

## 问题管理软件 JIRA 和 URTracker 对比评测报告

URTracker 和 Jira 都是两个比较强大的问题跟踪（Issue Tracking，也可以叫事务跟踪）软件。相对于 Bugzilla、Bugfree 等单一的 bug 跟踪软件，他们具有更全面的权限控制功能、更好的灵活性和定制能力，能适用于 bug 跟踪之外的其他类型 issue tracking 应用。

我们将对两者的基本功能做一些对比，方便大家了解和使用。比较版本为 URTracker3.3 版高级版和 Jira3.13 企业版。

名词：

Issue：译为“问题”或“事务”，表示各种类型的问题、bug、任务、需求等需要跟踪和管理的对象。

Status：状态，表示 Issue 所处的阶段。

Transition：状态的转变，表示 Issue 从一个状态转入另一个状态。

Assignee：“待办人”，表示 Issue 的当前负责人，Issue 被提交给的人。

图例：表示双方都有的功能，或者各有特色的功能。或表示一方有一方没有，或一方优于另一方的功能。

### (1)基础内容比较

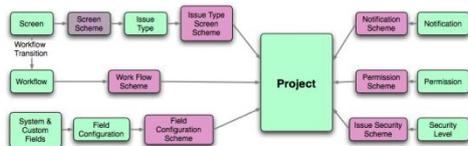
比较内容	Jira		URTracker
产地	澳大利亚（国内有代理）。		中国，北京
主页	<a href="http://www.atlassian.com">www.atlassian.com</a>		<a href="http://URTracker.cn">URTracker.cn</a>
版本划分	分为 3 个版本：标准版、专业版和企业版。 标准版功能限制较多，不支持流程定制等特性。专业版允许定制流程，但仅支持一个流程，所有项目必须使用相同的流程。要想将 Jira 用于超过一种用途，就 <b>只能使用企业版</b> 。		也分 3 个版本：免费版、标准版和高级版。 三个版本功能差别相对较小，都支持自定义多个流程和字段，都支持知识库等主要功能功能。
使用方式	通过 Web 浏览器		通过 Web 浏览器
技术平台	Java 可以运行在支持 Java 的操作系统，如 Windows、Linux 等。 <b>优点：</b> 可以跨平台。		Asp.Net 运行在 windows 环境中。 <b>优点：</b> 普通技术人员对 windows 平台比较熟悉，.net，IIS 等组件都是微软的产品，集成度较好，各种参数配置起来比较方便（如端口、域名绑定、应用程序池参数等）。

中文支持	<p>Jira 支持多语言，但中文化做的比较差，很多地方中英文混杂显示。内部似乎不是 Unicode 编码，造成了某些地方对中文支持不完善（如 Workflow 名称等规定不能用中文文字），有时会有乱码情况发生。</p> 		<p>URTracker 是中文软件（3.2 版本中支持中、简体、繁体的切换），界面全中文。内容的输入和显示使用 UTF8 编码，支持任意语言同时输入与显示，不会有乱码。</p>
应用范围	<p>主要应用于： BUG 跟踪 需求跟踪 任务跟踪</p>		<p>BUG 跟踪 需求跟踪 任务跟踪 用例管理 任务分解 WBS、计划和项目管理 IT 服务管理（事件管理、问题管理、变更管理等） 客户服务管理（问题咨询、服务请求处理跟踪、客户资料管理等） 各类日常办公流程 各类申请、审核流程 知识库管理</p>
<p>问题类型 Issue Type 的处理方式</p>	<p>一个项目可以创建多种类型的 issue。</p> <p><b>优点：</b> + 相关的内容可以在同一个项目里管理。</p> <p><b>缺点：</b> - 关联配置较多（Field Configurations、Field Configuration Schemes、Screens、Screen Schemes、Issue Type Screen Schemes 等），配置起来比较复杂。 - 多个项目共享相同的 Issue 类型及相关的设置，单独调整受到限制。 - 每个人在同一个项目中能够创建的 Issue 的类型是相同的，无法控制某些人能创建 Bug 但不允许创建 Task。 - 项目成员对项目各类型 Issue 的编辑、移动、分配等操作权限是相同的，无法单独控制。 - 多种类型的 Issue 使用相同的编码前缀。无法从 Issue 的编码了解 Issue 的类型。 - 创建 Issue 时，要多一个选择 Issue 类型的步骤。</p>		<p>每个项目一种 Issue 类型。 创建多个项目管理字段和流程不同的多种类型的 Issue。然后把相关的项目放入同一个项目目录下，方便查看和相互跳转切换。</p> <p><b>优点：</b> + 由于问题类型同人员角色、字段定义、流程定义具有非常高的相关性，把每种 Issue 放在一个单独的项目里，更符合一般的需求情况。 + 简化了人员分组、流程、字段定义的复杂程度。可以实现更细微和灵活的控制。 + 实现对不同类型 Issue 的创建、编辑等操作权限的控制。 + 不同类型的 Issue 拥有不同的编码前缀。通过 Issue 编码即可判断其类型。</p> <p><b>缺点：</b> - 可能要创建多个项目。</p>

业务自定义功能的实现思想

在 Jira 软件中，大部分的配置都是全局的。用户需要配置大量的 Scheme（如：Issue type scheme、Notification Scheme、Permission Scheme、Issue Security Scheme、Field Configuration Scheme、Screen Scheme、Issue Type Screen Scheme、Workflow Scheme 等）。然后在创建项目时选择需要使用的各个 scheme。

