JIRA使用手册

JIRA基本介绍

问题

JIRA跟踪问题（Issue），这些问题可以是Bug，功能请求或者任何其他想要跟踪的任务，每个问题有一些关联的信息：

* 问题类型（Issue Type）
* 摘要（summary）
* 问题描述（description）
* 问题所属的项目
* 问题关联的项目组件（component）
* 问题影响的项目版本（affecte version）
* 问题将被解决的项目版本（resolved version）
* 问题发生的环境
* 问题的优先级
* 问题的报告者
* 问题的指派处理人
* 问题的当前状态
* 问题相关的历史记录

1.1 问题类型

JIRA系统可以用于跟踪多种不同类型的问题。系统管理员可以根据需要添加。ＪＩＲＡ系统缺省提供的问题类型如下：

|  |  |
| --- | --- |
| Bug | 测试过程、维护过程发现影响系统运行的缺陷 |
| New　Feature | 对系统提出的新功能 |
| Task | 需要完成的任务 |
| Improvement | 对现有系统功能的改进 |

1.2优先级（Priority Levels）

在JIRA系统中用优先级来表示问题的严重级别。系统管理员可以在JIRA系统中添加优先级，JIRA系统缺省的优先级如下：

|  |  |
| --- | --- |
| 级别 | 参考描述 |
| Blocker | 阻塞开发或测试的工作进度，或影响系统无法运行的错误 |
| Critical | 系统崩溃，丢失数据或内存溢出等严重错误、或者必须完成的任务 |
| Major | 主要的功能无效、新增功能建议 |
| Minor | 功能部分无效或对现有系统的改进 |
| Trivial | 拼写错误、文本未对齐等 |

1.3 状态（Status）

每个问题有一个状态，用来表明问题所处的阶段，问题通过开始于open状态，然后开始处理（in Progress），再到解决（Resolved），然后被关闭（Closed）。根据情况的不同，可以根据项目来定制问题状态以及工作流。JIRA系统提供的缺省状态如下：

|  |  |
| --- | --- |
| Open | 表示问题被提交等待有人处理 |
| In Progress | 问题在处理当中，尚未完成 |
| Resolved | 问题曾解决，但解决结论未获认可，需要重新分派解决 |
| Reopened | 问题解决，等待结果确认，确认的结果是“Reopened”或“Closed” |
| Closed | 问题处理结果确认后，置于关闭状态 |

1.4 解决（Resolutions）

一个问题可以用多种方式解决，系统管理员是可以在JIRA系统中地址解决方式。JIRA系统默认的解决方式如下：

|  |  |
| --- | --- |
| Fix | 问题已经解决 |
| Won’t Fix | 问题未解决-将不会解决的问题 |
| Duplicate | 重复的问题 |
| Incomplete | 问题描述得不够准确、 |
| Cannot Reproduce | 问题重现失败，或者无足够的信息重现问题 |

2 项目

一个JIRA系统通常包括许多项目。这些项目相当于产品或者开发项目，您想要跟踪和管理这些项目相关的问题。每一个问题都属于一个项目。每一个项目都有一个名字和一个关键字（如：Java），以后属于这个项目的问题的关键字就会包含Java（如：Java-100，Java-101）。值得注意的是：在JIRA系统中有一个权限“Administer Projects”，通常将这个权限付给项目负责人，拥有这个权限的JIRA用户就可以管理项目的“版本”和“组件”。

2.1 项目版本

在一个项目上，一般会有多个版本，如：1.0alpha、1.0beta、1.0、1.2、2.0。

JIRA系统中的问题涉及到两个版本字段：

* 影响版本，如一个不过可能影响版本1.1和1.2
* 修复版本，如bug影响版本1.1和1.2，可能在2.0版本上被解决。

版本通常有三种状态：发布（released）、未发布（unreleased）和归档（archived）。版本还有发布日期，在特定的报告中会显示。在JIRA系统中可以为项目创建版本。

