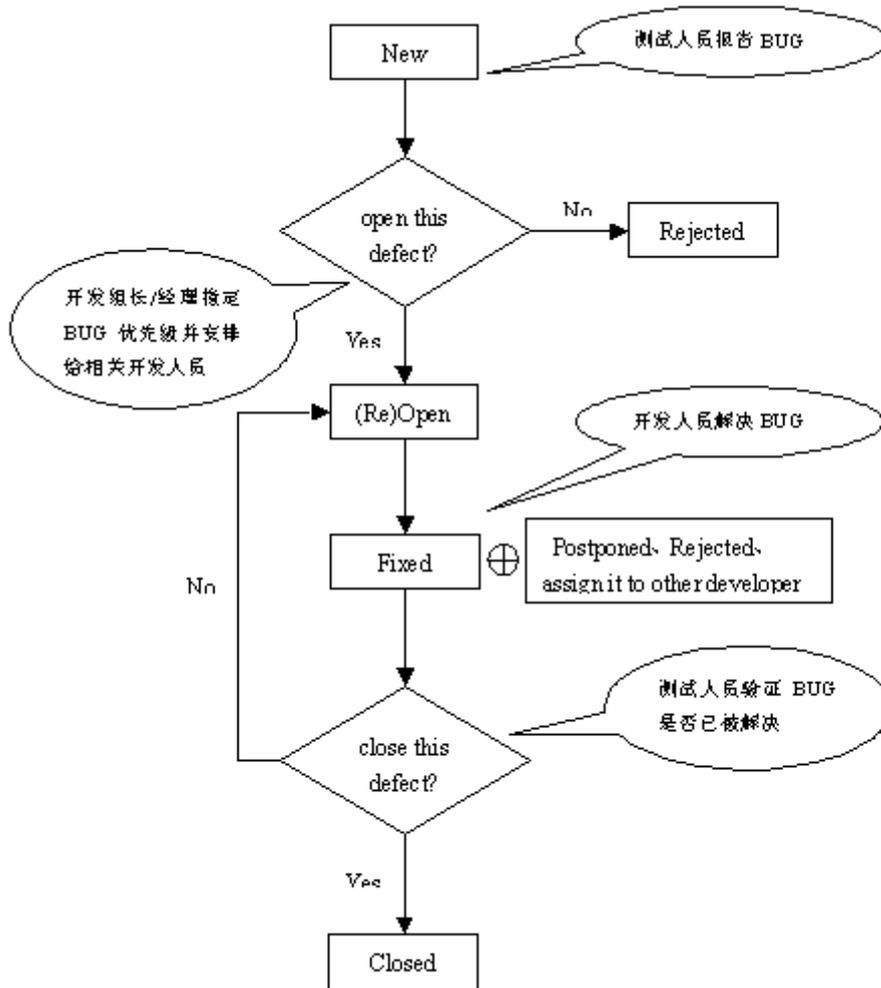


## Bug 状态流程图



## 对 Bug 的处理

### 开发组长/经理

每天对 Bug 进行分配，标注处理意见，给定优先级（发版前必须三方：需求、开发、产品共同确定）。问题分配时，应尽可能将咨询类、理解错误类等问题处理掉，而不是留给开发人员。有可能是需求的问题，分配给需求人员。定期对 Bug 库分析，找出常出错的模块，进行代码审查

### 开发人员

分析 Bug，写出问题原因，修改 Bug；实行 Bug 优先原则，严重程度 B-Major 类或紧急程度 3-High 类以上（包含）bug5 个或 5 个以上，停止新功能的开发。

