

系统测试缺陷跟踪管理细则

编写： 总行信息技术部测试部

日期： 2008 年 10 月

审核：

日期：

批准：

日期：

章节目录

1. 总则	1
2. 适用范围	1
3. 术语表	1
4. 角色定义	1
5. 参考资料	2
6. 缺陷跟踪与管理相关角色职责.....	2
6.1 管理人员职责	2
6.1.1 项目经理.....	2
6.1.2 技术经理.....	2
6.1.3 测试负责人.....	2
6.1.4 版本管理员	3
6.1.5 QA	3
6.2 需求人员职责	3
6.3 开发人员职责	4
6.3.1 分析设计人员	4
6.3.2 软件工程师	4
6.4 测试工程师职责	4
6.4.1 测试设计人员	4
6.4.2 测试执行人员	4
7. 缺陷处理流程及说明.....	6
7.1. 缺陷处理流程图	6
7.2. 完整缺陷状态流转图	8
7.3. 缺陷状态说明	8
7.4. 缺陷严重性级别说明	9
7.5. 缺陷类型说明	11
8. 缺陷处理常见问题（FAQ）.....	14
8.1. 管理人员 FAQ.....	14
8.2. 开发人员 FAQ.....	16
8.3. 测试工程师 FAQ.....	17

1. 总则

缺陷（Defect）管理是测试过程中非常重要的一个环节。为了提高测试效率和测试质量，规范所有缺陷管理流程，特制定本细则。

2. 适用范围

研发中心内所有需要经过测试中心进行系统测试的项目，仅针对系统测试过程中发现的被测系统相关缺陷的跟踪管理，不包括其他测试阶段缺陷的跟踪管理，也不包括对系统测试需求和系统测试案例评审过程中发现的缺陷。

3. 术语表

- **软件缺陷（Bug/Defect）**：软件缺陷指的是软件中存在的瑕疵，可能导致系统失效，或者出现错误的结果或表现，是一种对软件预期属性的偏离现象。一般来说，可能符合以下情况：
 - 软件未达到需求规格说明书标明的功能
 - 软件出现了需求规格说明书指明不会出现的错误
 - 软件功能超出需求规格说明书的范围
 - 软件未达到需求规格说明书未指明但应达到的目标（比如应符合法律法规要求、监管要求或常识）
 - 软件测试员认为软件难以理解、不易使用、运行速度缓慢，或者从测试人员的角度看，最终用户会认为不好
- **系统测试**：系统测试简称为 ST，是 System Testing 的缩写，指的是站在用户的角度上对系统作功能性的验证，同时还对系统进行一些非功能性的验证，包括性能测试、压力测试、容量测试、安全测试、恢复性测试等。

4. 角色定义

- **管理人员**：主要指组织管理协调保证项目有序进行的人员，包括项目经理、技术经理、测试负责人、版本管理员、QA等。
- **需求人员**：需求提供方，包括行内需求和行外需求提供方。一般行内需求方多为业务部门（业务负责人或业务代表）或提出技术改进需求的开发人员。

- **开发人员**：主要指在系统上实现需求的人员，包括分析设计人员、软件工程师。
- **测试人员**：主要指在开发过程对系统是否满足需求而进行检测的质量控制人员，包括测试设计人员、测试执行人员、环境支持人员。

5. 参考资料

《测试管理平台操作手册》

6. 缺陷跟踪与管理相关角色职责

6.1 管理人员职责

6.1.1 项目经理

- 组织缺陷管理规范的讨论，制定并督促适合本项目的缺陷管理规范。
- 定期召集缺陷分析会，根据分析结果调整项目计划并督促执行。
- 组织有争议的缺陷决策会。

6.1.2 技术经理

- 参与讨论本项目的缺陷管理规范。
- 跟踪设计缺陷，及时修复并评估影响范围，提供给开发、测试双方进行相应的调整。
- 保证设计文档的准确及时更新。
- 提供准确的版本发布说明，列出本版本新增功能和修复的缺陷清单。
- 参与有争议缺陷的决策讨论。

6.1.3 测试负责人

- 在项目启动时，与项目经理、技术经理一起制定符合项目特点的缺陷管理规范（包括版本和轮次管理办法、缺陷描述的标准、缺陷跟踪管理方法、争议缺陷解决方法等，必须与部门的管理方法不冲突），并在实际工作中进行微调，保证全体项目成员理解的一致性。
- 及时督促开发人员修正缺陷（包括分析设计人员和软件工程师）
- 组织测试设计和测试执行，客观及时的将系统的缺陷反馈给各方。
- 定期跟踪缺陷，提供缺陷的统计分析数据，向项目经理提出召开缺陷分析会的建议和申请。建议测试负责人每周向项目成员发送缺陷统计分析报告，主要包含当前未关闭（非问题关闭）的缺陷数量、

缺陷关闭趋势，重点标注超期（如已提交超过 1 周的）的打开（重新打开）、已分配、待讨论缺陷，同时可以对缺陷的分布情况做一定分析。对缺陷较多的项目或涉及多个开发团队的项目群，建议每天或隔天召开缺陷分析会议，直到缺陷收敛。

- 检查缺陷数据，确定缺陷严重程度、缺陷类型选择是否正确，检查缺陷信息填写是否规范、正确，需要关注缺陷类型为“其他”的缺陷，只有确实不能归类到现有明确缺陷类型定义的，才能归入“其他”类型。
- 参与有争议缺陷的决策讨论。
- 督导测试设计、测试执行中关于缺陷的提交、跟踪、处理规范落实情况。
- 根据缺陷情况组织测试设计人员、测试执行人员进行案例调整和重新执行。
- 在编写测试计划时规划好测试环境，并在测试执行过程中积极协调解决测试环境相关问题。

