

测试管理工具 testlink

根据实际情况，基本上我们可以按照如下的流程来实施我们的测试管理方案：

首先创建项目

然后创建需求

创建计划

创建用例

给需求指派用例（可能不止一个）

给计划添加用例

为用例指定执行者

执行计划/报告 bug

查看分析结果

1. 创建项目：

主页左边的列表栏有“Test Project management”的菜单，子菜单中有“create new test project”，通过它可以创建新的测试项目。

同时，菜单中的其它子菜单可以实现对已存在的 test project 的编辑，删除，以及设置已存在的用户对于某一个测试项目的权限。

默认设置下，只有 admin 组的成员拥有对 test project 进行操作的权力。

2. 创建需求：

主页左边的列表栏中有“Requirements”的菜单，子菜单中有“Requirement Specification”，可以添加编辑需求规格说明书。

同时，菜单中的另一项可以为需求指定测试用例（结果统计的时候会有一种根据需求覆盖率进行统计的方式）。

需要说明的一点：每一个需求都必须有相应的多个 Req——实际上就是我们对需求进行分析，然后把它分成一条一条的，测试用例是与这些 Req 相对应的。

默认设置下，只有 admin 组的成员拥有对 Requirements 进行操作的权力。

3. 制定测试计划：

主页右侧列表，有专门的“Test Plan Management”的菜单，选择其子菜单中的“Test Plan Management”，进入的页面会出现“create”的按钮，点击即可以创建新的测试计划。