JIRA Enterprise Entity Scheme Relationship Diagram v.1



**优点:**

+ 多个项目可以使用相同的 Scheme 配置，从而减少重复的配置劳动。

**缺点:**

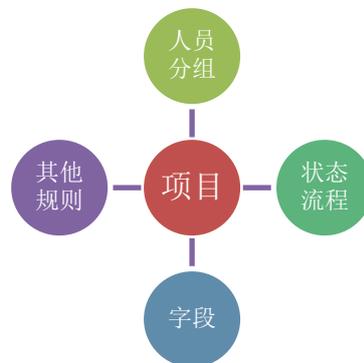
- 概念太多，相互之间关系比较复杂，理解起来比较费力，有很陡峭的学习曲线。
- 配置起来很麻烦，要操作的地方比较多。
- 配置不直接，从一个具体目标要倒推每个 Scheme 的配置。
- 基本上所有的配置都需要系统管理员配置好，项目本身的管理人员可以调整的范围很小。
- 如果各个项目有不同的需求，则要创建不同的 Scheme。scheme 数量增多，会带来管理上的困难，导致更多的出错机会。

配置管理模式

因为绝大多数的配置，如字段定义、工作流等，都是全局的，可能被多个项目共享，所以基本上所有的管理操作都要由系统管理员来完成。项目中的管理人员可以调整的内容很少。



URTracker 中，每个项目的配置都是独立的。没有全局 Scheme 的概念，只需要直接配置当前项目所需的人员分组、权限、流程、字段等参数即可。



**优点:**

- + 大大减少了要配置的内容，简化了配置操作，同时又增加了灵活性。
- + 一个项目中的配置不会影响到其他项目，所以可以任意设置。
- + 可以将配置项目的权限赋予项目中的某个工作组。有利于将权限下放，减少系统管理员的工作。使用人员了解项目参数的配置方式后，可以将软件应用于更多类型 issue 类型的管理和更广的范围。

**缺点:**

- 每个项目须单独配置。（可以通过在创建新项目时从现有项目或配置文件中复制配置，避免重复的配置操作）

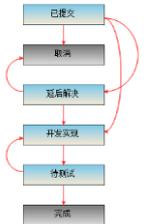


因为每个项目的配置参数是独立的，所以一方面可以由系统管理员配置项目，另一方面也可以允许某些项目成员配置和管理项目。

**优点:**

有利于使用软件的人随时根据需要调整项目配置。

## (2) workflow 功能比较

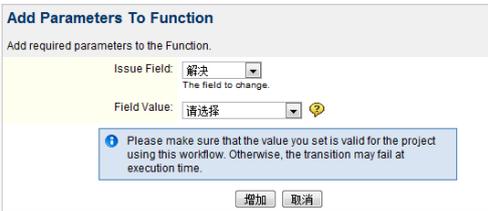
比较内容	Jira		URTracker
配置过程	<ol style="list-style-type: none"> <li>1. 定义问题“状态” (Issue Status)</li> <li>2. 创建 Workflow。</li> <li>3. 定义一组步骤 (Step)，每个 Step 绑定一个状态。</li> <li>4. 定义步骤之间的“转变” (Transition)。</li> <li>5. 对每种需要编辑 Issue 字段的情况，创建 Screen，并绑定到各 Transition。</li> <li>6. 对每种需要特定邮件通知规则的情况，定义 Events，并绑定到各 Transition。</li> <li>7. 定义 Notification Scheme，绑定 Event 所对应的通知规则。</li> <li>8. 定义 Permission Scheme，控制各角色的权限。</li> <li>9. 定义 Workflow Scheme，指定每个 Issue Type 所对应的 Workflow。</li> <li>10. 创建项目，并选择对应的 Workflow Scheme</li> </ol> <p><b>优点:</b></p> <ul style="list-style-type: none"> <li>+  workflow 定义是全局的。多个相同情况的项目或 Issue 类型可以共享配置。</li> </ul> <p><b>缺点:</b></p> <ul style="list-style-type: none"> <li>- 增加了配置复杂程度，不直接。特别是在流程较复杂的情况下，配置起来比较困难。</li> <li>- 希望单独调整某个项目的流程时，需要再创建一套 Workflow。</li> </ul>	👍	<ol style="list-style-type: none"> <li>1. 定义问题“状态” (Issue Status) 并指定各项参数。</li> <li>2. 定义状态之间的“转变步骤” (Transition) 并指定各项参数。</li> <li>3. 定义“初始状态”规则。</li> </ol> <p>和 Jira 中的“步骤”含义不同，URTracker 中的“步骤”表示状态的转变 (Transition)。</p>  <p><b>优点:</b></p> <ul style="list-style-type: none"> <li>+ 每个项目的流程定义是独立的，互不影响。</li> <li>+ 大大简化了配置过程，降低了复杂度，增加了灵活性。</li> </ul> <p><b>缺点:</b></p> <p>每个项目需要单独配置。(可以通过从其它项目或模板复制配置的方式避免这个问题。)</p>
功能限制	<p>标准版：不支持定义 workflow；</p> <p>专业版：只能使用唯一的工作流设置，但是允许修改；</p> <p>企业版：可以定义多个 workflow；</p>	👍	<p>标准版和高级版都支持任意数量 workflow 设置。免费版支持 5 个。</p>
显示流程图	不支持。	👍	<p>支持。配置流程时可以直观地预览。</p> 