6.1.4 版本管理员

- 负责需求文档版本管理
- 负责设计文档版本管理
- 负责代码版本管理
- 负责参数配置版本管理
- 负责测试文档版本管理
- 及时跟踪和解决版本的缺陷。

6.1.5 QA

- 关注缺陷的提交、分析、跟踪、处理过程的规范性

6.2 需求人员职责

- 提供清晰准确的需求文档，并根据项目情况及时更新
- 参与有争议缺陷决策的讨论会
- 及时提供业务参数配置文档
- 最后一轮 ST 测试执行前，与测试工程师、开发人员一起，梳理所有的遗留缺陷，包括项目外缺陷，明确转需求处理的方式，如果是转业务需求，明确对应的业务部门；如果是转内部需求，明确对应的开发负责人

6.3 开发人员职责

6.3.1 分析设计人员

- 及时跟踪和分析设计缺陷，并根据方案及时更新提交设计文档，并督促编码人员按照最新的设计文档更新代码。
- 参与有争议缺陷决策讨论会。
- 对有争议的问题提交技术经理进行分析讨论。
- 对编码人员进行指导，对编码错误进行跟踪和协助解决。

6.3.2 软件工程师

- 及时处理缺陷。严重性级别为“1-致命”的需要在 8 小时内做出回复；严重性级别为“2-严重”的需要在 48 小时内做出回复；严重性级别为“3-一般”或“4-较小”的需要在 72 小时内做出回复。
- 对于已修复的缺陷，需要在缺陷记录中简要说明修复方法。
- 对于否决或推迟处理的缺陷，需要在缺陷记录中说明原因。如否决的缺陷无说明原因，测试工程师将直接重新打开。
- 最后一轮 ST 测试执行前，与测试工程师、需求人员一起，梳理所有的遗留缺陷，包括项目外缺陷，明确转需求处理的方式，如果是转业务需求，明确对应的业务部门；如果是转内部需求，明确对应的开发负责人

6.4 测试工程师职责

6.4.1 测试设计人员

- 根据业务需求或设计文档，及时更新测试案例。
- 指导和跟踪测试执行人员的执行质量，并规范提交缺陷。
- 参与有争议缺陷决策讨论。
- 向测试负责人反馈缺陷情况并对缺陷进行分析和跟踪。

6.4.2 测试执行人员

- 缺陷报告中说明了对应的测试用例编号。
- 缺陷报告标题简短、准确，提供缺陷的本质信息。
- 报告的缺陷是已知缺陷列表中没有的。
- 原则上，每条缺陷报告只针对一个软件缺陷。如果能确定多个问题都是一个原因导致的，也可以只提一个缺陷，此时必须通知测试设计人员，并在修复后的验证过程中，验证所有相关测试案例。

- 在缺陷报告中准确记录版本信息和轮次信息。
- 在缺陷报告中准确记录错误消息、错误代码。
- 缺陷报告中详细描述了错误现象、可以重现的步骤。
- 对于无法重现的缺陷，需要在“是否可重现”字段中予以说明选择正确的选项，并在缺陷描述中说明问题的偶然性，并详细描述当时出现缺陷的场景、环境。
- 缺陷报告中为了更好描述错误，必须附加截屏图（无界面的特殊业务除外，如电话银行业务）。
- 对于软件工程师已修复并已提交版本到测试环境的缺陷，及时进行验证。
- 对于提出的缺陷，需要不断跟踪，直到缺陷关闭或处于其他终态为止。（缺陷终态包括“非问题关闭”、“关闭”、“推迟处理”、“转业务需求”和“转内部需求”五种，其中“推迟处理”既可以是中间状态，也可以是终态，需要持续进行跟踪。）
- 在测试中发现的性能问题，记录缺陷，并通知技术测试室。
- 对于重新打开的缺陷，需要说明重新打开的原因。
- 最后一轮 ST 测试执行前，项目组内梳理遗留缺陷后，对于明确了需要转化为业务需求或者内部需求的缺陷，测试工程师修改其状态为“转业务需求”或者“转内部需求”。
- 测试结束后，所有的缺陷都必须是“已关闭”、“非问题关闭”、“推迟处理”、“转业务需求”、“转内部需求”这五种状态中的一种。