### 需求人员

解释需求，给出处理意见，将 Bug 库中的建议整理成需求文档。评审确定后列入开发计划

### 测试人员

不参与问题的优先级的定位，只用 Bug 级别反映 Bug 的严重程度。验证 Bug 是否已被解决

### 测试组长/经理

审核测试人员提交的 Bug。定期对 Bug 库进行分析，描绘出曲线图等，报告现状、预测趋势。在测试总结报告中给出意见

### 产品人员

可以对优先级和处理意见等进行审核，如果有意见，和项目组商量定夺

**Bug 状态(Status)**：指缺陷通过一个跟踪修复过程的进展情况。包括 New、Open、Reopen、Fixed、Closed 及 Rejected 等

New	为测试人员新问题提交所标志的状态。
Open	为任务分配人（开发组长/经理）对该问题准备进行修改并对该问题分配修改人员所标志的状态。Bug 解决中的状态，由任务分配人改变。对没有进入此状态的 Bug，程序员不用管。
Reopen	为测试人员对修改问题进行验证后没有通过所标志的状态；或者已经修改正确的问题，又重新出现错误。由测试人员改变。
Fixed	为开发人员修改问题后所标志的状态，修改后还未测试。
Closed	为测试人员对修改问题进行验证后通过所标志的状态。由测试人员改变。
Rejected	开发人员认为不是 Bug、描述不清、重复、不能复现、不采纳所提意见建议、或虽然是个错误但还没到非改不可的地步故可忽略不计、或者测试人员提错，从而拒绝的问题。由 Bug 分配人或者开发人员来设置。

**Bug 严重级别 (Severity, Bug 级别)**：是指因缺陷引起的故障对软件产品的影响程度。由测试人员指定。

A-Crash	错误导致了死机、产品失败（“崩溃”）、系统悬挂无法操作；
B-Major	功能未实现或导致一个特性不能运行并且不可能有替代方案；
C-Minor	错误导致了一个特性不能运行但可有一个替代方案；
D-Trivial	错误是表面化或微小的（提示信息不太准确友好、错别字、UI 布局或罕见故障等），对功能几乎没有影响，产品及属性仍可使用；
E-Nice to Have (建议)	建设性的意见或建议。

**Bug 优先级(Priority)**：指缺陷必须被修复的紧急程度。由 Bug 分配者（开发组长/经理）指定。

5-Urgent	阻止相关开发人员的进一步开发活动，立即进行修复工作；阻止与此密切相关功能的进一步测试
----------	--

4-Very High	必须修改，发版前必须修正
3-High	必须修改，不一定马上修改，但需确定在某个特定里程碑结束前须修正
2-Medium	如果时间允许应该修改
1-Low	允许不修改

**功能模块(Subject):** TD 中需在 Test Plan 页中定义好 Subject，才能在 Defects 页中使用。

**问题描述、附件附图** 请参见后面第四部分 ‘[Bug 描述要求](#)’ 的有关内容。

**处理意见:**开发组长/经理(或具体 Bug 分配人员) 在审核新 Bug 时、将 Bug 分配给开发人员解决前，需要给出该 Bug 的处理意见。

Fixable	可修改。表示 Bug 可以被修复或更正
Duplicated	重复。表示该 Bug 已经被其它测试人员找出来了（‘纯粹’重复），或者开发认为原因是相同的（但从测试来看，认为出现的地方有所不同、表现有所不同等）
Postponed	延后。由于时间、进度、重要程度或者技术/需求等方面的原因，认为不能解决、须延期解决、或者本版不做留待到后续版本解决的 Bug。 （注：因 ‘Bug 状态’ 字段中也有该值，根据各组各自使用情况，可以只保留一个，或者开发/测试各有侧重地使用这两个 Postponed）
By Design	因设计结构问题无法修改。测试人员认为是 Bug，不符合逻辑，也不符合用户的要求，但开发人员则认为是按照设计做的、只能如此处理，否则修改代价太大
Can' t Reproduce	不可复现。不能重现（如因 Bug 出现的环境重现不了了），或以前出现的某个 Bug 自动消失了（可能是在处理其他 Bug 的时候把这个 Bug 一并修复掉了）。 （注：因 TD 本身亦带有 ‘是否复现(Reproducible)’ 字段，根据各组各自使用情况，可以用它来标识，或者不用它而在 ‘处理意见’ 字段中用该值标识出）
Disagree With Suggestion	不同意所提意见或建议，不采纳
Not Error	不是问题。测试人员提错了
Won' t Fix	这个 Bug 是一个错误，但还没有重要到非要更正不可的地步，可以忽略不计

