

用JIRA、CVS、XPlanner、WIKI来进行项目管理 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/65/2021_2022__E7_94_A8JIRA_E3_80_81_c41_65481.htm JIRA，一个非常出色的Issue跟踪系统，这里的Issue不单单是指BUG，很多时候也可以是TASK，IMPROVEMENT，NEW FEATURE，甚至是一个QUESTION。在多年前，我曾经尝试使用过那个经典的Bugzilla，但是一个项目作下来，大家都反映那个东西的界面实在是太粗糙，简直无法忍受而且报表功能也是在太弱。最后大家就讨论自己作一个BUG的跟踪系统，就在大家已经完成了设计文档准备编码的时候，我们发现JIRA原来就是我们要找的东西，而且比我们要的更多。它内置一个可以配置的工作流引擎(osworkflow)，一个快捷的全文检索功能(基于Apache Lucene)和一个可以配置的Dashboard(portlet)，以及一个和CVS连接的引擎，通过这个连接，在一个Issue中直接可以看到修改的文件名称，如果配置了viewcvs的话，还直接直接定位到行，根据一个问题可以跟踪到代码的行，这正式我们梦寐以求的功能。也正是这种特性，才使我们能够把一个个Issue当作发布和版本管理的一个单元。CVS，这个应该大家都知道。在系统开发过程中，一切的源代码和设计文档都应该进入版本管理系统来进行管理，有的时候可能资源库可能会膨胀的很大，但这个代价是值得的。XPlanner,在整个管理体系中，进度管理一直是一个比较薄弱的环节，我也曾试过dotproject这样的管理软件，但由于dotproject管理的太过详细，填报起来太复杂，大家渐渐都失去了填报的热情。这个XPlanner软件可就简单多了。指定了迭代，story，然

后就可以填写进度了。由于这个软件也是OpenSource的，所以如果觉得不满意，修改起来也很方便，现在老林就对这个系统作了些改进，可以直接和JIRA系统连接起来，JIRA中建立issue后，可以在XPlanner中反映出来，连填写story的时间都省去了，然后在下班之前可以生成一个详细的报告，列出每个人在这一天内在自己负责的Issue在上的处理时间和进度。

WIKI，在项目管理中，我们一直把它当作文档管理和Portlet系统来使用，它现在已经变成我们的小组的工作台，在WIKI中我们制定了包括系统开发设计规范在内的一切设计文档，以及数十个经常的HOWTO项目，例如如何配额一个标准的开发环境，如何使用CVS客户端，如何使用JIRA，以及自己的JavaDoc，JSDoc等。我们也可以通过Wiki来简单的整合系统，在Wiki中我们列出了所有开发环境和开发工具的入口，例如上面就放了进入JIRA，XPlanner以及我们各个Project的连接，甚至到Apache中常用的Project的JavaDoc的连接，现在再也没有人去记录这些URL了，只要打开Wiki所有的资源都在面前了，并且由于wiki本身的开放性，所以每个团队的成员都是一个维护者，同时也是这个系统的受益者。

在很多的团队中经常出现的情况是一个小子对某个技术特别在行，大家遇到这方面的问题都问他，在小的团队中，面对面的交流通常是最快的交流方式，但是放到大的团队中，这个就不大可行了，那个小子迟早有一天会被问的烦到吐血为至，特别是他自己的工作也无法按时完工的时候。还是抽一个小时写出来，放到wiki里面吧，别问我，自己去查Wiki。

基于ISSUE的发布管理。从版本管理的角度来考虑，最理想的发布方法就是把CVS中的代码拿下来，打上一个tag，编

译并且测试一直到发布。这样的管理方式的确是很简单的，但事实上用户可不买帐的，用户觉得在新的版本中某个新的功能他还不想要，这可能是他还没有整理好业务初始数据或者在实际的业务流程上或人员上没有做好准备，上帝说了不要咱就不能把这个新功能发布。在这个情况下，基于Issue的发布管理是一个好的方案。这里讲的Issue就是前面JIRA系统中的一个issue。通常每个Issue的完成都会伴随这一些代码的修改。基于Issue的发布简单的来说就是把一组Issue变更的文件用patch的形式发布到正式的系统。基于Issue发布的前提就是要在Issue和Source之间建立连接，使发布人员清楚的知道每个Issue修改的源代码是什么。我们实践下来最简单的办法就是在提交source的时候必须加上JIRA编号，没有JIRA编号代码是不能提交的。这样有以下好处。1) 防止一些没有经验的程序员无意义的提交，比如一个小子今天提交了一个java文件，明天发现这个变量命名有点不爽，修改后就要提交，在这种情况下，这个提交是没有意义的，如果测试组已经测试这个Issue，是否测试组要重新测试？为一个变量名称化这样的时间和冒险是可嫩的。小伙子还是在第一次提交的时候就把变量名想好了再提交。2) 程序员偷偷的修改代码，一个小伙子发现自己的已经Closed的Issue中有一个Bug，便偷偷的修改代码。这个当然也是不可能的，凡是提交到CVS中的代码就不是自己的了，那是大家的，没有足够的理由想改当然没有那么容易。先自己建立建立个Issue，向Team leader报告，然后再去修改代码。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com