7. 缺陷处理流程及说明

7.1. 缺陷处理流程图

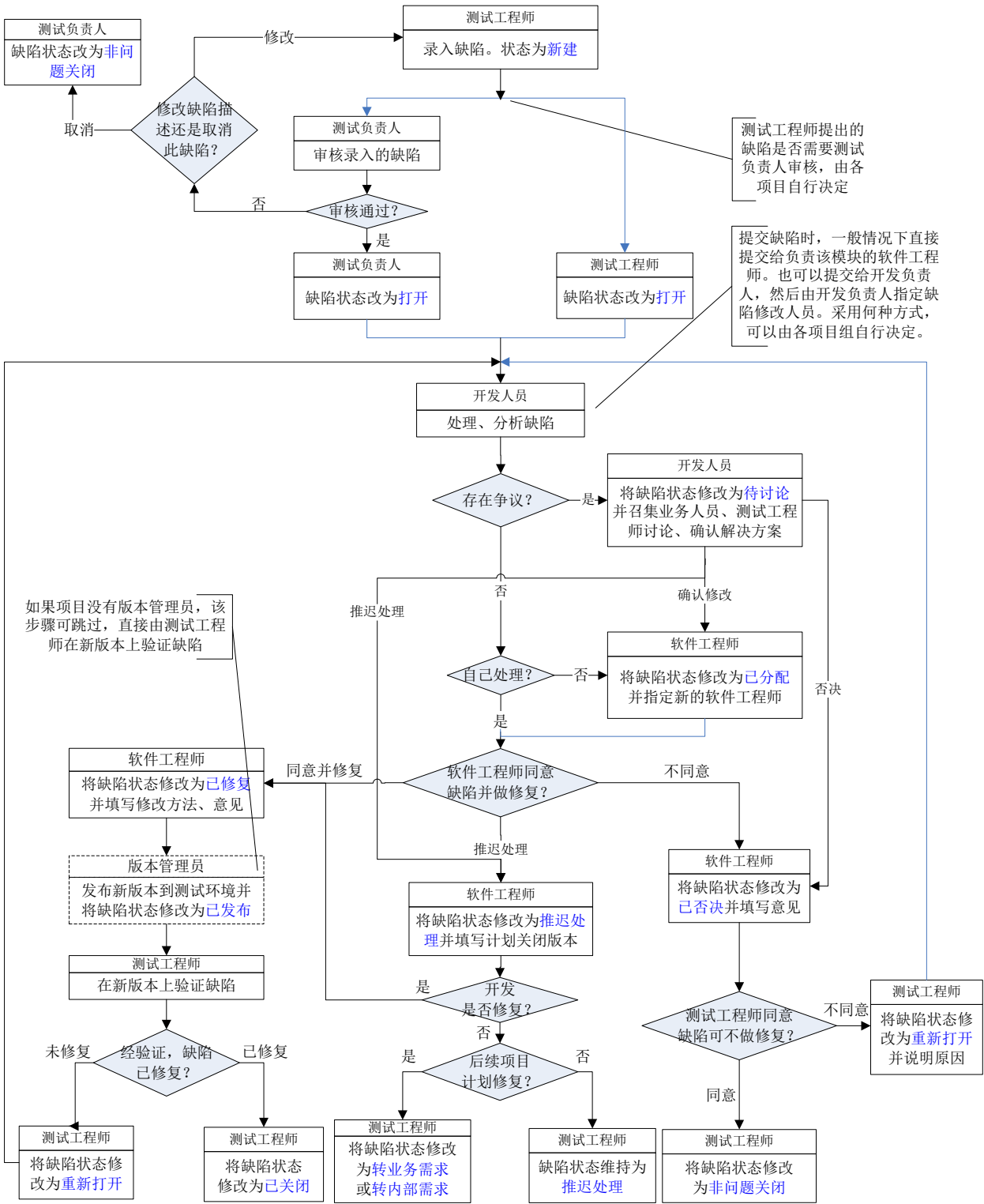
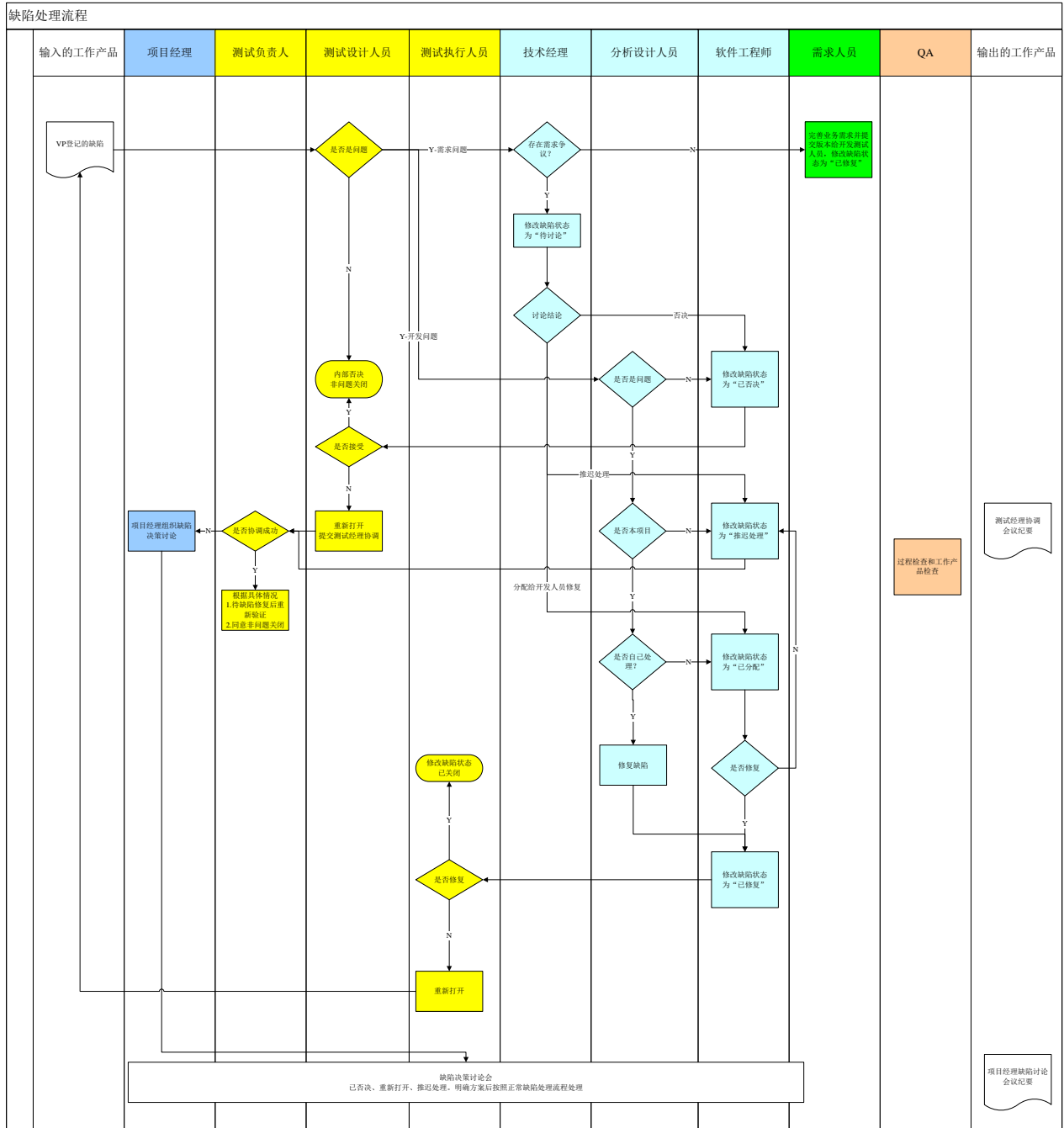


