

## 本周小结（第五周）

祝贺大家完成《软件测试与质量》课程第五周的学习。

本周我们讨论了课程的第三部分 测试管理与应用篇，并主要围绕第 4 章 测试管理与应用展开讨论。

本周我们主要回答了如下的问题。

## 1 如何管理测试用例

测试用例的管理中，主要应解决两个问题：

- 如何组织测试用例

即通过分析用户需求，设计测试用例，从测试用例库中抽取不同的测试用例，构建多个测试集，在不同的测试轮次中使用不同的测试集执行测试，记录发现的缺陷，并对应到相关的测试用例。

- 如何报告测试用例

即通过测试用例报告来详细记录测试用例的输入、预期输出，体现测试用例与需求的对应，支持测试用例的精确执行和责任划分。测试用例报告就是要回答：谁，在什么条件下，对什么进行测试，如何进行测试，依据的需求是什么，与其他测试用例有何关联。

## 2 如何管理缺陷？

### 2.1 缺陷的基本属性

缺陷的属性主要包括：可重现性，严重性，优先级，。

无法重现的缺陷无法修复，但不是任何缺陷都可以重现。因此，要保证发现的缺陷可以重现的措施如下：

- 备份相关环境和数据；
- 测试过程中，详细记录每一个执行步骤和系统的响应；
- 改变输入条件，输入数据，操作步骤，或测试环境，尝试重现缺陷；
- 一旦确定缺陷可以重现一次，重复执行至少 3 次。

严重性是缺陷的客观属性，用于客观评价缺陷对系统造成的破坏力。一般分为三级：严重的，一般的，次要的。

优先级是缺陷的主观属性，是指项目组对缺陷的处理优先级，带有主观意见。一般分为三级：高、中和低三种。

可修复性是指缺陷在软件产品发布之前是否可以得到修复。不是所有缺陷在产品发布之

前都可以得到修复，但应尽量促使缺陷在产品发布之前进行修复。

## 2.2 缺陷的报告

一份缺陷报告实际就是要回答如下问题：

- (1) 谁，何时，在何处，发现了什么缺陷？
- (2) 谁，何时，提出怎样的处理意见？
- (3) 谁，何时，如何修复该缺陷？
- (4) 谁，何时，如何验证该缺陷？测试结果如何？