说明：

1. 定为 Duplicated 的 Bug，必须注明和 XXXbug 重复
2. 测试人员对标明为 Duplicated 的 Bug 复测，需要 XXXBug 修改后方可进行
3. 定期回顾 Can' t Reproduce, Postponed
4. 定期整理 By Design

其它一些字段（及所定义的枚举值）的定义解释，供有需要用到的组参考：

**测试状态 (TestState) :** 新提交的 Bug 定位标准。由测试人员指定。一般有 8 个（提交 Bug 时给出）

1—New Defects (或写成 Defect)	新 Bug
2—Second Defects (或写成 SB)	复测时新出现的 Bug
3—Faculative	偶发性
4—Reappear	原来修改过的问题又重新出现
5—By Requirement	需求要求但没有做的功能
6—Suggestion	需求需要完善
7—Differ With Requirement	与需求不一致
8—By Design	设计要求但没有做的功能

**复测状态(ReTestState):** 复测时给出的状态, 测试人员对于经过验证的 Bug 应按以下几种标准进行定位。由测试人员指定。一般有 1—OK、2—PD、3—DV、4—NB、5—NR、6—AR。

OK	正确
PD	此问题悬而不决
DV	有错误可以暂时不考虑
NB	不是错误
NR	不能复现的错误
AR	需求不明确

**问题定位:**

Calculate_error	计算错误, 指计算过程中、计算结果错误。
Data_error	数据错误, 指非计算结果类的的数据错误。
Graphics_error	图形错误, 指绘图、图形显示、图形编辑时发生的错误。
Interface_error	界面错误
Requirement_error	需求错误
Function_error	功能错误
Unknown_error	未知错误

**缺陷来源(Source):** 指引起缺陷的起因。

Requirement	由于需求的问题引起的缺陷
Architecture	由于构架的问题引起的缺陷
Design	由于设计的问题引起的缺陷
Code	由于编码的问题引起的缺陷
Test	由于测试的问题引起的缺陷
Integration	由于集成的问题引起的缺陷

**类型(Type):** 是根据缺陷的自然属性划分的缺陷种类。

F- Function	影响了重要的特性、用户界面、产品接口、硬件结构接口和全局数据结构。并且设计文档需要正式的变更。如逻辑，指针，循环，递归，功能等缺陷
A- Assignment	需要修改少量代码，如初始化或控制块。如声明、重复命名，范围、限定等缺陷
I- Interface	与其他组件、模块或设备驱动程序、调用参数、控制块或参数列表相互影响的缺陷。
C- Checking	提示的错误信息，不适当的数据验证等缺陷。
B- Build/package/merge	由于配置库、变更管理或版本控制引起的错误
D- Documentation	影响发布和维护，包括注释。
G- Algorithm	算法错误。
U- User Interface	人机交互特性：屏幕格式，确认用户输入，功能有效性，页面排版等方面的缺陷
P- Performance	不满足系统可测量的属性值，如：执行时间，事务处理速率等。
N- Norms	不符合各种标准的要求，如编码标准、设计符号等。

(以上依各组实际情况可以作适当调整)

## 项目组各角色在 Bug 库中的权限

**管理员：**全部权限

**测试组长/经理：**全部权限

**测试人员：**可添加 Bug、不能删除 Bug、可添加注释评论 (R&D Comments)、不可修改他人所提 Bug、可调整：Bug 概要 (题目, Summary)、问题描述、附件附图 (Attachments)、Bug 状态、Bug 级别、测试版本、测试产品、功能模块、测试状态、问题定位、复测状态、注释评论 (R&D Comments)、复测人、复测日期、修改人

**开发人员/需求人员：**不能删除 Bug、可添加注释评论 (R&D Comments)、可调整：注释评论 (R&D Comments)、是否复现、Bug 状态 (不过无法直接标为 closed)、问题描述、处理意见、待测版本、修改人、修改日期。可添加 Bug。