图 7-1 缺陷状态在不同环节的变化



7.2. 完整缺陷状态流转图

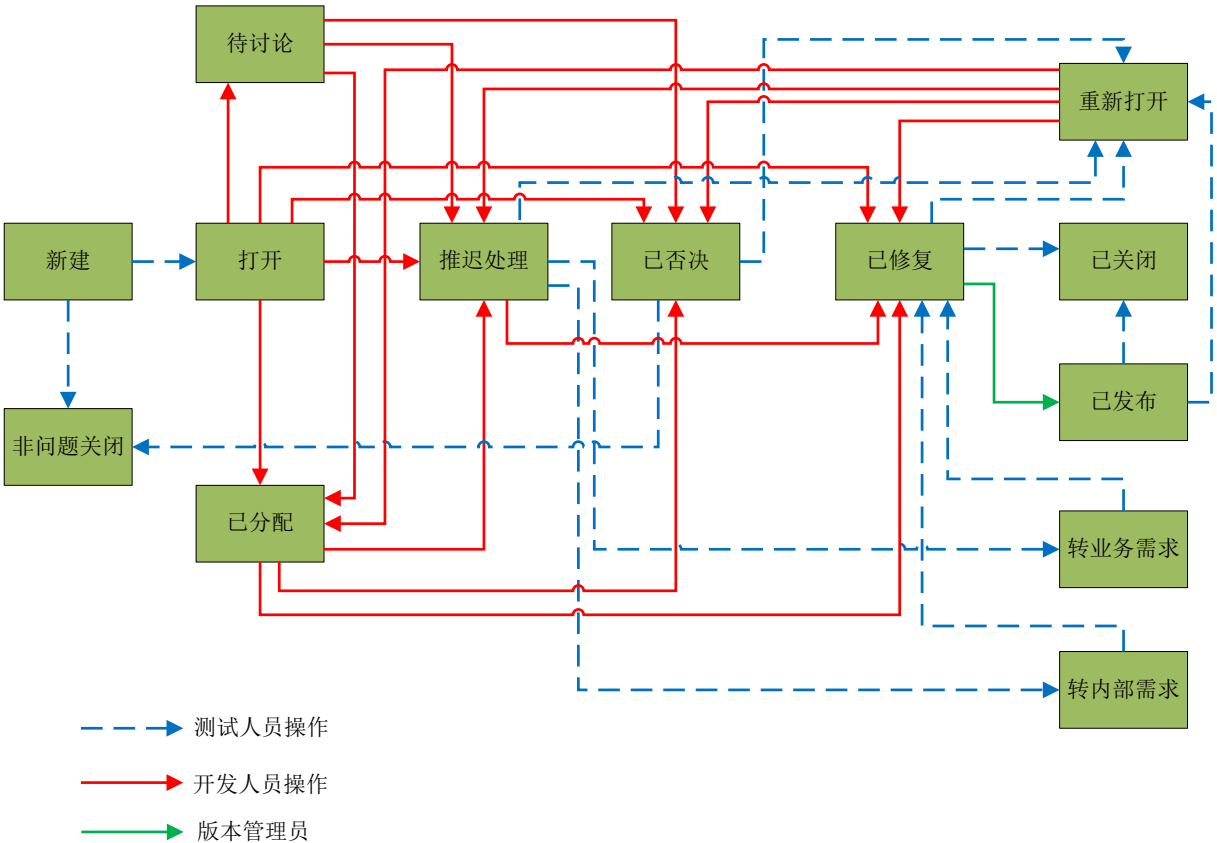


图 7-2 缺陷状态流转图

7.3. 缺陷状态说明

状态名称	说 明
新建（New）	测试中新建的软件缺陷，还没有提交给软件工程师。一般由测试负责人进行评估，如果属于缺陷，修改为“打开”状态；如果不属于缺陷，修改为“非问题关闭”状态。如果录入的缺陷不符合 6.4.2 节中的规则，也需要进行修改。
打开（Open）	测试中新建的软件缺陷，已提交给相关软件工程师。
待讨论（TBD）	存在争议，需要软件工程师、需求人员和测试工程师共同确认；该状态是临时状态，在进入最后一轮 ST 测试执行之前，必须进行处理。
已修复（Fixed）	软件工程师已修正缺陷，等待测试工程师进行验证。
已发布（Released）	软件工程师已提交修复版本，并且版本管理员已将修复版本发布到测试环境。
推迟处理（Postponed）	由于各种原因，提交的缺陷需要在后续版本才能修改，或者不修改。推迟处理的缺陷需要由需求人员、软件工程师和测试工程师最终确认。对准备修改的缺陷，在进入最后一轮 ST 测试执行之前，测试工程师就必须和需求人员、开发人员讨论确定是“转业务需求”还是“转内部需求”，在后续项目进行修复，或者

