

- 缺陷严重性：表示软件缺陷所造成的危害的恶劣程度
  - **Fatal**：致命的错误，造成系统或应用程序崩溃、死机、系统悬挂，或造成数据丢失、主要功能完全丧失等。
  - **Critical**：严重错误，主要指功能或特性没有实现，主要功能部分丧失，次要功能完全丧失，或致命的错误声明。

# 缺陷严重级别



- **Major:** 主要错误，这样的缺陷虽然不影响系统的使用，但没有很好地实现功能，没达到预期效果。如提示信息不太准确，或用户界面差，操作时间长等。
- **Minor:** 一些小问题，对功能几乎没有影响，产品及属性仍可使用。
- **Suggestion:** 一些友好的建议。





**Bug严重程度和优先级一定成正比吗？**



# 严重性&优先级



- 一般情况：严重程度高的缺陷优先级高
- 特殊情况：不成正比
- 没有必然联系，结合实际综合思考

缺陷严重性



缺陷优先级



1

什么是缺陷

2

缺陷产生的原因及修复的成本

3

缺陷报告的编写

4

缺陷的严重性和优先级

5

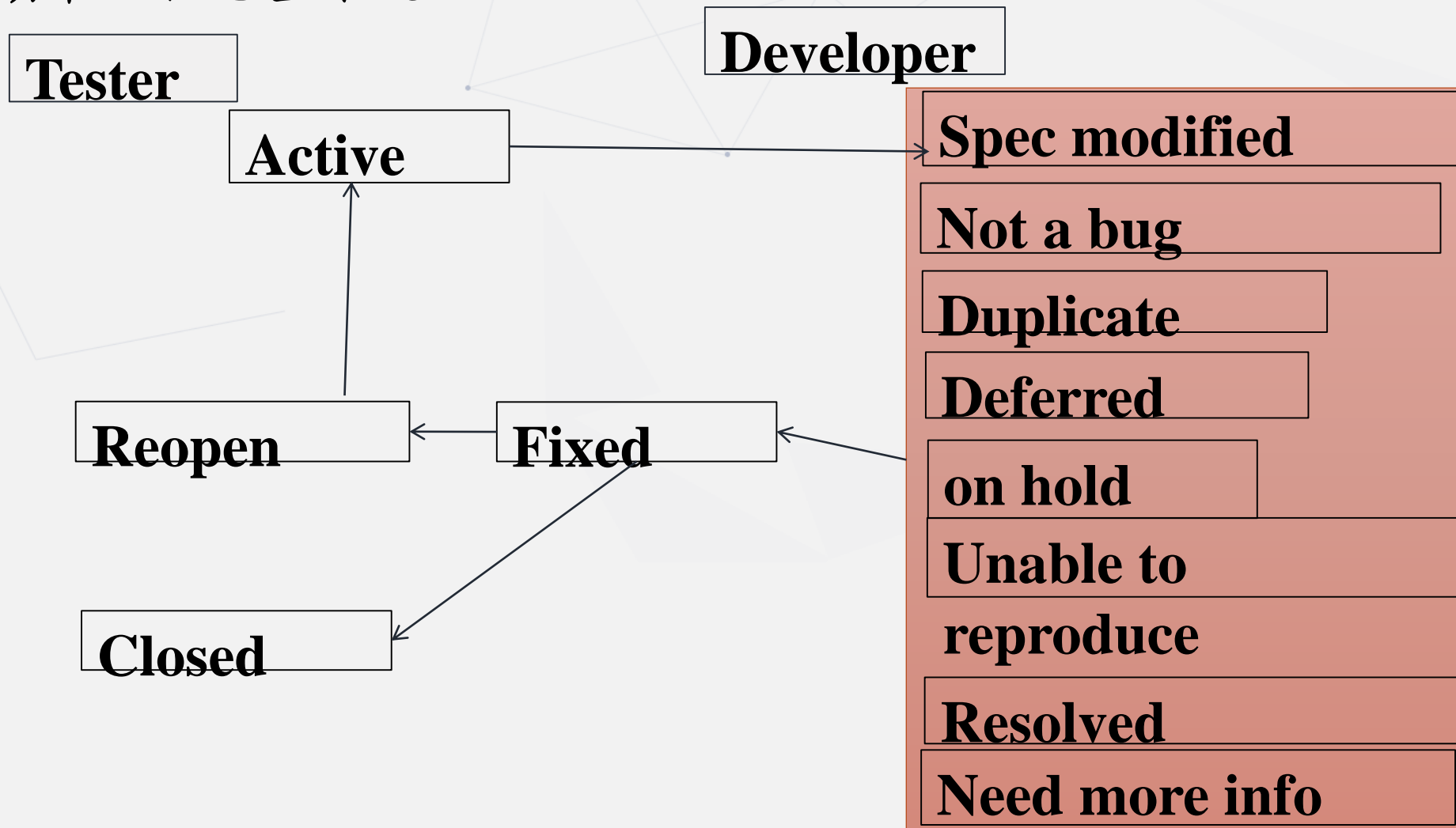
缺陷状态及周期性

6

缺陷工具的使用

# 软件缺陷——状态

- 缺陷状态：缺陷通过一个跟踪修复过程的进展情况，也就是在缺陷生命周期中的状态基本定义



# 软件缺陷--状态

重要! 理解! 灵活应用!  
不同bug管理工具中存  
在差异!