<p>创建 Issue 时的初始状态</p>	<p>初始状态是固定的（根据 Issue Type 所对应的 Workflow 确定），没有可以选择的地方。</p>		<p>可以对不同的工作组设置不同的允许使用的初始状态的集合。</p>  <p><b>优点:</b></p> <ul style="list-style-type: none"> <li>+ 可以用于控制不同的人从流程的不同阶段开始执行。如：“测试员”提交 bug 时，只能提交给项目经理“待分配”。而项目经理提交 bug 时，可以直接选择负责处理 bug 的开发人员“待解决”。</li> <li>+ 用于实现在同一个项目中使用多个子流程。不同的初始状态作为不同子流程入口起点。</li> </ul>
<p>流程的运行方式</p>	<p>在某个状态下，有权限的人执行某个从本状态开始的 Transition，将 Issue 更新到新的状态。</p> <p>执行 Transition 的人不一定是待办人，也可能是其他被允许的人。</p> <p><b>优点:</b></p> <ul style="list-style-type: none"> <li>+ 比较灵活。可以允许待办人以外的人处理 Issue。</li> </ul> <p><b>缺点:</b></p> <ul style="list-style-type: none"> <li>- 每个 Transition 都需要单独配置。</li> <li>- 如果有多个人同时能够处理 Issue，会引起责任不明确，流程不容易控制的问题（比如 Assignee 还没处理完问题，状态就被别人改掉了）。</li> </ul>		<p>在某个状态下，Issue 的当前待办人（Assignee）执行某个有权限的 Transition，将 Issue 更新到新的状态，并根据流程设定自动或手动将 Issue 提交给新的待办人（也可能是自己）。（此过程被称作“处理”Issue。）</p> <p><b>优点:</b></p> <ul style="list-style-type: none"> <li>+ 流程执行模式明确（Issue 待办人处理 Issue 将其更新到新的状态和新的待办人，新的待办人再处理 Issue，直至 Issue 关闭。其他人不能处理 Issue，只能通过工作记录、评论等功能参与 Issue 处理）。</li> <li>+ 不需要配置。</li> </ul>
<p>一个项目中实现多个流程</p>	<p>通过不同的 Issue Type 实现。</p>		<p>通过不同的初始状态实现。</p>
<p>流程非正常跳转</p>	<p>不支持，必须按照流程执行。</p>		<p>提供了“重分配”功能，允许具有相应管理权限的工作组的成员跳转流程和重分配问题。很多特殊情况会用到此功能。</p> 

<p>“待办人” (Assignee) 的指定方式</p>	<p>(1) 默认：“提交给 Project Lead”。  (2) “不提交给任何人”。  (3) 自动提交给 Component 的负责人。  (4) 从在 Permission Scheme 中设定的具有 Assignable User 权限的人中选择。</p>  <p><b>缺点:</b></p> <ul style="list-style-type: none"> <li>- 模式比较固定，不太适合其他类型的 issue 跟踪。</li> <li>- 不能根据步骤限定可选的待办人范围。(比如在 bug 修复后，只需要提交给某个测试人员。)</li> <li>- 在流程的中间过程，不支持自动提交给指定的人。</li> <li>- 可选 Assignee 较多时，不支持用搜索的方式快速找到要选择的人。</li> <li>- 将 Assignee 的中文译为“开发者”，似乎不大合适。根据流程的不同，测试员也有可能作为 Assignee。</li> </ul>		<p>根据目标状态或步骤的不同设置不同的分配规则：</p> <ul style="list-style-type: none"> <li>● 处理人从指定的项目成员或工作组中选择待办人（如：项目经理分配任务时，选取一个开发人员）</li> <li>● 自动提交给某个固定的人或工作组（如提交审核时，自动分配给“项目经理”）</li> <li>● 自动提交给事务创建人(如任务完成后，提交给创建人确认)</li> <li>● 自动提交给上一个处理人(如审核失败，需要重新提交)</li> <li>● 从历史处理人中选择</li> </ul> <p>URTracker 还支持通过部门、工作组等条件进一步过滤 assignee 的可选范围。  (如某个申请单 Issue 需要提交给“部门经理”审核，可以通过此功能让每个部门的人只能提交给自己部门的经理)</p> <p><b>优点:</b></p> <ul style="list-style-type: none"> <li>+ 符合流程设置需求;</li> <li>+ 适用于各种类型的 issue;</li> </ul>
<p>自动提交给模块负责人</p>	<p>支持。  模块只支持一个层次。</p>		<p>支持。  模块可以多层，子模块可以继承父模块的负责人。</p>
<p>提交给组</p>	<p>不支持</p>		<p>通过分配规则的设置，允许自动或手动将 issue 提交给某个工作组处理。组里的每个人在“待办 Issue 列表”中都可以看到这个问题。然后，组中的某个合适的人员可以“领取”并处理该问题。</p> 
<p>匿名创建</p>	<p>不支持</p>		<p>可以将创建人自动替换为某个虚拟账号。  (适用于匿名投诉、意见等 Issue 类型)</p>