	在本项目最后一轮 ST 测试执行前修复。对一致认可但不准备修复的缺陷，或者不能重现的缺陷，可以将“推迟处理”作为终态。
已否决（Rejected）	软件工程师认为不需要修改，或者按照设计就应该是这样的。对于非本项目或者非本人缺陷，由于设计变更或者版本变更之前的问题现在无法重现，不能否决。无理由否决的，测试工程师一律重新打开。
已分配（Assigned）	转发给其他人处理。也可以转给自己处理，转给自己处理时，表明软件工程师已接受此项缺陷。
重新打开（Reopen）	测试工程师对已修复的缺陷进行验证，发现缺陷仍旧存在。或者测试工程师认为已被否决的缺陷确实需要修改。
已关闭（Closed）	缺陷已被修复，且回归测试验证成功。
非问题关闭	对于软件工程师否决的缺陷，如果测试工程师认为确实不属于缺陷，则将缺陷状态标记为“非问题关闭”。
转业务需求	该缺陷与业务相关，需要解决，但在本项目中不处理，由需求人员在后续版本中作为业务需求提出。
转内部需求	该缺陷主要是设计实现的问题，不涉及业务需求的变更，需要解决，但在本项目中不处理，由软件工程师在后续版本中作为内部改进需求提出。

表 7-1 缺陷状态说明

7.4. 缺陷严重性级别说明

严重性级别	定义	示例
1-致命	<ol style="list-style-type: none"> 1. 由于程序所引起的死机,非法退出或系统挂起 2. 死循环 3. 导致数据库发生死锁 4. 数据转换错误,通讯程序不稳定或通讯程序错误 5. 主要功能缺失 6. 记账错误 7. 严重的数值错误,比如数据库或回单上的客户信息,余额,利息,积数,日期等 	<ol style="list-style-type: none"> 1. 套录管理修改版本号到 99 后,再修改版本,变量溢出,程序崩溃 2. GMS 早期版本在多个用户并行操作时会导致数据库死锁 3. 双机事务保护程序在多包模式下,组包位置计算时发生重叠,导致通讯报文中丢失部分数据 4. 凌晨 3:00 滚积数失败,但要到当日晚上日终清算时才会报错。当日日间销户时的利息就会发生错误,没有包含上日积数 5. 回单上的金额与数据库中不一致 6. 重复记账或出现单边表内账
2-严重	<ol style="list-style-type: none"> 1. 影响业务进行的主要功能与需求不符。包括设计与需求不符,以及实现与设计不符 2. 非主要功能未实现 	<ol style="list-style-type: none"> 1. 一卡通开通周转易后,发生银联预授权交易,先使用周转易的额度,而不是预期的一卡通活期账户余额 2. 内部账户关户后打印查询功能没有

	<ul style="list-style-type: none"> 3. 程序引起的接口或数据通讯错误, 导致数据缺失, 信息传输错误等 4. 导致资金或是账务错误等关键字段的数值计算错误 5. 操作权限错误 6. 异常处理遗漏或错误 7. 系统处理速度很慢 8. 显示或打印的关键信息格式与需求定义存在大的偏差 	<p>实现</p> <ul style="list-style-type: none"> 3. 测试时发现前后台接口不匹配, 导致程序调用错误 4. 登记旧代理行, 如果报错, 则登记 FTS 工作流任务, 此时新代理行已经记了旧提出贷的账务, 但旧提出贷报错后, 新代理行没有回滚旧提出代理行账务的数据库操作 5. 授权操作未实现 6. 3G 的核算种类与科目对应关系界面查询功能性能过低, 耗时约 10 秒 7. 综合查询时明细部分金额栏位整数位数只有 6 位, 大金额时显示错误
3-一般	<ul style="list-style-type: none"> 1. 与设计文档不符的界面错误 2. 非主要功能实现不正确, 但不影响业务进行 3. 非关键信息的计算错误, 如页码的计算错误 4. 打印的非关键信息错误; 由于内容超长导致的格式错误 5. 简单的输入限制未放在前台进行控制 6. 业务操作缺少交互式的确认提示信息 7. 错误提示信息错误 	<ul style="list-style-type: none"> 1. 科目管理_科目界面字段与需求设计字段名称不一致 2. 机构日初, 没有修改 03 机日结标志为*LCK 的柜员的运行状态 3. 内部账户开户币种为空系统无控制 4. 修改内部资金号_账户主机为 03 机的内部资金号修改后报资金节点不能为空, 但实际已修改成功 5. 增加重复交易定义与增加重复套录代码的报错提示信息内容位置反了
4-较小	<ul style="list-style-type: none"> 1. 辅助说明描述不清楚 2. 显示格式不规范, 界面有错别字 3. 长时间操作未给用户进度提示 4. 提示窗口文字未采用行业术语 5. 错误提示信息不清晰 6. 可输入区域和只读区域没有明显的区分标志 7. 界面布局不合理、用户体验效果不佳、操作界面与用户使用习惯不一致, 但不影响系统正常使用 8. 建议 	<ul style="list-style-type: none"> 1. 记账方式中“单笔计帐”的‘帐’字为错别字 2. 证件类别管理, 新增、修改成功, 没有操作成功的信息提示, 建议增加操作成功的相应信息提示。 3. 测试管理平台早期版本上传大量测试案例时没有进度条提示 4. 不录入证件号码, 错误提示内容不合适, 建议提示信息能够更明确, 如“必须输入证件号码”等

表 7-2 缺陷严重级别说明

7.5. 缺陷类型说明

缺陷类型分类方法（红色字体部分为缺陷类型）：

一级分类	二级分类	三级分类	四级分类	五级分类	六级分类
非重复错误	项目内缺陷	应用程序错误	功能错误	正常处理过程 错误	需求一致性错误
					界面错误
					通讯错误
					配置错误
					计算错误
					数据处理错误
					逻辑控制错误
					信息提示错误
					建议及警告
					性质不明错误
				异常处理过程 错误	
		性能错误			
		安全性错误			
	项目外缺陷	文档错误			
		版本错误			
		环境错误			
		测试中的错误			
重复错误					

缺陷类型说明如下：

缺陷类型	说 明
需求一致性错误	提供的需求描述错误 实现与提供的需求不一致
界面错误	控件布局、格式不统一 界面控制错误 界面布局与界面规范不一致 打印或输出格式错误 不符合常识或不符合操作习惯