2.2 项目组件

每一个项目通常会包含多个组件和模块，如：后台，GUI，邮件子系统等。一个问题可以和组件关联起来。在ＪＩＲＡ系统中可以为项目添加组件。

3 角色

JIRA作为一个缺陷跟踪管理系统，可以被企业管理人员，项目管理人员，开发人员，分析人员，测试人员和其他人员所广泛使用。

3.1 管理人员

根据ＪＩＲＡ系统提供的数据，更加准确地了解项目的开发质量和状态，以及整个团队的工作效率。

3.2 项目管理者

可以针对登记JIRA系统中问题，进行评估，分配缺陷；还可以通过JIRA系统的统计报告了解项目进展情况以及团队的工作量、工作效率等信息。

3.3 开发人员

在JIRA系统中查看分配给自己的问题，及时进行处理，填写处理情况并提交工作量记录。

3.4 测试人员

根据测试情况，在JIRA系统中及时快速的记录问题并对开发人员处理后的问题进行验证和跟踪。

4 分配给

分配给（Assign To）：把某个问题Issue指定给某个团队成员去解决。

5 上传附件或屏幕截图

上传附件或屏幕截图（Attach file/Screenshot）：附件文件或屏幕截图说明issue。

6 问题投票

投票（Voting）：除了问题的创建人，其他人可以用投票来表明支持某个问题的内容。

7 监视问题

监视问题（Watching）：每人都有一个Watch List/监视列表，可以监视自己干兴趣的issue的状态等。

JIRA基本操作

1 项目管理（Administrator）

第一步：在成功登陆JIRA后，点击管理，进入管理界面；

第二步：点击“项目”链接，右边会有“Add Project”链接，点击进入，就可以创建新项目了 ；

第三步：点击增加，项目就创建完成了。



图 创建新项目



图 项目创建成功

1.1 创建新模块

创建新模块的作用是方便系统的开发，一个系统中可能有很多模块，一个项目经理可以把项目分成很多的模块，再把所有的开发人员分成很多的小组，每小组有一个组长，每个组副则其中的一个模块。

如：毕业论文管理系统，可以有论文管理模块，学生管理模块，教师管理模块等

第一步：点击“Add a new Component”进行链接；

第二步：进入创建页面，填写其中的相关信息，Component Lead要有相应的权限；

第三步：点击添加，就完成了。



图 创建新Component

注意：一个项目可以添加多个模块，看项目的大小而定，也可以进行修改

1.2 管理版本

一个项目不可能一下子就做的很完美，在设计初期，项目会有版本管理的需求，像我们使用的软件都是有版本的（如IE浏览器，QQ等），不断的向其中添加新的功能，同样这里为了系统的不断完善，我们也可以设置版本控制。版本通常有三种状态：发布（released）、未发布（unreleased）和归档（archived），开发者可以根据系统进行的层度采取相应的版本控制。