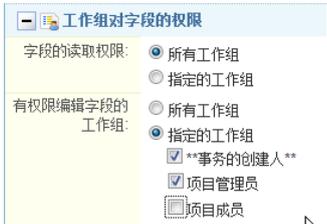
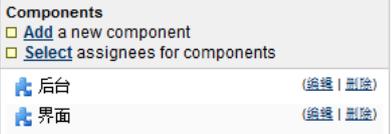
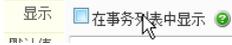
<p>提交给多个人</p>	<p>不支持</p>		<p>支持“协处理”功能，在指定 Assignee 的同时，指定“协办人”。协办人可以提交处理信息，但不能更改 issue 的状态。</p> 																								
<p>关闭问题</p>	<p>通过 Resolution 字段是否有值判断问题是否关闭。</p>  <p>实际配置中，必须通过手动方式或用 Transition 的 Post Function 功能设置 Resolution 字段的值来关闭事务。感觉有些多余，有点难以理解。</p>		<p>用 Issue 所处的状态来表示问题是否关闭。这种方式更直接，更易于理解和配置。</p> 																								
<p>自动邮件通知</p>	<p>支持。</p> <p>(1) 创建一系列 Event</p> <p>(2) 将 Event 绑定到工作流的 Transition</p> <p>(3) 在 Notification Scheme 中设定每个 Event 所对应的通知规则。</p> <table border="1" data-bbox="351 1198 837 1512"> <thead> <tr> <th>Event</th> <th>Notifications</th> <th>操作</th> </tr> </thead> <tbody> <tr> <td>Issue Created (System)</td> <td><input type="checkbox"/> Current Assignee (添加) <input type="checkbox"/> Reporter (添加) <input type="checkbox"/> All Watchers (添加)</td> <td><input type="checkbox"/> 添加</td> </tr> <tr> <td>Issue Updated (System)</td> <td><input type="checkbox"/> Current Assignee (添加) <input type="checkbox"/> Reporter (添加) <input type="checkbox"/> All Watchers (添加)</td> <td><input type="checkbox"/> 添加</td> </tr> <tr> <td>Issue Assigned (System)</td> <td><input type="checkbox"/> Current Assignee (添加) <input type="checkbox"/> Reporter (添加) <input type="checkbox"/> All Watchers (添加)</td> <td><input type="checkbox"/> 添加</td> </tr> <tr> <td>Issue Resolved (System)</td> <td><input type="checkbox"/> Current Assignee (添加) <input type="checkbox"/> Reporter (添加) <input type="checkbox"/> All Watchers (添加)</td> <td><input type="checkbox"/> 添加</td> </tr> <tr> <td>Issue Closed (System)</td> <td><input type="checkbox"/> Current Assignee (添加) <input type="checkbox"/> Reporter (添加) <input type="checkbox"/> All Watchers (添加)</td> <td><input type="checkbox"/> 添加</td> </tr> <tr> <td>Issue Commented (System)</td> <td><input type="checkbox"/> Current Assignee (添加) <input type="checkbox"/> Reporter (添加) <input type="checkbox"/> All Watchers (添加)</td> <td><input type="checkbox"/> 添加</td> </tr> <tr> <td>Issue Comment Edited (System)</td> <td><input type="checkbox"/> Current Assignee (添加) <input type="checkbox"/> Reporter (添加)</td> <td><input type="checkbox"/> 添加</td> </tr> </tbody> </table>	Event	Notifications	操作	Issue Created (System)	<input type="checkbox"/> Current Assignee (添加) <input type="checkbox"/> Reporter (添加) <input type="checkbox"/> All Watchers (添加)	<input type="checkbox"/> 添加	Issue Updated (System)	<input type="checkbox"/> Current Assignee (添加) <input type="checkbox"/> Reporter (添加) <input type="checkbox"/> All Watchers (添加)	<input type="checkbox"/> 添加	Issue Assigned (System)	<input type="checkbox"/> Current Assignee (添加) <input type="checkbox"/> Reporter (添加) <input type="checkbox"/> All Watchers (添加)	<input type="checkbox"/> 添加	Issue Resolved (System)	<input type="checkbox"/> Current Assignee (添加) <input type="checkbox"/> Reporter (添加) <input type="checkbox"/> All Watchers (添加)	<input type="checkbox"/> 添加	Issue Closed (System)	<input type="checkbox"/> Current Assignee (添加) <input type="checkbox"/> Reporter (添加) <input type="checkbox"/> All Watchers (添加)	<input type="checkbox"/> 添加	Issue Commented (System)	<input type="checkbox"/> Current Assignee (添加) <input type="checkbox"/> Reporter (添加) <input type="checkbox"/> All Watchers (添加)	<input type="checkbox"/> 添加	Issue Comment Edited (System)	<input type="checkbox"/> Current Assignee (添加) <input type="checkbox"/> Reporter (添加)	<input type="checkbox"/> 添加		<p>支持。</p> <p>直接根据目标状态或步骤设置“自动通知规则”。</p>  <p><b>优点:</b></p> <p>+ 设置很简单，直接</p>
Event	Notifications	操作																									
Issue Created (System)	<input type="checkbox"/> Current Assignee (添加) <input type="checkbox"/> Reporter (添加) <input type="checkbox"/> All Watchers (添加)	<input type="checkbox"/> 添加																									
Issue Updated (System)	<input type="checkbox"/> Current Assignee (添加) <input type="checkbox"/> Reporter (添加) <input type="checkbox"/> All Watchers (添加)	<input type="checkbox"/> 添加																									
Issue Assigned (System)	<input type="checkbox"/> Current Assignee (添加) <input type="checkbox"/> Reporter (添加) <input type="checkbox"/> All Watchers (添加)	<input type="checkbox"/> 添加																									
Issue Resolved (System)	<input type="checkbox"/> Current Assignee (添加) <input type="checkbox"/> Reporter (添加) <input type="checkbox"/> All Watchers (添加)	<input type="checkbox"/> 添加																									
Issue Closed (System)	<input type="checkbox"/> Current Assignee (添加) <input type="checkbox"/> Reporter (添加) <input type="checkbox"/> All Watchers (添加)	<input type="checkbox"/> 添加																									
Issue Commented (System)	<input type="checkbox"/> Current Assignee (添加) <input type="checkbox"/> Reporter (添加) <input type="checkbox"/> All Watchers (添加)	<input type="checkbox"/> 添加																									
Issue Comment Edited (System)	<input type="checkbox"/> Current Assignee (添加) <input type="checkbox"/> Reporter (添加)	<input type="checkbox"/> 添加																									
<p>手动抄送邮件通知</p>	<p>不支持手动邮件通知抄送功能。</p>		<p>在创建、编辑、处理、评论、登记工作、重分配等功能处，提供手动选择通知抄送对象的功能。可以抄送给组、指定的用户，或任意的外部 Email 地址。</p> 																								

