

缺陷管理的流程

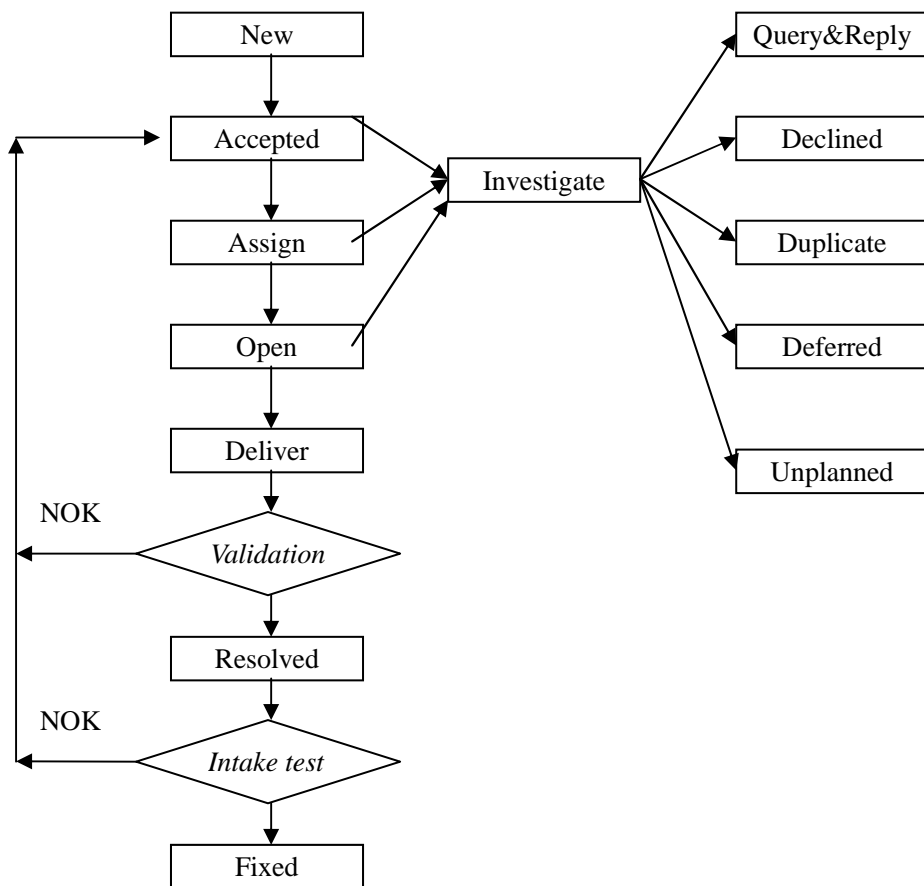
在软件的开发、评审、测试和使用的过程中，我们都可能面临或碰到软件产品没有按照设计要求运行、使用不方便或在某种程度上不能满足用户的要求等此类问题，这些我们可以通称为缺陷。

软件缺陷是软件开发过程中的“副产品”。缺陷会存在于软件产品的整个生命周期中：可以是软件代码的问题、系统文档（开发文档和测试文档等）存在的问题，或者是用户的帮助文档和使用指南方面的问题等。

测试是发现缺陷的主要手段，也是它的主要目的。测试活动和开发活动一样，是项目质量保证不可或缺的重要部分。因此，对于测试活动的主要产物：缺陷，我们需要建立一个完善的缺陷管理流程，来对缺陷进行报告、查询、分类、跟踪、处理和验证等。

本文主要针对在开发测试活动中发现的缺陷，其相应的缺陷管理流程，以及在流程中主要的缺陷状态、参与缺陷的角色和缺陷相关的主要活动以及缺陷的等级分类等。

1. 缺陷状态的主要处理过程：



2. 和缺陷相关的角色：

- 测试工程师：在这里主要是指发现和报告缺陷的测试人员。在一般流程中，他需要对这个缺陷后续相关的状态负责：包括相关人员对这个缺陷相关信息的询问回答，以及在 build 中的验证测试和后面正式版本的验证测试。
- 开发工程师：这里主要指对这个缺陷进行研究和修改的开发人员。同时，他需要对

修改后的缺陷在提交测试人员正式测试验证之前需要进行验证测试。

- **缺陷评审委员会**：主要由项目经理、测试经理、质量经理、开发经理以及资深的开发、测试工程师等组成。他们对缺陷进行确认以及将之分配给相应的开发人员进行修改。
- **版本经理**：负责将已经解决的缺陷相关的配置信息融入到新的版本，提交新的测试和相关的验证测试。