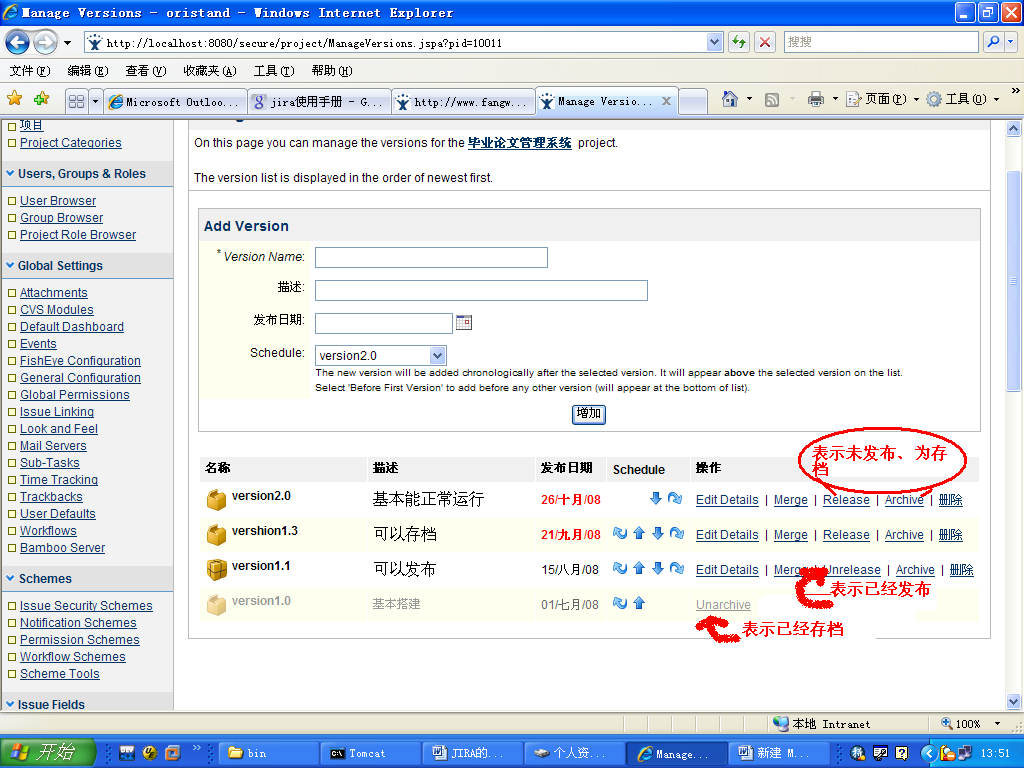


图 管理版本

1.3 创建项目种类

一个大型的公司，不可能只有一个项目，可以有很多的项目在同时进行，像软件行业的公司，可能同时在开发很多系统，这都是有关开发方面的；还有销售人员可能每年都有销售计划，这些计划也可以很多；还有采购部门等，如果项目不进行分类，一方面是不方便管理，同时也影响整体的进度。

第一步：点击“Projiect Categories”进行链接；

第二步：进入添加界面，只要输入名称和描述；

第三步：点击添加就完成了。

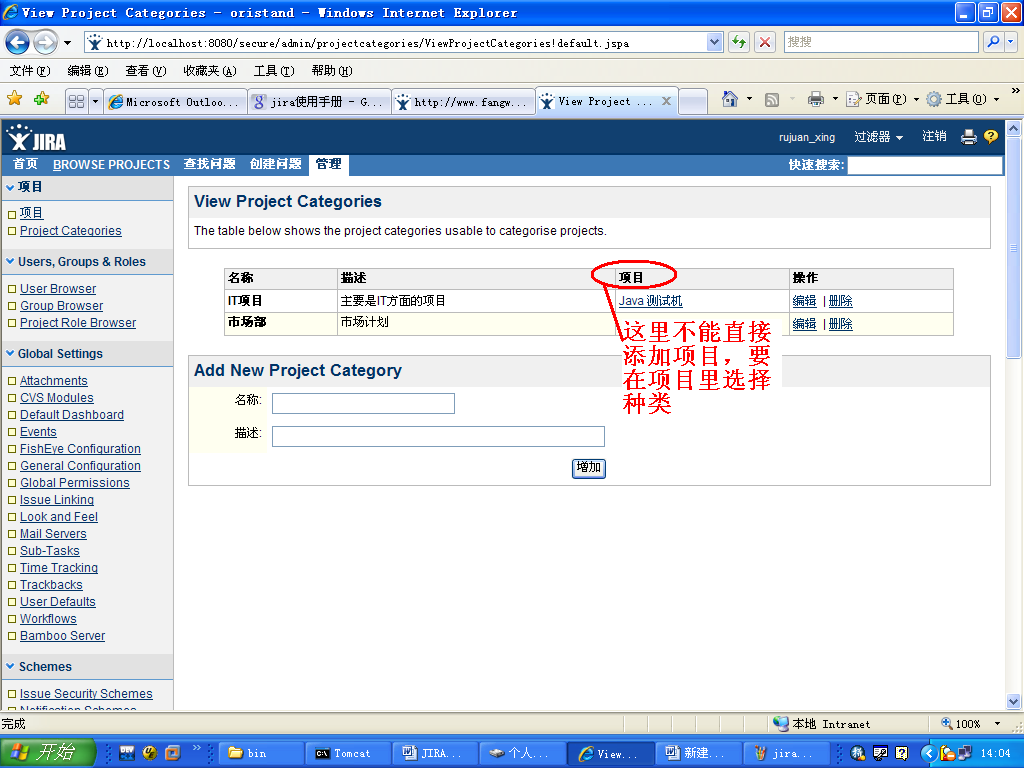


图 创建项目种类

注意：在project categories添加完成后不能直接把所属的项目放进去，要进入项目管理界面中进行更新，选择所属的种类。

1.4 问题类型方案（**Issue Type Scheme）**

这里主要是限制测试人员提交的问题的类型，通常有四种:Bug、New Feature、Task和Improvement。我们可以创建自己项目需要的问题类型，也可以使用系统默认的。