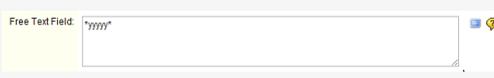
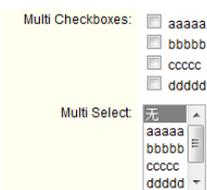
短信通知	不支持		支持。 类似于邮件通知。可以自动或手动通知。 可以通过模板设定短信通知的内容。
桌面助手通知	不支持		支持。 可以实时显示通知消息,减少邮件的数量 (桌面助手通知后就不发邮件了,通知不到才发邮件)。
工作用时记录	支持。 全局选项选择是否开启此功能。  <b>优点:</b> 带有“调整估算时间”功能。 <b>缺点:</b> 在创建或处理 Issue 时不支持记录用时。		支持。 在创建、处理或记录工作时输入。 每个项目可以控制是否启用。 每个状态有是否需要记录用时的选项。 
状态或步骤的处理时限 (控制流程的执行效率)	不支持		(1) 自动设定状态时限: 根据 issue 的优先级、严重级、分类等条件的组合,自动设置某个状态的停留时限。 (2) 手动设置状态时限: 由上一步的处理人设定下一状态的处理时限。
定时器功能 (控制流程的执行效率)	不支持		有强大的定时器功能,可以实现对符合条件的 issue 进行自动定时通知公告、自动转派等功能。
挂起与激活 Issue	不支持		支持。 挂起 issue 表示暂时不处理,不跟踪。 可以在设定的开始时间自动激活 issue。 常用于计划性质的 issue。
子任务与任务分解	要为子任务创建特殊的 Issue Type。 只支持一个级别的子任务。 		有很强大的 Issue 分解功能。 可以当前项目或其他项目创建子 Issue。 (比如一条客户需求可能在开发、美工、运维等部门分别产生子任务)。 可以任意层次的分解、调整子任务的顺序,实现 WBS 功能。 

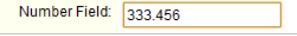
<p>自动设置字段 置值功能</p>	<p>支持 (通过 Transition 的 Post Functions 功能实现)。</p> 	<p> 支持。 支持宏定义, 允许将当前处理人、当前处理人所在部门、下一个待办人及所在部门、当前时间等记录到设定的字段中(方便查询统计)。 支持对数字字段进行加 1 操作, 从而实现 Transition 计数功能。(如统计 Issue 被 Reopen 的次数)。</p> 
------------------------	--	---

### (3) Issue 字段功能比较

比较内容	Jira		URTracker
<p>字段定义的有效范围</p>	<p>在全局范围内定义字段，然后通过各项配置将字段和具体的项目、问题类型以及处理过程联系起来。</p>  <p><b>优点:</b></p> <ul style="list-style-type: none"> <li>+ 多个 Issue 类型可以共享相同的字段配置。</li> </ul> <p><b>缺点:</b></p> <ul style="list-style-type: none"> <li>- 字段相关的配置很复杂，要配置的地方比较多。</li> <li>- 配置时需要格外小心，避免一个项目的字段在配置的时候影响到其他项目。</li> <li>- 只能有系统管理员可以更改字段设置，无法授权给项目内的管理人员。</li> </ul>	<p>👍</p>	<p>每个项目中字段定义是独立的。直接在项目中添加和配置本项目需要的字段。</p>  <p><b>优点:</b></p> <ul style="list-style-type: none"> <li>+ 项目之间的字段配置是隔离的，互相不会产生影响，可以根据需要随意配置。</li> <li>+ 字段配置简单直接。</li> <li>+ 在一个页面中即可修改字段的所有配置。</li> <li>+ 可以将字段配置权限下放给项目内的具有管理权限的工作组成员。</li> </ul>
<p>字段编辑控制</p> <p>(创建、编辑 Issue, 以及处理 Issue 的某个步骤, 所需要填写的字段集)</p>	<p>(1)创建一系列 Screen, 配置每个 Screen 允许编辑的字段。</p> <p>(2)将 Screen 绑定到 Screen Scheme, 实现创建、编辑、查看 Issue 时的字段控制。</p> <p>(3)将不同的 Screen 绑定到对应的 Workflow Transition 上, 实现 Issue 处理过程中的字段编辑控制。</p> <p>(4)绑定 Screen Scheme 和 Issue Type</p>  <p><b>缺点:</b></p> <ul style="list-style-type: none"> <li>- 较复杂。</li> <li>- 需要创建和配置较多的 Screen。</li> <li>- 无法控制字段在各个 Screen 是否必填等选项。</li> <li>- 无法根据操作人权限的不同在各个 Screen 中隐藏或显示字段。</li> </ul>	<p>👍</p>	<p>直接根据源状态、目标状态或状态转变步骤配置字段编辑规则。</p>  <p><b>优点:</b></p> <ul style="list-style-type: none"> <li>+ 配置比较简单直接，不需要创建 Screen 等配置，只需要点鼠标选择即可。</li> <li>+ 控制方式灵活，可以根据目标状态、源状态、或具体的状态转变步骤进行控制。</li> <li>+ 可以实现在任意步骤时字段是否允许编辑、是否必填的控制。</li> <li>+ 可以结合工作组的权限设置，根据处理人的不同，自动隐藏没有编辑权限的字段。</li> </ul>

字段读写权限控制	不支持		可以每个字段设置各个工作组对其读取和编辑的权限，从而在创建、编辑、处理、显示 issue 时，自动隐藏相应的内容。 												
系统内置字段	概要、问题类型、安全级别、优先级、逾期日期、模块、影响版本、修复版本、开发者、环境、概述、附件等。		摘要、模块、优先级、严重级、类型、期限、开始时间												
隐藏某个内置字段	设置各个相关的 Screen 实现字段隐藏。		设置“是否启用该系统字段”的选项。 												
为内置字段改名	不支持。		可以为同一个内置字段在不同的项目中设置不同的名称(从而用于不同的用途)。												
更改内置字段的录入控制	不支持更改。		可以任意修改。(和自定义字段一样。)												
模块 Components 分级(创建子模块)	不支持分级，只有 1 级。 		支持任意分级，子模块可以继承父模块的负责人。 												
将字段加入列表视图	设置 Navigator Columns 实现。 <b>重新排列</b> 下面的表格使用假设的数据来举例。你的问题导航栏如果使用选中用  和  来重新设置列的顺序。用  从列表中移除一个列。 <table border="1" data-bbox="359 1500 694 1556"><thead><tr><th>T</th><th>Key</th><th>概要</th><th>开发者</th><th>报告人</th><th>优先</th></tr></thead><tbody><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr></tbody></table> <b>缺点:</b> <ul style="list-style-type: none"><li>- 是全局的设置，所有项目的字段在一起。无法对单独的项目进行单独的控制。</li><li>- 字段多时比较乱，难以管理。</li><li>- 只有系统管理员可以操作。</li></ul>	T	Key	概要	开发者	报告人	优先								在字段选项中选中“在列表视图中显示”即可。 
T	Key	概要	开发者	报告人	优先										
															