	操作使用不友好，不易操作等
通讯错误	通讯接口不匹配 通讯接口数据转换错误 通讯程序不稳定 通讯程序错误
配置错误	操作系统或底层支持软件配置错误 应用系统参数配置错误 业务参数配置错误
计算错误	使用的计算方法或计算公式错误 数据来源错误导致的计算结果错误 重点是影响账务或可能导致法律纠纷的关键数据的计算错误，比如针对数据库和客户回单上账户余额、利息、积数、冻结金额、可用余额、额度等关键信息的计算错误
数据处理错误	变量初始数据错误 系统数据初始化错误 数组越界或缓冲区溢出 数据结构或数据库表结构设计有问题 写入数据库表的内容错误，比如出现重复记账或表内账单边账
业务逻辑控制错误	死循环 业务功能重复或缺失 业务功能实现与需求不符 流程控制不符合需求 流程实现不完整
信息提示错误	提示信息重复 提示信息内容错误 提示信息内容模糊，不能准确判断错误原因 提示信息出现时机不对，或焦点位置错误
建议及警告	功能类建议 流程类建议 操作类建议

	数据校验类建议 提示、说明等易用性建议 特定条件下的警告信息等
性能不明错误	不能判断出具体属于哪种错误类型的错误
异常处理错误	程序未进行异常处理 异常处理不正确或者不合理
性能错误	批量处理时间过长 联机业务响应时间过长 导致资源争用，造成互锁或死锁问题，如线程同步问题或数据库锁等待
安全性错误	用户和权限控制错误 有 SQL 注入漏洞 有跨站点攻击漏洞 有其他安全性漏洞
文档错误	需求文档中的描述错误 设计文档中的描述错误 其他文档（如安装手册、用户手册等）中的描述错误
版本错误	被测版本不正确
环境错误	通讯链路、中间件配置、数据库异常、操作系统及支撑软件故障 只有与业务系统无关的底层基础架构出错才能归入该类型的错误 在计算测试出口标准的时候，本类错误不需要计算在内
测试中的错误	测试案例错误 测试人员操作错误 测试环境中的测试数据有问题 其他因测试本身导致的错误 在计算测试出口标准的时候，本类错误不需要计算在内
项目外缺陷	不是本项目引起的，但属于行内系统的缺陷。如跟本项目相关的其他行内关联产品的缺陷，或者本产品的历史遗留缺陷等。虽然不在本项目需求范围内，但属于本产品或软件的缺陷，在计算测试出口标准的时候，需要计算在内
行外缺陷	主要包括两类：1) 对接项目，在测“与他行或其他机构对接的项目”过程中，发现的属于他行或其他机构方程序的缺陷，非我行软件的错误，

	在计算测试出口指标的时候，不需要计算在内，但是需要在总结报告中特别说明；2) 外购项目，在测“由外包公司负责开发的项目”过程中发现的缺陷，在计算测试出口指标的时候，需要计算在内
重复错误	与其他缺陷相同的重复错误，在计算测试出口指标时，本类缺陷不需要计算在内

表 7-3 缺陷类型说明

8. 缺陷处理常见问题（FAQ）

8.1. 管理人员 FAQ

1、 测试版本如何发布？

A: 建议每个项目开始时，测试负责人就需要与项目经理、技术经理讨论版本规划，确定测试版本的发布方法。包括正常的、紧急的版本多长时间发布一次，每次发布程序时需要整理一个本次发布程序修改/本次发布修复缺陷列表给测试工程师。

对较大的项目，建议设专门的版本管理员，由版本管理员检查每次版本发布是否正确，发布版本到测试环境，提供正确的版本信息给测试工程师，并将本次发布版本中已修复的缺陷改为“已发布”状态。

2、 测试环境、测试数据和参数配置类问题是否算缺陷？

A: 首先明确测试环境相关的问题是缺陷，归入缺陷类型“环境异常”。环境问题可能是参数配置错误、测试数据问题、被测版本错误、程序未编译、硬件错误、进程/服务/PJ 异常、队列满、连接数过多、通讯异常、网络阻塞等各种问题，有的是测试工程师造成的，有的是软件工程师造成的，不管是谁造成的，都必须提缺陷到测试管理平台。

如果是参数配置类错误，需要检查测试环境参数配置与生产环境参数配置的差异，确认在生产环境是否会发生问题，避免问题泄露到生产环境，出现生产事故。

如果项目“环境异常”缺陷较多，或者环境异常导致测试无法进行，测试负责人必须协调相关各方（软件工程师、测试工程师、需求人员和测试环境管理支持人员等），讨论问题解决方案，快速解决问题。

对于比较大型的项目，在大版本发布到测试环境时，应指定少量人员提前做好环境验证工作，避免造成大量测试工程师无法正常工作的情况。

3、 文档问题如何处理？

A: 不管是需求文档、设计文档还是测试文档，文档相关问题也需要提缺陷，类型为“文档错误”，分配给文档编写人员，由文档编写人员负责修复缺陷。

如果不只是文档本身的问题，而是流程问题，则可以提交给 QA 和项目经理，改进项目流程。比如需求变化直接修改代码没有修改文档，需求变化没有经过正式的需求变更流程等。

4、对已修复的缺陷，如何确定已修复的版本已经发布到测试环境？

A：对“已修复”状态的缺陷，修复后的版本可能还没有发布到测试环境，为避免测试工程师做无用的验证工作，建议测试负责人与项目经理、技术经理商定使用“已发布”状态，项目组安排一个版本管理员（可兼职）负责版本的确认和发布，修改本版本已修复缺陷的状态为“已发布”，测试工程师只验证“已发布”的缺陷，不验证“已修复”的缺陷。