图 编辑问题类型

1.5 通知方案（Notification Scheme）

可以选择通知模型，当测试人员创建问题后，我们中的什么人员可以看见这些问题都需要进行控制，在这个模型中我们可以进行设置。

第一步：点击项目下的Notification Scheme后的“编辑”链接；

第二步：可以增加、删除人员、组等；

第三步：保存就行了。

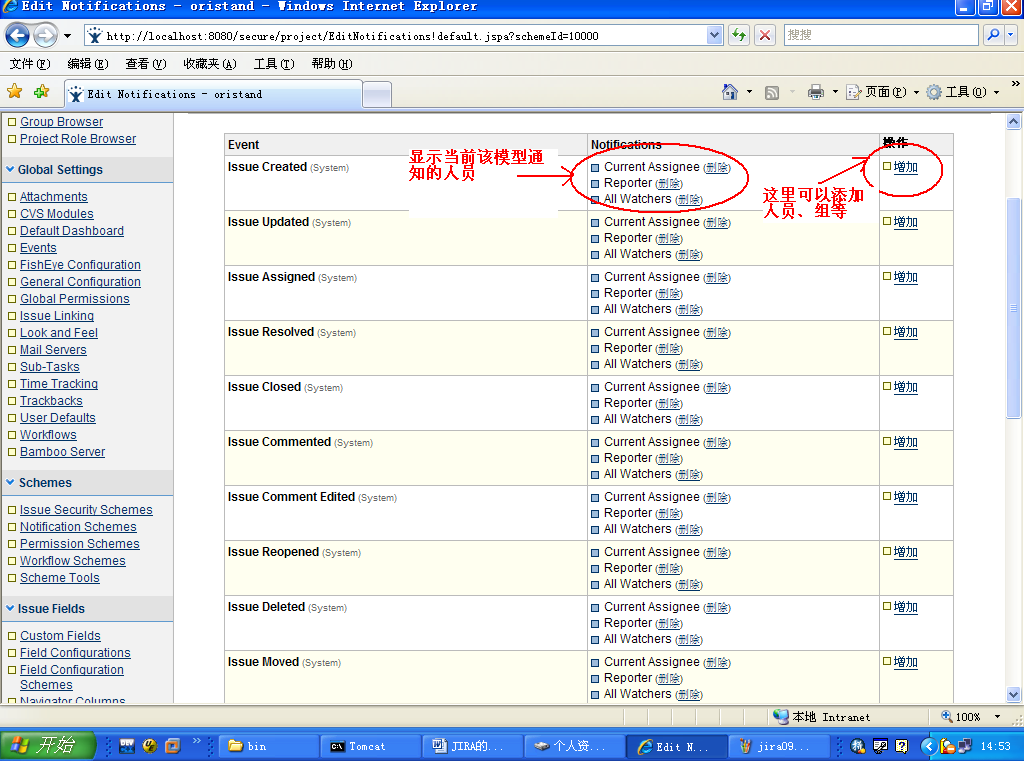


图 编辑通知方案

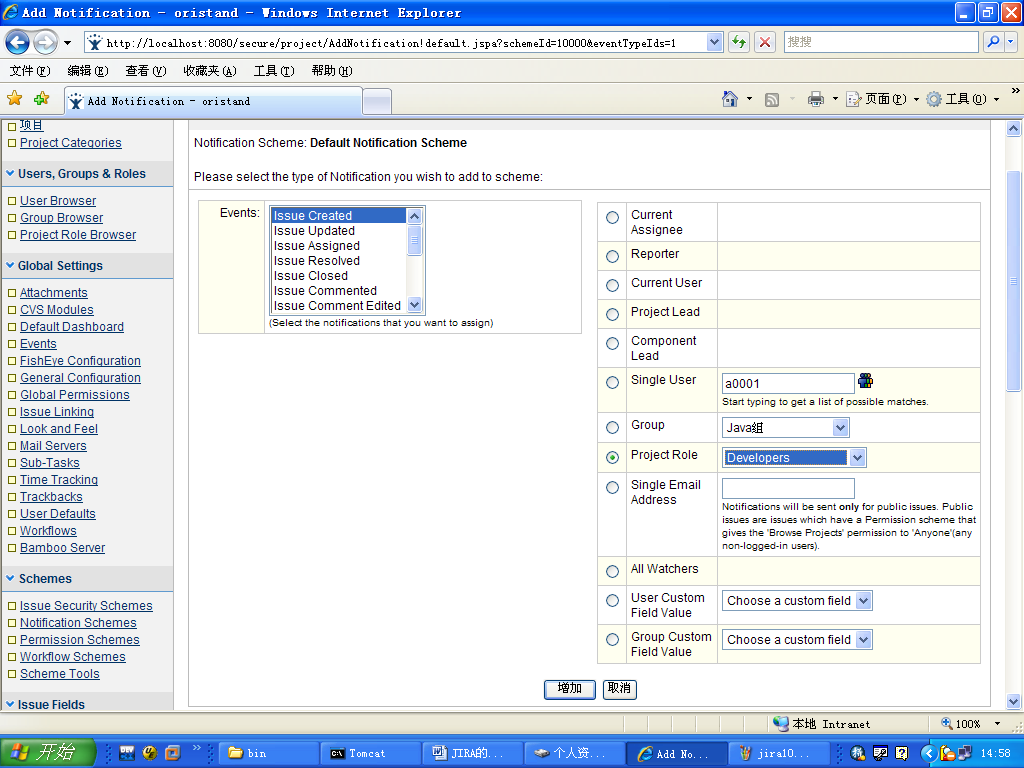


图 编辑通知方案

1.6 权限管理（Permission Scheme）

在学校、单位一样，每个人的身份是不一样的，学生的任务是学习，不可能有管理老师的权限；同样，老师的任务是育人，不会有评教的权利。在一个项目开发过程中，每个人所拥有的权限也是不一样的。项目的管理人拥有最高的权限，他能够管理开发者，测试人员等，也可以增加项目的模块、管理版本等。