字段排列顺序	在每个 Screen 中单独配置。 <b>优点:</b> 可以单独控制。 <b>缺点:</b> 需要分别设置。Screen 多时，比较麻烦。		在项目中统一调整字段顺序。 <b>优点:</b> 设置一次即可。												

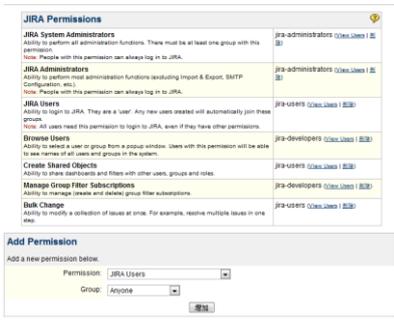
<p>自定义字段类型：单行文本</p>	<p>支持。 无特殊的控制功能。</p> 	<p>支持。</p> <p>+ 允许设定控件宽度</p> <p>+ 允许设定最大输入长度</p> <p>+ 支持使用正则表达式验证输入内容的合法性</p>  <p>+ 支持格式化输出</p> <p>+ 允许根据输入的值查找具有相同值的 Issue (避免创建重复的条目)</p>  <p>+ 允许使用字段的值过滤 Issue 列表或参与统计 (类似于通过 Issue 状态字段)</p> <p>+ 允许从以前输入的值中选择。</p> 
<p>自定义字段类型：长文本字段</p>	<p>支持 (名称为“Free Text Field”)</p> <p><b>缺点:</b></p> <ul style="list-style-type: none"> <li>- 仅支持纯文本方式输入。</li> <li>- 可以通过在 Field Confirmation 配置中选择 Wiki Style Renderer, 允许通过 wiki 语法输入带格式的内容, 但是不直观。</li> </ul>  <ul style="list-style-type: none"> <li>- 不支持调整控件外观等输入属性</li> </ul>	<p>支持 (名称为“多行文本字段”)</p>  <p>可选两种输入方式:</p> <ol style="list-style-type: none"> <li>(1) 纯文本</li> <li>(2) 格式化文本 (支持设置字体颜色或大小、绘制表格、上传图片等功能)</li> </ol> <p><b>优点:</b></p> <p>+ 允许设定输入框的高度和宽度。</p>
<p>自定义字段类型：多项选择</p>	<p>分两种类型:</p> <ol style="list-style-type: none"> <li>(1) Multi CheckBox 多选框</li> <li>(2) Multi Select 多选列表</li> </ol>  <p><b>缺点:</b></p> <ul style="list-style-type: none"> <li>- 两种字段不能转换。</li> </ul>	<p>一种字段类型支持两种输入方式, 实现 Jira 两种字段的功能。</p>  <p><b>优点:</b></p> <ul style="list-style-type: none"> <li>+ 随时切换两种输入方式。</li> <li>+ 可选值可以直接指定, 也可以使用全局定义的选项列表。</li> <li>+ 使用 ListBox 时, 可以控制列表框显示的宽度和高度。</li> <li>+ 使用 CheckBoxList 时, 选项横排显示, 节省空间。</li> </ul>

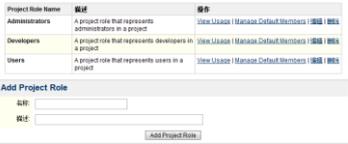
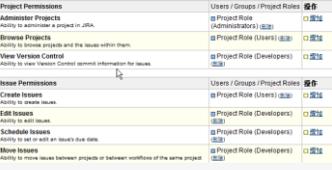
<p>自定义字段类型：单项选择</p>	<p>分两种类型：                  (1) Radio Buttons                  (2) Select List</p>  <p><b>缺点：</b>                  - 两种字段不能转换。</p>	<p>👍 一种字段两种输入方式，实现 Jira 两种字段的功能。</p>  <p><b>优点：</b>                  + 随时切换两种输入方式。                  + 可选值可以直接指定，也可以使用全局定义的选项列表。                  + 使用 Radio Button List 时，选项横排显示，节省空间。                  + 使用下拉框时，可以调整宽度。                  + 允许根据值过滤 Issue 列表                  + 允许进行统计(如统计每个值出现的次数)</p>
<p>自定义字段类型：时间日期</p>	<p>分两种类型：                  (1) Date Picker 类型：可选择日期                  (2) Date Timer 类型：可选择日期+小时和分钟                  默认值选项：指定具体的时间当前时间（操作时）。</p>  <p><b>缺点：</b>                  - 已定义好的时间日期字段不能改变时间精度（因为不能改变字段的类型）                  - 不支持精确到月份、日期+小时、日期+小时+分+秒等精确度。</p>	<p>👍 一种字段支持多种精度（月份；日期；日期+小时；日期+小时+分钟；日期+小时+分钟+秒）。实现比 Jira 两种日期字段更多的功能。</p>  <p>默认值选项：具体时间、当前时间（操作时）、当前时间加减 n 天。</p> <p><b>优点：</b>                  + 可以随时根据需要调整时间精度。                  + 强大易用的日期选择控件。</p>
<p>自定义字段类型：级联选择</p>	<p>支持。（=Cascading Select）                  支持 1 层子选项。</p> 	<p>👍 无单独的级联选择类型字段。                  （模块字段支持级联选择。可支持多层次子选项。）</p> 
<p>自定义字段类型：数字</p>	<p>支持（浮点数）</p>  <p><b>缺点：</b>                  - 无法实现只允许输入整数的需求。</p>	<p>👍 支持。分两种字段：整数和浮点数。                  + 可以设置宽度。                  + 可以设置最大输入长度。                  + 可以通过格式化字符串控制输出。</p> 