| 缺陷状态                        | 描述                            |
|-----------------------------|-------------------------------|
| 激活或打开 (Active or Open)      | 新提的缺陷，确认“提交的缺陷”，问题还没有解决，等待处理。 |
| 已修正 (Fixed or Resolved)     | 已被开发人员检查、修复过的缺陷，已解决但还未被测试人员验证 |
| 关闭或非激活 (Closed or Inactive) | 测试人员验证后，确认缺陷不存在之后的状态。         |
| 重新打开 (Reopen)               | 测试人员验证后，还依然存在的缺陷，等待开发人员进一步修复  |

| 缺陷状态                        | 描述   |
|-----------------------------|--|
| 推迟 (Deferred)               | 这个软件缺陷可以在下一个版本中解决                                |
| 保留 (on hold)                | 由于技术原因或第三者软件的缺陷，开发人员不能修复的缺陷                      |
| 不能重现 (Unable to reproduce)  | 开发不能复现这个软件缺陷，需要测试人员检查缺陷复现的步骤。                    |
| 需要更多信息 (Need more info)     | 开发能复现这个软件缺陷，但开发人员需要一些信息，例如：缺陷的日志文件，图片等。          |
| 重复 (Duplicate)              | 这个软件缺陷已经被其他的软件测试人员发现。                            |
| 不是缺陷 (Not a bug)            | 这个问题不是软件缺陷                                       |
| 需要修改软件规格说明书 (Spec modified) | 由于软件规格说明书对软件设计的要求，软件开发人员无法修复这个软件缺陷，必须要修改软件规格说明书。 |

# 软件缺陷---生命周期

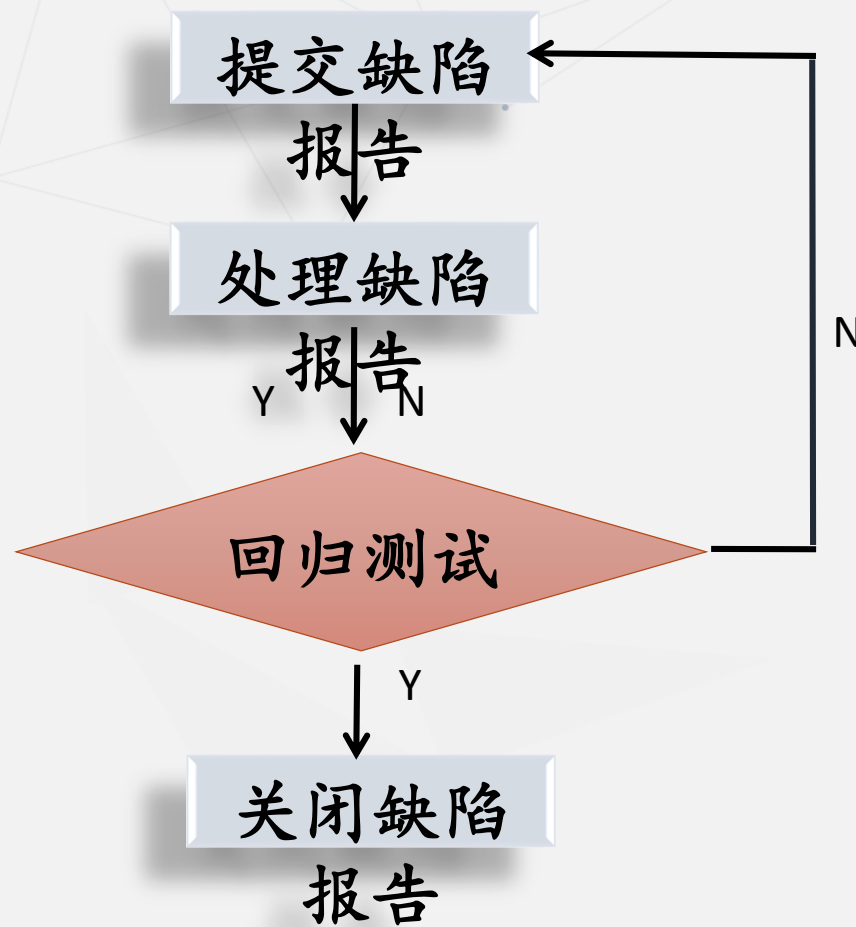


- 测试人员

- 开发人员

- 测试人员

- 测试人员





1

什么是缺陷

2

缺陷产生的原因及修复的成本

3

缺陷报告的编写

4

缺陷的严重性和优先级

5

缺陷状态及周期性

6

缺陷工具的使用



# Bug工具使用



提Bug



所属产品

雪梨学院

所属模块

/

所属项目

影响版本



当前指派

Bug类型

代码错误

操作系统

浏览器

Bug标题

严重程度

3

优先级

?



重现步骤

[步骤]

[结果]

[期望]

相关需求

相关任务

抄送给

选择要发信通知的用户...



关键词

附件(5M)

选择文件

未选择任何文件

标题:

+

x

保存

返回



# Question