这里的权限管理有六个模块，主要为：Project Permissions、Issue PermisTime Tracking Permissionssions、Voters & Watchers Permissions、Comments Permissions、Attachments Permissions。

第一步：点击Permission Scheme后面的编辑进行管理；

第二步：进行需要的操作保存就行了。

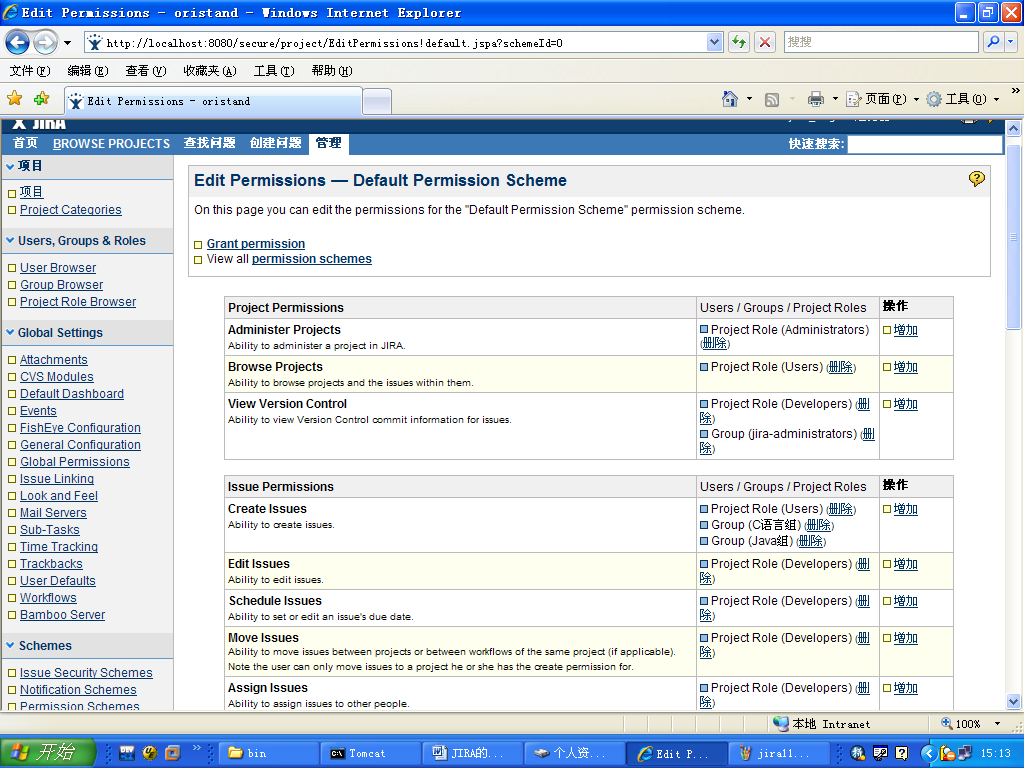


图 管理Permissions

1.7 工作流（WorkFlows）

JIRA中有默认的工作流，这个很容易就能理解。

当一个问题被创建时，问题就处于open状态，在有人解决的时处于In Progress状态，解决后处于Resloved，最后处于Closed状态，当然问题如果解决的不好，还可以重新开启。