自定义字段类型: URL 类型	支持		支持
自定义字段类型: Email	不支持		支持
自定义字段类型: 布尔类型	不支持。		支持
自定义字段类型: 用户和组选择类型	Group Picker: 选择一个用户组 Multi Group Picker: 选择多个组 Multi User Picker: 选择多个用户 Project Picker: 选择一个项目 Single Version Picker: 选择单个版本 Version Picker: 选择多个版本 User Picker: 选择单个用户		单选和多选类型的可选值允许使用项目或系统内用户。 一般需要选择用户或组的功能, 如 Issue 的相关人、协办人、邮件通知抄送对象, URTracker 都提供了专门的功能, 而不是通过字段扩展来实现。
和其他系统结合用户的字段	Hidden Job Switch Import ID Job Checkbox 似乎用于与 Perforce 集成		不支持
Read-only Text Field	支持		普通文本字段将所有人的编辑权限去掉即可。
字段输入时的提示信息	支持, 在 Field Configurations 中设置。 原估算时间: <input type="text"/> 估算一个问题解决的如时间期限 格式是 "w*d*h*m" (表示星期.天.小时.分钟) - *表示可以是任意数字; 例如: 4d, 5h 30m, 60m and 3w.		支持, 在字段本身的设置页面中设置。 购买日期: 2009-07-30 (日期时间) 时间精确到日期。默认值设置为当前时间。
对长字段内容进行折叠显示	不支持		支持 

## (4) 用户管理和权限控制

比较内容	Jira		URTracker
<b>用户管理</b>			
用户注册方式	JIRA 可以工作在 2 种模式下： 1. 开放 - 任何人都可以注册并提出问题。 2. 私有 - 只有管理员可以创建用户。	👍	通过参数配置，可以实现几种方式： 1. 只允许管理员添加 2. 用户使用者自己注册，但需要管理员审核 3. 允许使用者自己注册，并自动审核
导入用户账号	不支持	👍	支持从 csv 文件导入 支持从 Active Directory 导入
导出账号清单	不支持	👍	支持
账号状态控制	不支持	👍	支持
账号过期时间控制	不支持	👍	支持
IP 访问控制	不支持	👍	支持
用户信息（属性）	<p>内置：用户名、全名、邮件 其他信息需要对每个账号单独添加属性。</p>  <p><b>优点：</b> 能够添加任意的属性。</p> <p><b>缺点：</b> 操作有些麻烦。</p>	🤝	<p>内置：登录账号、真实姓名、Email、密码提示、部门、地址、办公电话、移动电话、MSN、备注、账号过期时间、账号状态等。</p>  <p><b>优点：</b> 比较简单，能够进行导出导入。</p> <p><b>缺点：</b> 不能添加额外的属性。</p>
管理员修改其他账号的密码	不允许	👍	允许
集中设置用户所属的组	支持	👍	支持。 提供从一个账号所属的组复制到另一个账号的功能。
查看在线用户	不支持	👍	支持

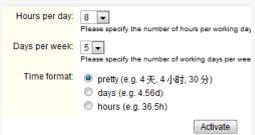
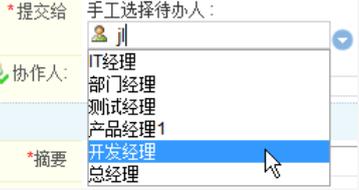
<p>部门属性</p>	<p>没有</p>		<p>有。 在很多功能下可以根据部门过滤要选择的人员。</p> 
<p>员工离职后的账号处理</p>	<p>通过修改用户权限的方式，不让该用户使用 Jira 系统。</p>		<p>设置用户状态为“禁用”。</p>
<p>找回密码</p>	<p>不支持</p>		<p>支持获取密码提示信息。</p>
<p>域认证</p>	<p>支持 LDAP 集成（用于验证密码） 仍然需要手工创建账号， 登录时仍需输入密码。 支持 Atlassian 自身的 SSO 方案</p>		<p>支持 Active Directory 域认证方式。 可以通过从域导入的方式批量添加账号， 能够实现不输入密码即可访问。</p>
<p><b>系统管理权限</b></p>			
<p>系统管理权限的实现方式</p>	<p>通过“组”控制。</p> 		<p>通过“用户-角色-权限”的映射控制。</p> 
<p>系统权限的划分</p>	<p>JIRA System Administrators JIRA Administrators JIRA Users Browse Users Create Shared Object Manage Group Filter Subscriptions Bulk Change</p> <p>Jira 的系统权限更像是一种角色的划分。 不能再将具体的管理权限细分。</p>		<p>系统管理。 账号管理：创建用户、管理用户、删除用户、角色权限管理。 项目管理：创建项目、管理项目。 知识库管理：管理知识库、审核文章。</p>
<p>设置用户管理员</p>	<p>不支持</p>		<p>创建一个角色，赋予账号管理的所有权限即可。</p>
<p>允许部门经理创建自己使用的项目</p>	<p>必须由系统管理员创建</p>		<p>创建一个角色，赋予“创建项目”权限即可。</p>