如果不使用“已发布”状态，在回归缺陷前，测试工程师需要与软件工程师确认修复版本是否已发布到测试环境。

5、项目外缺陷如何确定，如何处理？

A：原则上，项目外缺陷需要软件工程师、需求人员和测试工程师的共同确认。如果是其他项目提出的与本项目相关的缺陷，需要在本项目中新建缺陷，进行跟踪管理；如果是本项目发现的其他项目缺陷，需要将该缺陷转发给对应的项目。

6、需求或设计变更后，原来的缺陷存在的前提发生了变化，如何处理？

A：要求软件工程师将该缺陷改为“已修复”状态，并说明影响该缺陷的变更内容，测试工程师做关闭处理。

7、无法重现或无法定位的缺陷如何处理？

A：首先，该类缺陷需要提出，不能随意修改为“关闭”或“非问题关闭”状态，因为该类问题有较大可能是由于隐蔽的逻辑错误导致的，对应致命和严重级别的，必须跟踪解决，实在无法在上线前解决的，尽可能给出详尽有效的应急方案。对测试结束时无法解决的缺陷，可以转为内部需求进行处理。

对此类缺陷，可以给具备一定能力的测试工程师开放一些权限，进行白盒或灰盒测试，扩大覆盖范围。

不要因为该类缺陷会遗留下来，导致出口条件无法满足要求而不提缺陷，或随意将缺陷状态修改为“关闭”或“非问题关闭”。如果无法满足出口条件，领导确认后仍然可以结束测试，但缺陷必须记录下来，以便于后续分析和优化。

8、非问题关闭或已否决的缺陷是否有效缺陷？

A：测试工程师提出一个缺陷时，处于“新建”状态，如果在测试组内部审核时，确认不是缺陷，将该缺陷置为“非问题关闭”状态，那么不算有效缺陷。

对测试组内部确认为缺陷，被软件工程师否决的，对测试工程师来说是有效缺陷，但在统计时，不算软件工程师泄露的缺陷。

9、如何提高缺陷解决效率？

A：首先，测试负责人在项目启动时，就需要与项目经理、技术经理就缺陷管理方法达成一致，在测试执行开始前，可以让QA给项目组成员进行一次缺陷管理相关流程、规范的培训，使项目组成员在观念、流程和方法上达成一致，有利于提高效率。

在测试执行阶段，可以每天或隔天召集相关各方进行缺陷分析会议，并跟踪各类严重程度缺陷的解决时间，确保严重以上的缺陷优先得到解决。

8.2. 开发人员 FAQ

1、 分配给我的缺陷实际上与我无关，不是我负责的程序的问题，如何处理？

A：如果明确知道是那个软件工程师负责的，直接转给相应软件工程师。如果不能确定，写明不是自己负责模块的问题，转给技术经理，由技术经理进行再次分配。请注意，此时不要修改缺陷状态，只需要修改缺陷对应的开发人员即可。

2、 无法跟踪定位或无规律重现的问题如何处理？

A：力争通过增加日志信息、调试时设置不同条件，覆盖不同分支等方法来查找问题，如果实在无法解决，也需要保留该缺陷，后续再看有没有解决方案。

如果因为该问题导致无法达到出口条件，可根据流程，找相关领导签字后，结束测试过程。

3、 文档问题是否算缺陷？

A：文档问题也是缺陷。如果是没有经过变更流程，导致文档与代码不一致，需要提醒项目经理和QA，应按照变更管理流程来管理变更操作。

4、 缺陷修复细节是否需要告知测试工程师？

A：项目经理和技术经理应制定规范，缺陷修复方法应在缺陷管理的备注中进行描述。这样有助于测试工程师和软件工程师共同积累经验，在后续的测试中可以提高效率，对类似问题防患于未然。

5、 否决缺陷需要填写什么内容？

A：必须填写否决理由，对无理由否决的缺陷，测试工程师将重新打开。

6、 重复提交的缺陷是否可以直接否决？

A：重复提交的缺陷有多种原因，有可能是不同测试工程师发现的相同问题，也可能是版本问题导致的，或者出错原因相同，但表现现象不同，测试工程师无法判断等，对重复提交的缺陷，需要与测试工程师沟通，如果确认是重复缺陷，由测试工程师将缺陷修改为“非问题关闭”。

7、 对于已知的问题，需求人员也接受现状，不打算修改的缺陷，是否可以否决？

A：不要否决此类缺陷，可置为“推迟处理”状态，并在测试结束前置为“转业务需求”状态。

8、 需求分析和设计过程是否需要让测试工程师参与？

A：需要，只有让测试工程师了解需求和设计变化过程，才能更深入地理解业务，了解系统实现方法，保障测试效果。

9、 一个缺陷需要多个软件工程师修改不同模块，自己负责部分修复完成了，如何处理？

A：测试管理平台上每个缺陷只能对应一个软件工程师，对这种情况，需要确定一个主负责人，修复完成后，通知主负责人，由主负责人修改缺陷状态为“已修复”，并协调版本管理员发布完整版本（涉及所有修复的模块）。

10、 缺陷报告中的软件工程师是我的名字，缺陷状态是“已分配”，是否需要处理？

A：需要处理。这一般是技术经理分配给你来处理的缺陷，如果确定不是你应该负责的问题，可以转回给技术经理，由技术经理进行再次分配。

8.3. 测试工程师 FAQ

1、对于有争议的缺陷，应如何处理？

A：由需求人员、软件工程师和测试工程师共同讨论，最终由三方共同决定。当测试工程师认为该缺陷不能被否决时，即使是开发和业务都同意否决，也需要上报测试中心领导进行确认。