当然我们也可以自己定义项目的流，在项目的Workflow Scheme后面的编辑中创建。

Open

In progress

Resolved

Close

Reopend

图 问题workflow

2 Users、Groups&Roles管理（Administrator）

2.1 人员管理

第一步：点击User Browser中的Add User进行链接；

第二步：输入相关的信息，进行保存。

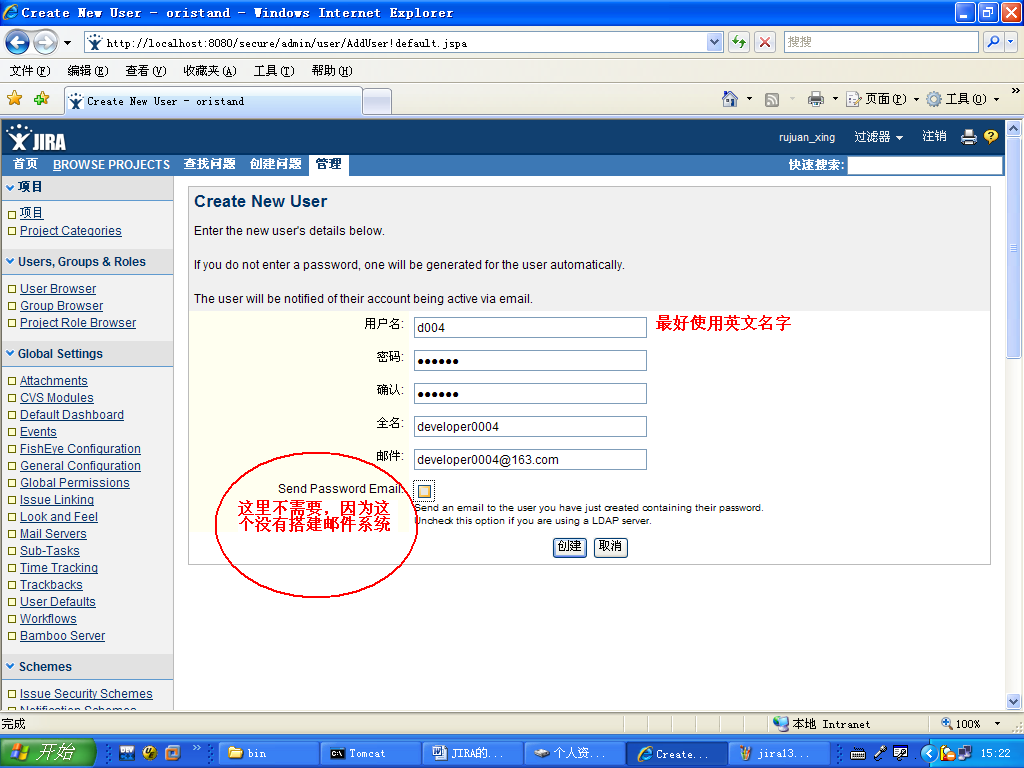


图 创建新User

完成后进入的界面，在这里可以分配所属的组以及工程权限：



图 新User编辑界面

2.2 组管理

在公司我们经常会看到人力资源部、市场部、采购部等，这里的组就相当于公司的部门，或者更小的一个部门的小组。每个组的成员拥有的权限是一致的，这样就方便管理。

第一步：点击Group Browser中的Add Group进行链接；

第二步：输入相关的信息，进行保存。



图 组管理界面

2.3 工程角色管理

这里创建的角色在所有的项目中都会起作用。

第一步：点击Project Browser中的Add Project Role进行链接；

第二步：输入相关的信息，进行保存。