3. 缺陷状态的含义解释：

- **New（新缺陷）**：软件中新发现报告的缺陷，一般由测试人员提交。当然也可能是开发人员自己在单元或代码测试过程中提交，或从软件使用的最终用户或测试现场反馈得到的缺陷报告。
- **Accepted（接受）**：经过缺陷评审委员会的确认，认为缺陷确实存在。
- **Assign（分配）**：将这个缺陷分配给相关的开发人员来进行修改。
- **Open（打开）**：处于这个状态时，缺陷已经被确认并已经分配给相关的开发人员进行相关的修改。
- **Deliver（交付）**：解决缺陷问题的方法已经找到，并且已经将修改后的代码等打上标签，交付给版本经理。
- **Resolved（解决）**：版本经理将相关的标签等融入某个 build，交付给相关的开发小组进行验证测试，测试通过，则缺陷状态改为解决状态。
- **Fixed（已修改）**：版本经理将已经解决的缺陷标签融入某个版本，交付给相关的测试小组进行验证测试，测试通过，则缺陷状态修改为已修改状态。
- **Closed（结束）**：缺陷状态处于已修改后，自动变为结束状态。

上面简单介绍的缺陷状态是在缺陷管理过程中主要的状态，或者是在缺陷处理顺利时所经历的状态。实际上，缺陷还有其他一些其他的状态，或者可以认为是辅助的状态，分别是：

- **Investigate（研究）**：当缺陷分配给开发人员时，开发人员并不是都直接可以找到相关的解决方案的。开发人员需要对缺陷和引起缺陷的原因进行调查研究，这时候我们可以将缺陷状态改为研究状态。
- **Query&Reply（询问和回答）**：负责缺陷修改的工程师认为相关的缺陷描述信息不够明确、或希望得到更多和缺陷相关的配置和环境条件、或引起缺陷时系统产生的调试命令和信息等。
- **Declined（拒绝）**：缺陷评审委员会通过相关的讨论研究，认为不是缺陷。或通过开发人员的调查研究，认为不是缺陷，开发人员可以将具体的理由加入到缺陷描述中，缺陷评审委员会根据此将缺陷状态修改为拒绝状态。
- **Duplicate（重复）**：缺陷评审委员会认为这个缺陷和某个已经提交的缺陷是同一个问题，因此设置为重复状态。
- **Deferred（延期）**：缺陷不在当前版本解决。
- **Unplanned（无计划）**：在用户需求中没有要求或计划。

4. 缺陷的严重度和优先级分类：

缺陷的严重度指得是假如缺陷没有修改，由这个缺陷引发的问题对客户的影响程度。而缺陷的优先级指得是解决这个缺陷需要的时间（或者在多少时间内必须解决这个缺陷）。对于一个缺陷，我们首先会给它指定一个严重度，而后给出它的优先级。我们下面来简单介绍缺陷的严重度和优先级的分类，提供一些分类的建议和思想。

缺陷的严重度，我们可以通过 1 到 4 来划分：

- **严重度 1—最高级别**：产品在正常的运行环境下无法给用户提供服务，并且没有其他的

工作方式来补救。我们可以将下面的问题定义为严重度 1 级：

1. 问题会自发的影响系统的数据传输。
 2. 用户使用正常的操作步骤，就会影响系统提供的服务。
- 严重度 2—高级别：极大的影响了系统提供给用户的服务，有其他的工作方式来缓解这种影响。举例：
 1. 系统中的一些单板会自动重启，并没有影响它们所提供的传输性能。
 2. 用户使用正常的命令，会导致系统重启或挂起，但不影响系统的数据传输。
 - 严重度 3—中等级别：系统需要增强的或存在的一些缺陷，但有相应的补救方法来解决这个缺陷。举例：
 1. 系统的一块单板失效了，但系统没有上报相应的告警。
 2. 功能特征设计不符合系统的需求，不影响系统的业务，并且有相应的补救方法。
 - 严重度 4—低级别：细小的问题，不需要补救方法或功能增强的请求。举例：
 1. 上报的信息不符合系统的需求，描述不精确或可能对用户有些误导。
 2. GUI 界面问题，不精确或可能对用户有些歧义。

缺陷的优先级，我们可以进行下面的分类：

- 紧急的 (Emergency)：缺陷会对系统引起重大问题，必须尽快解决。
- 必须的 (Must)：在客户的下个交付之前必须解决。
- 应该的 (Should)：在客户的下个交付之前应该解决。
- 可选的 (Optional)：在客户的下个交付之前可选择解决。
- 不需要 (Don't)：在客户的下个交付之前不需要解决（由于解决的风险太大或这个功能特征不需要等）。