## 2.3 缺陷的跟踪

缺陷的跟踪如图 2.1 所示。

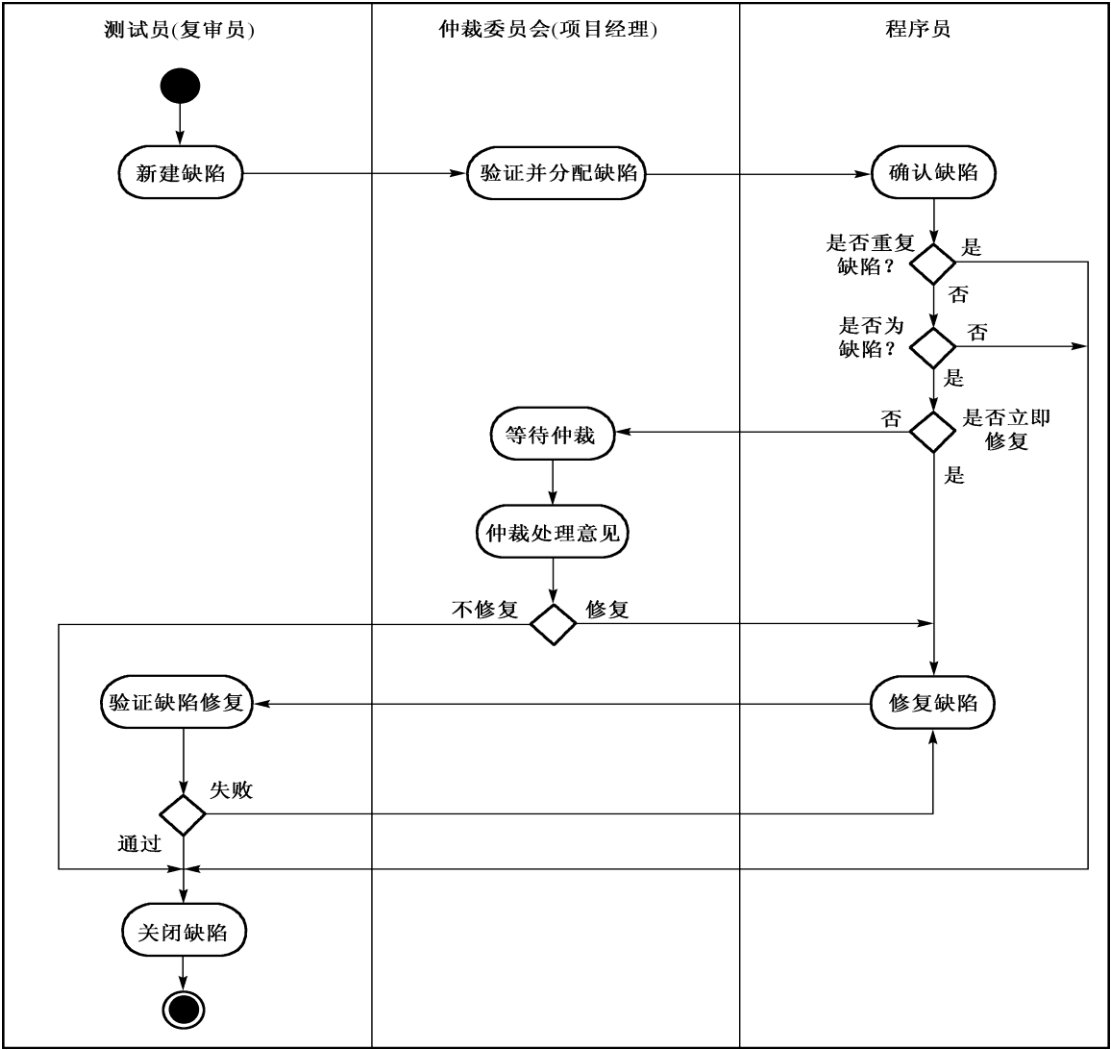


图 2.1 缺陷的处理流程

简言之，缺陷跟踪的流程就是缺陷从激活，到分配，到解决，最后关闭的过程。

# 3 如何利用管理工具实现测试管理？

## 3.1 TC 的基本使用流程

TC 的基本使用流程如图 3.1 所示：

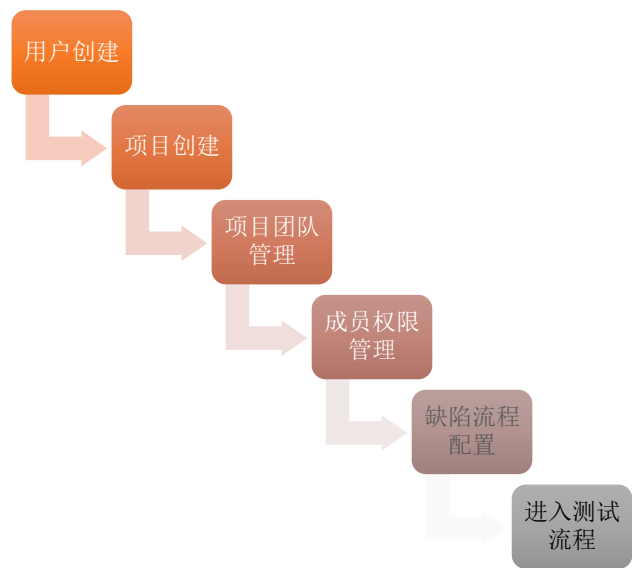


图 3.1 TC 的基本使用流程图

TC 的测试流程如图 3.2 所示。

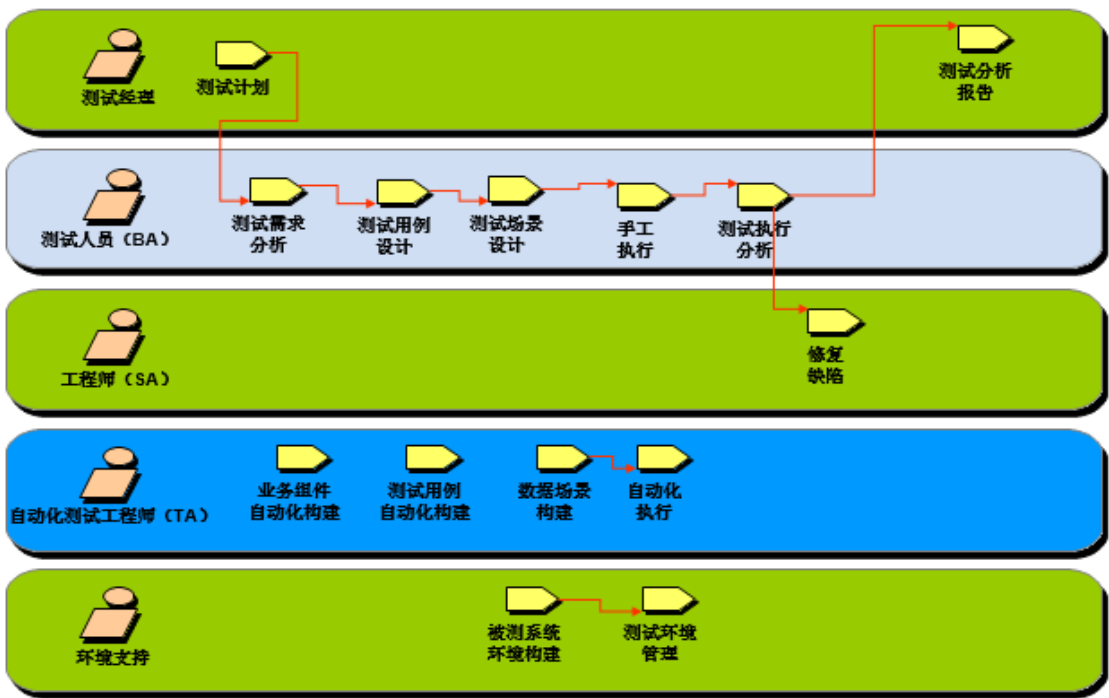


图 3.2 TC 的测试流程图

## 3.2 设计测试计划和测试需求

测试计划可按版本进行管理，不同软件版本对应不同测试计划，在测试计划下，设计测试轮次，一个测试计划可以包含多个测试轮次，测试执行是按照轮次来完成的。

测试需求则实际对应功能需求，可通过导入 word 文档的方式批量添加需求，或者手写输入需求，最终形成需求树，测试用例必须与需求相对应。

## 3.3 设计测试场景和测试用例

测试场景是针对业务流程设计的，且与测试需求相对应，包含步骤设计和数据设计。