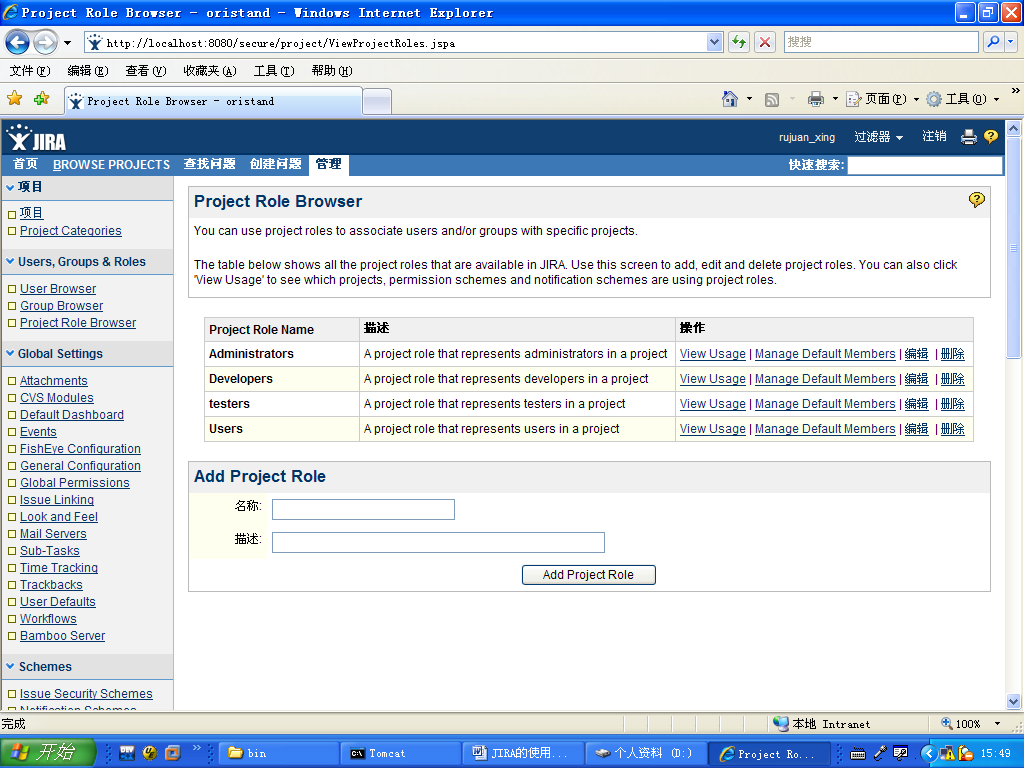


图 工程角色管理界面

3 创建问题（Users）

系统默认添加的使用者都是在jira-users组，这个组的成员都有创建问题的权限。

第一步：选择项目和问题类型



图 创建问题第一步

第二步：填写问题的详细情况



图 创建问题第二步

第三步：创建和的状态



图 创建问题完成

4 查找问题（Users）

这里主要涉及过滤器的使用。在一个开发团队中，每个人的任务是不一样的，一个问题的提出不是每个人都需要去解决，所以过滤器可以帮助我们更有效的定位到需要解决的问题。每个User可以根据自己的需要创建多个过滤器。同时JIRA也提供了很多个索引供我们选择。

创建一个过滤器，一共有四项可供选择，分别为：项目&问题属性、文本搜索、问题属性、日期时间。



图 编辑过滤器

5 跟踪解决问题

当一个问题被创建后，如果指定了开发者，就会直接提交到开发者手中去解决，如果没有指定则直接交到项目管理人手中，当然此时项目管理人有移动问题的权限，直接交给开发者去处理。

当开发者在处理问题时，就可以开始进行，问题的状态就会自动修改成In Progress，这时问题的创建者也会看到问题正在被解决中；当问题已经被解决了，开发者可以把问题设置成Fixed的。