4. 创建用例：

首先需要说明一下 testlink 用例树的层次：

Test project —— test suite —— test case

所以在创建测试用例之前，需要保证用例隶属于的 test project 和 test suite 都已经存在了。

上面已经讲过如何创建测试项目了，接下来说明一下如何创建 test suite 测试集。

当测试项目创建完毕的时候，选择横向导航条中的“specification”，出现的页面还是分左右两部分——左侧是 用例树。

树的根节点就是咱们创建的测试项目（页面右上角可以切换测试项目，类似 mantis）。

点击测试项目，右侧页面内容中会有“new test suite”的按钮，点击可以创建 test suite（测试集——可以理解成测试项目的一个功能模块）。

Test suite 创建完成以后，刷新用例树（左侧页面内容，update tree），可以看到用例树中已经出现了我们刚才创建的测试集。

点击测试集，右侧页面内容中会出现“create test case (s)”的按钮，点击可以创建新的测试用例。

测试用例创建完毕以后，刷新用例树，则会看到用例树中 test suite 的下一级中出现了我们刚刚创建的 test case。

注：用例是可以指定版本的——因为随着需求的变化，或者其他某些因素，用例是要不断变化的，需要用版本号来区别这种变化。

PS：选择不同的 level，右侧页面中会出现不尽相同的各种按钮——每个按钮对应的操作与其字面意思是相对应的，例如

a) 用例树中我们选择的是一个 test project，右侧页面中会出现如下按钮：

New test suite —— 创建测试集

Reorder children —— 对该测试项目的子项（test suite）进行重新排序

Import test suite —— 导入测试集

Export all test suites —— 导出所有的测试集

b) 用例树中我们选择的是一个 test suite，右侧页面中会出现如下按钮：

Edit —— 编辑测试集

Delete —— 删除测试集

Move/copy —— 移动或者复制测试集

Reorder children —— 对该测试集的子项进行重新排序

Export test suite —— 导出测试集

New test suite —— 新建测试集

Import test suite —— 导入测试集

Create test case(s) —— 创建测试用例

Import test case(s) —— 导入测试用例

Export test case(s) —— 导出测试用例

c) 用例树中我们选择的是一个 test case，右侧页面中会出现如下按钮：

Edit —— 编辑当前用例

Delete —— 删除当前用例

Move/copy —— 移动/复制当前用例

Deactivate this version —— 将当前用例版本设置为 无效 状态

Create a new version —— 为当前用例创建一个新版本

Export —— 导出用例

5. 为需求指派用例：

主页左边的列表栏，“Requirements”的子菜单中有“Assign Requirements”的选项。

选择以后，会进入“specification”类似的界面。

左侧用例树中选择某个测试用例，右边页面内容会出现需求列表。

前面我们已经说过，测试用例是与需求的某一个 Req 相对应的。

在合适的 Req 前面的复选框中打勾，然后点击下面的“Assign”按钮，就完成需求的指派了。

当然，也可以撤销掉需求与用例的关联——该页面会同时有“unassign”的按钮。

6. 给计划添加用例：

主页右侧列表中有“test plan contents”的菜单，其子菜单中有“Add Test Case(s)”的子菜单。

点击这一项，会进入类似“specification”的页面——但是左侧用例树中只列到 test suite 这一级。

选择某个 test suite，右侧页面会列出该测试集所包含的所有测试用例，在需要添加到计划中的测试用例前面的复选框中打勾，然后点击下方的“add selected”按钮即可将选择的测试用例添加到测试计划中。

当然，也可以移除添加到计划中的用例。

添加到计划中的测试用例会用黄色打底，后面出现 remove 的复选框，勾选，点击下方的“add/remove selected”即可完成移除操作。

7. 为用例指定执行者：

接下来我们要做的事情，是为测试计划中所包含的每个用例指定一个具体的执行人员。

首页，右侧列表，“Test Plan Contents”，其子菜单中有“assign Test Case execution”，选择这一项我

们可以进入下一个页面，为测试用例指定实际的执行者。

该页面中，左侧用例树中选择 test suite 或者 test case，右侧页面会出现下拉列表让你选择 user，选择合适的人员，然后 test case 前面打勾，点击右侧页面下方的按钮即可完成用例的指派工作。

当然，这里也可以进行批量指定——右侧页面的最上方，有一个下拉列表可以选择用户，下面的 test case 列表中选择要指派给该用户的用例，然后点击一下后面的“do”按钮即可完成将多个用例指派给一个人的操作。

8. 执行计划/报告 bug:

我们把他们放到一起，是因为报告 bug 是在执行的过程中同步进行的——即执行用例的过程中一旦发现 bug 我们需要立即将其报告到我们的 bug 管理系统中去。

执行测试计划以前，需要为测试计划创建一个 build 版本——我们可以这样，用日期来标识，表明我们执行测试计划的日期；当然了，也可以用其他含义的标题，诸如本次测试执行的侧重点什么的。

首页右侧列表，“Test Plan Management”菜单，其子菜单中有一项“Build Management”，选择这一项进入的页面会出现“create”的按钮，即为测试计划创建新的 build 的操作。

PS：首页中，右侧最上方有一个下拉列表，用来选择当前对其进行操作的测试计划。

接下来我们就可以执行测试计划了。

首页横向导航栏中的“execute”菜单，点击进入执行页面。

该页面，同样一分为二，左侧是用例树，右侧页面内容为主体内容。

这里有一点要说明一下，虽然“执行”表面上针对的是测试计划，而实际上对应的是测试计划中测试用例的执行情况。

左侧用例树中，选择某一个 test suite，右侧页面上方会出现测试计划，build 描述，测试集的说明等等信息，还有一个批量设置该测试集中所包含的测试用例状态的按钮，即“Bulk TC status management”。

接下来则是该测试集中所包含的所有测试用例的详细信息。

每一个测试用例的最后部分，“notes/Description”，“result”是需要我们执行完测试用例以后自己来填写的。

该部分填写完成以后，在用例的开始部分会对这个结果有所记录。

同时，可以把 bug management 系统中执行该测试用例时发现的 bug ID 记录到此处——将 testlink 与 mantis 集成以后，可以通过点击一下鼠标进入到 mantis 查看 bug 的具体情况，很方便。

9. 查看分析结果:

首页，横向导航栏中的 **results** 菜单，点击可以进入结果查看界面。

该页面，可以从各种各样的角度查看执行的结果——例如，从需求覆盖的角度，用例状态角度等等。

当然，**testlink** 还有其他某些辅助性的功能，例如“**custom field**”，你可以利用这个功能自定义某个字段，让其作用于你的用例——例如，我们可以自定义一个“**priority**”的字段，给每个用例指定优先级。

对于这些辅助性的功能，我们会在以后的具体使用中慢慢的一个个去总结。

测试管理注重的实际上就是一个流程，这里我们先把大的流程给大家做一下简单的说明，这样大家可以从全局的角度来认识这个系统。

至于细小的东西，我们需要在具体的实践中不断的去摸索熟悉总结。