测试用例与测试场景相对应，一个场景可能包含多个测试用例。测试用例的设计包括数据设计（输入数据）、步骤设计（可从场景设计中导入），和预期输出。测试用例的设计方法包括：

- 在测试场景中添加测试用例（上节中已介绍）
- 在测试构建中，在测试用例组中添加测试用例
- 通过设置用例模板，批量导入测试用例

## 3.4 构建测试集并关联测试集

测试集的构建主要用于对测试用例进行管理，在版本更新迭代中，通过测试集将测试用例组合起来，不同的版本可选择不同的测试集进行执行，具体方式包括：

- 测试集树中添加测试集组
- 添加测试集
- 在测试集中添加测试用例

然后，需要在测试计划中关联测试集，从而将需求→（对应到）测试用例，测试轮次→（关联）测试集→测试用例，相互对应起来。

最后发起手工执行。

## 3.5 测试执行和缺陷管理

发起手工执行之前，由项目经理分配执行人，执行人对照用例信息，进行手工执行，如果执行失败，需将用例执行状态改为【错误】，并填写和提交缺陷报告。

报告缺陷的方式有两种：

- 方法 1：在执行测试用例的过程中，用例执行失败，触发提交缺陷；
- 方法 2：在缺陷视图中，直接提交缺陷，适合于非用例执行过程中（例如，随机测试）发现缺陷的情况。

最终由项目经理登录系统，查看项目进度、缺陷统计、项目分析等。

## 6 下周预告

下周，我们将继续讨论第四章 测试管理与应用的讨论，并围绕功能测试和性能测试，展示功能和性能测试工具的使用。