2、不确定的问题要不要提缺陷？

A：要提缺陷，后续还有测试设计人员和测试负责人把关。对于典型的不能确定的问题，可以进行组内分享，说明为什么是缺陷或为什么不是缺陷，提高测试工程师的判断能力，下次遇到类似问题就可以确定了。

3、软件工程师总是否决缺陷，该怎么办？

A：软件工程师否决缺陷，必须填写理由，如果无理由否决，可以重新打开。可以与软件工程师保持较好的沟通，确定在哪些方面存在争议，如果测试执行人员将缺陷重新打开后，软件工程师又进行否决，可以上报给上级，由上级进行协调。

对于典型的案例，可以进行组内交流和分享，提高测试工程师分析问题的能力。

4、软件工程师和需求人员意见不一致，听谁的？

A：测试工程师要有自己的分析和想法，不能轻信任何一方的观点。对于存在争议的问题，可召集相关人员在 GKE 上讨论解决，或者开会解决，如果不能达成一致，上报给自己的上级，让上级与开发人员和需求人员进行协调。

5、重复的缺陷如何处理？

A：同一表象可能是不同原因导致的，也可能是版本问题，如果能确定是相同原因导致的，已有缺陷修复后，本缺陷也会消失的就不需要提新缺陷，否则需要重新提一个新缺陷，并注明是重复缺陷，与原来哪个缺陷关联。如果是回归“已修复”或“已发布”的缺陷，将缺陷状态改为“重新打开”。测试负责人或测试设计人员需要对缺陷进行审核，如果发现是明显重复的缺陷，需要关闭重复缺陷，降低软件工程师的工作量。

6、提的缺陷太多，软件工程师生气了，怎么办？

A：与软件工程师沟通，说明测试和软件工程师的目的都是一样的，都是为了程序质量，不是针对某个软件工程师，我们不提出缺陷，泄露到生产环境，问题只会更严重。另外，力争发现更有价值的缺陷，得到软件工程师的认可。

如果与软件工程师沟通存在困难，提交给测试负责人，由测试负责人出面协调。

7、与测试内容无关的缺陷要不要提？

A：要提，只要是缺陷就要提出来。与测试内容无关的缺陷可以算项目外缺陷，可以由测试负责人提交给相应项目或相应软件工程师。

8、测试环境相关的缺陷要不要提？

A：要提，在提出缺陷时，描述当前的测试环境，重点记录可能与当前缺陷相关的环境参数和配置信息，并跟踪解决过程，记录解决方法。

测试人员需要了解环境问题的解决方法，总结经验教训，提升解决环境问题的能力，在出现类似问

题的时候，提高解决环境问题的效率。

9、 何时提交缺陷？可否集中提交？

A：缺陷提交务必及时准确，有些缺陷是需要保留现场的，时间长了无法重现和定位，建议对缺陷进行分类，需要保留现场的缺陷不但要及时提出，还要及时上升到项目管理层，立即跟踪定位。

10、 缺陷在哪个版本进行验证？

A：测试负责人在项目启动时，需要与项目经理和技术经理商定版本管理策略，确定多长时间发布一个测试版本，如何提供当前发布的测试版本涉及的内容（包括新增业务功能、变更和修复的缺陷列表等），如果遇到导致测试无法继续进行的严重或致命缺陷如何处理，是否启用“已发布”状态等。在测试执行过程中，测试负责人应密切关注缺陷状态，主动联系相关软件工程师，确保严重级别和优先级高的缺陷能得到及时处理，保证缺陷修复的速度符合预期，以使测试工作进展符合测试计划的要求。对测试执行人员来说，需要关注测试版本的更新情况，尽快完成最新测试版本中已修复缺陷的回归测试。如果已启用“已发布”状态，或者每次测试版本发布均列出了当前版本已修复的缺陷清单，测试执行人员就可以明确当前可以回归测试哪些缺陷了；如果不具备上述条件，即使缺陷处于“已修复”状态，也需要主动联系软件工程师，确认是否已在本次测试版本更新中发布，避免无效测试。

11、 无规律重现的缺陷或者不能重现的缺陷要不要提出？

A：要提出。对这类缺陷，一旦重现，需要保留现场，通知软件工程师到测试环境进行调试分析，以便定位问题。

12、 项目 ST 测试结束时，无规律重现的缺陷或者不能重现的缺陷也没有解决，如何处理？

A：测试工程师将该缺陷置为“推迟处理”状态，在出口条件缺陷遗留数或缺陷遗留率的计算时，遗留缺陷数的计算不包括无规律重现或不能重现的缺陷。

13、 对于已知的缺陷，生产环境已经存在该问题，需求人员也接受现状，软件工程师也不打算在本项目中修改，如何处理？

A：测试工程师提出缺陷，软件工程师可将其置为“推迟处理”状态，如果需求人员希望在后续项目中修改，在最后一轮 ST 测试执行开始前，测试工程师应将其置为“转业务需求”状态。后续需求人员可以跟踪该问题，可以在合适的版本中提出业务需求，要求解决该问题。如果是维持现状，后续项目也不做修改，那么维持该缺陷为“推迟处理”状态。

14、 有的缺陷软件工程师、需求人员都承认存在问题，是缺陷，但改动的成本更大，经协商就不在本项目修改了。这类缺陷如何处理？

A：测试工程师提出缺陷后，经协商不在本项目修改，可由软件工程师将缺陷状态改为“推迟处理”，在测试结束前，测试工程师将该缺陷置为“转业务需求”或“转内部需求”。在该产品后续的项目中，如果有合适的机会，可以作为业务需求或内部改进需求提出。如果确定后续项目也不修改，那么可维持该缺陷为“推迟处理”状态。

15、 存在争议时，如何进行沟通？

A：争议的沟通务必保留书面记录，可以是邮件或会议纪要，必须得到相关方的确认，并将沟通记

录作为缺陷记录的附件，保存到测试管理平台上。