**开发组长/经理/需求经理：**除了开发人员的权限，还可调整：优先级别、责任人、Bug 概要 (题目, Summary)、附件附图 (Attachments)

**项目经理：**可添加 Bug、可添加注释评论 (R&D Comments)、可修改字段：Bug 概要 (题目, Summary)、问题描述、附件附图 (Attachments)、Bug 状态 (不过无法直接标为 closed)、修改人、优先级别、问题定位、处理意见、注释评论 (R&D Comments)、是否复现、责任人、待测版本。也可删除 Bug，但要与测试组长/经理协商。

不属于项目组成员的其他人如研发中心经理组成员等，有必要查看 TD 库的话，可分配给其帐号及查看的权限。

## Bug 描述要求

Bug 描述的要求为分类准确、叙述简洁、步骤清楚、有实例、易再现、复杂问题有据可查（截图或其它形式的附件）。测试组长/经理把关，以开发人员的角度来审查 Bug 描述，看其是否描述清楚了 Bug，不好描述的把工程文件或截图作为附件提交。具体要求为：

- 问题描述一般格式：问题描述时，建议分几步描述：模块或功能点=>测试步骤=>期望结果=>实际结果=>其它信息，可依实际情况调整；
- 单一：尽量一个报告只针对一个软件缺陷，报告形式应方便阅读。在主报告之后应注明不同的条件；
- 简洁：每个步骤的描述应尽可能简洁明了。只解释事实、演示和描述软件缺陷必要的细节，不要写无关信息；
- 再现：问题必须在自己机器上能复现方可入库（个别严重问题复现不了也可入库，但需标明）；
- 复杂的问题应附截图补充说明或直接通知指定的修改人；考虑到网络数据传输效率，截图的文件格式建议用 JPG 或 GIF，不建议用 BMP；抓图可用 TestDirector 自带的功能，亦可用 HyperSnap 之类的专用抓图工具。
- 报告中不允许使用抽象词句：比如“有错误”之类；
- 有关操作系统特征问题：应在不同操作系统上进行操作，看是否能重现，并在 Bug 报告中标识；
- Bug 描述示例：

<p>例一 河北 98 土建标准换算 <b>操作：</b> 1. 输入 9-24 2. F8 3. 在 F8 输入 10 <b>期望结果：</b>进行换算 <b>实际结果：</b>提示“输入的厚度应大于 20”</p>	<p>例二（模块或功能点也可在‘功能模块’字段中规定，则 Bug 描述中就不必写了） <b>操作：</b> 1. 打开新建向导； 2. 在“新建”中的“项目名称”中输入&gt;80 个字符； 3. 点击“下一步” <b>期望结果：</b>“项目名称”应≤80 个字符，输入大于 80 个字符，点击“下一步”应有错误提示 <b>实际结果：</b>进入“比重调整”界面</p>	<p>例三（程序员知道期望结果的情况下） 云南 98 土建 <b>操作：</b> 1. 输入 13-170 2. F5 3. 在 F5 中修改 3240008 的名称，处于编辑状态 4. 到人材机，再回来 <b>实际结果：</b> F5 中变白板 <b>注：</b>若 3 不处于编辑态切换则正常</p>	<p>例四（建议、需求类） <b>功能：</b>预算页，子目排序后可恢复原顺序 <b>用途：</b>用户误操作后可复原</p>
---	--	--	---

注：所有项目采用 TestDirector 进行 Bug 管理，该工具能从测试步骤自动生成 Bug 报告，因此对于 Bug 描述要求在测试方案用例设计（在 Test Plan 页中）阶段就可以进行控制。

附：好的 Bug 报告应满足以下几方面的要求：

- 结构清晰
- 复现故障再写报告
- 隔离 Bug：更改条件复测
- 归纳：是否其他模块也有相同的 Bug
- 比较：其他测试用例是否使用到此 Bug
- 总结：报告的开头有 Bug 的总结
- 精简：不要有多余的步骤和语言
- 无歧义：语言明确
- 中立：无批评性语言
- 讨论：将要发出的报告送其他测试人员讨论

## 小结

- 通过专业的技术测试出精确的 Bug；
- 通过准确的文档报告 Bug；
- 通过良好的沟通使 Bug 尽快解决。