图问题解决进行中



图问题已经解决

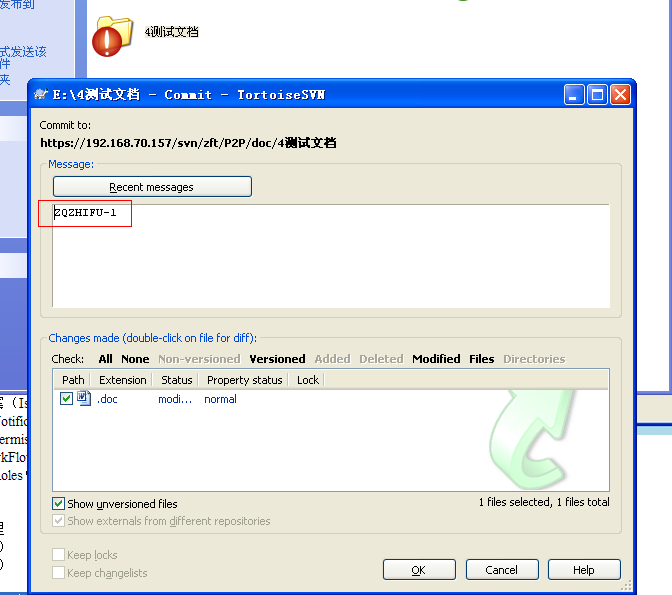
注意：问题已经解决了，但是解决问题的人没有权限关闭，只有问题的提出在才有关闭问题的权限。

6 解决问题与SVN变更

当开发者在处理问题时，通常会修改代码，或是做需求评审，变更需求。这时就要向SVN提交修改后的代码，或是文档。JIRA中的跟踪目标将会与SVN关联，记录与跟踪目标相关的SVN变更记录，方便知道，跟踪目标的变化情况。也方便以后出现类似问题的定位解决。

关联方法如下：

一、在向SVN提交变更时，将“关键字”，也就是我们通常理解的问题单号复制进图中所示红框位置。

图 变更提交

等待5钟后，我们会看到在该问题单的**Subversion Commits中有变更记录，具体记录到变更文件，及文件路径。**

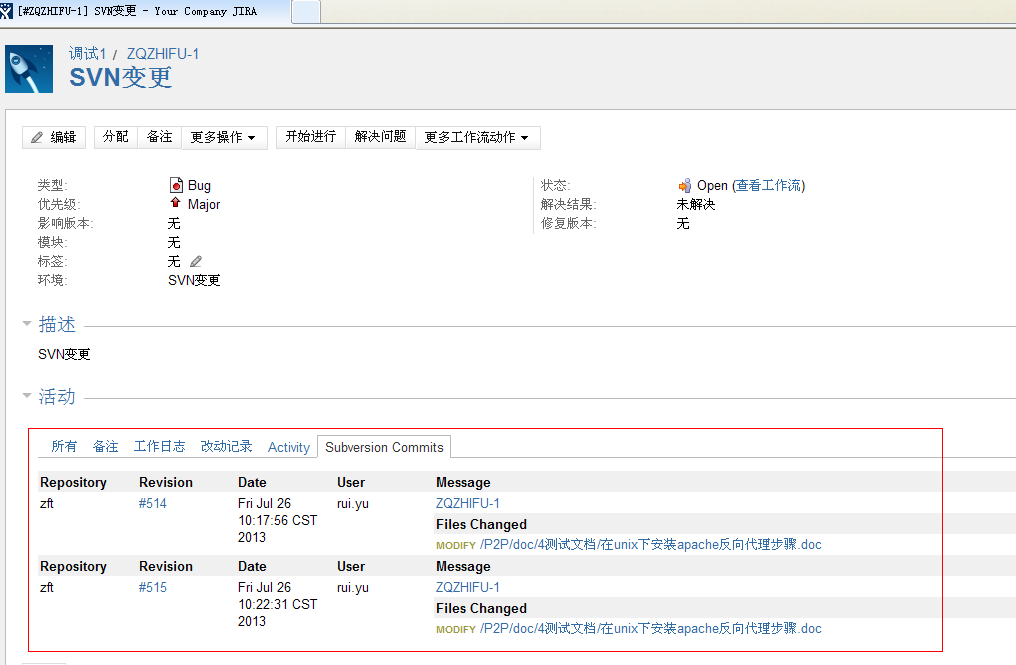


图 JIRA中的SVN变更记录

# 说明

本文档只具有基础的指导意义，是引用网上资料，结合我们所用版本的情况，增删编辑而成，部分图文与我们实际应用的服务器可能不相符合，但是已将90%以上的操作界面汉化，结合文档，应该能很快了解、使用。JIRA的其他方面与研发的日常使用无关，可以不必了解。如需进一步了解，可登陆[JIRA 5.0 新特性 - Atlassian产品资料中心 - Confluence中文演示平台 - 安迈无限 Confluence中国代理 Confluence技术服务](http://www.confluence.cn/pages/viewpage.action?pageId=2163615)