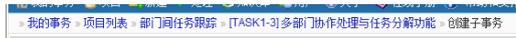
<p>全局用户分组</p>	<p>支持 用于定义 Project Role 的成员、各种权限控制</p>		<p>支持 主要用于两个方面：（1）项目中的工作组可以从某个全局用户组继承用户。（2）在知识库功能中，按照全局用户组设置访问权限。</p>
<p><b>项目和事务权限</b></p>			
<p>项目成员的角色划分</p>	<p>定义全局的“Project Role”，并在具体的项目中为每个 ProjectRole 指定具体成员。</p>  <p><b>优点:</b> + 可以共用相同的默认成员定义。</p> <p><b>缺点:</b> - 所有的 Project Roles 对每个项目都可见 - 只能由系统管理员配置。</p>		<p>在每个项目单独定义“工作组”</p>  <p><b>优点:</b> + 各个项目之间的组和成员定义互相隔离。可以根据需要随意设置。 + 配置起来更简单。 + 使得项目内的管理人员有权限自由调整人员分组。</p>
<p>用户在项目内的权限控制方式</p>	<p>通过 Permission Schemes 控制。</p> 		<p>通过直接设置工作组的权限控制。配置比较直观、简单。</p> 
<p>项目内管理员的权限</p>	<p>只能调整本项目内各 Project Role 的成员、Components、Versions 等信息。权限很小。</p>		<p>可以调整项目的所有配置，包括人员分组、权限、字段、流程等。</p>
<p>Issue 安全级别</p>	<p>通过“问题安全级别”控制哪些人允许查看特定的 Issue。</p> <p><b>缺点:</b> Issue 级别及所对应的查看权限须需事先设定好。</p>		<p>通过设置工作组权限，让某些用户只能查看和自己有关的 Issue，然后设置 Issue 的相关人，可以实现对单个 Issue 的权限控制。</p>
<p>某些人只允许查看和他们有关的 Issue</p>	<p>不支持</p>		<p>去除工作组的“查看项目内所有事务”的权限。 (比如公司内部人员可以查看项目中的所有问题,但是公司的客户只能查看自己提交的问题。)</p> <p><b>优点:</b> + 简单,不需要对每个 Issue 设置访问权限。 + 符合一般的应用需求。</p>

## (5) 文档和知识共享功能比较

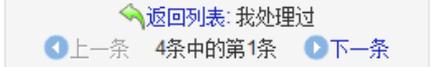
比较内容	Jira		URTracker
项目内文档共享功能	不支持		提供“文档列表”功能,供项目成员之间共享文档资料。 
知识库功能	不支持	 	内置非常完善的知识库功能。 可以将 issue 转变成知识库文章。 可以设置复杂的知识库目录访问控制权限。 可以在 Issue 中引用知识库文章。 

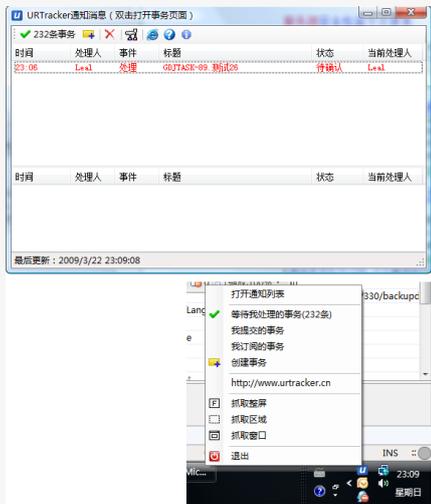
## (6)其他功能比较

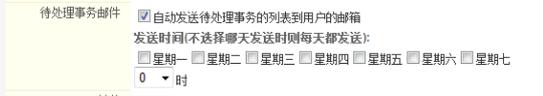
比较内容	Jira		URTracker
项目分组功能	支持单层分组。各个分组及分组中的项目按照字母顺序排列。 <b>缺点:</b> - 不支持多层分组, 在项目比较多的时候, 管理和查看比较混乱。 - 不支持自定义排序。	👍	支持树形多层分组, 支持调整各个分组的排列顺序和分组中项目的排列顺序。在项目比较多的时候, 树形分组可以更好的组织项目, 有很大的优势。 
项目状态设置	不支持	👍	可以通过项目状态关闭已经不再使用的项, 使其从项目列表中隐藏。
配置模板	Jira: 可以将 workflow 导出成 xml 文件。 <b>缺点:</b> - 不能将字段等和 Workflow 相关的配置导出。	👍	URTracker: 可以将工作流、字段设置、工作组设置等相关配置导出成一个文件。 (由于各种配置往往有很强的关联性, 所以一起导出更有价值。方便用户之间共享流程配置。)
工作时间设定	Jira: 可以设定每天工作几个小时, 每周工作几天。  <b>缺点:</b> - 无法根据工作时间查看用户对问题的响应和处理效率。	👍	URTracker: 可以设定每天的工作时段(上午和下午), 每周哪几天工作。还可以定义假期及特殊工作日。  使用工作时间设定可以用于计算定时器的触发时间、用于统计员工处理 issue 的效率。
数据库备份	提供自动备份功能, 允许设定备份目标目录。 备份格式为 xml 文件。	🤝	提供自动备份功能, 允许设定备份目标目录。自动保存最后 7 次的数据库备份文件。允许手动备份。备份格式为 sql server 数据库备份文件。
选择用户时根据汉语拼音搜索	不支持, 只支持英文	👍	可以使用拼音、拼音一部分(如首字母)的组合方式过滤可选的人员清单。当系统内人员较多时, 此功能可以很大的提高效率。 
Dashboard	支持。 允许用户自定义首页显示的内容。	👍	不支持。 首页内容是固定的, 主要是和当前用户相关的问题清单、公告栏、收藏夹等。

设定 Issue 的相关人员	不支持		支持。用于：在通知对象中选择、控制某个 Issue 的访问权限。
导航条	不支持		支持。  <b>优点：</b> + 让用户了解自己所处的位置 + 方便回到任意上一层操作界面
创建 Issue 操作	(1) 点击工具栏上的“创建问题” (2) 选择要创建 Issue 的项目 (3) 选择要创建 Issue 的类型 (4) 录入 Issue 信息，提交 <b>缺点：</b> 每次都要重复选择项目和 Issue 类型。		在任意页面： (1) 点击工具栏上的“新建” (2) 选择项目 (3) 录入 Issue 信息，提交 在某个项目中时： (1) 点击“新建”（直接在当前项目中创建） (2) 录入 Issue 信息，提交 创建好后，可以选择： (1) 继续在本项目中创建新的 Issue (2) 继续新建，并复制刚创建 Issue 的字段值 <b>优点：</b> 可以有效提高录入 Issue 的效率。
编辑 Issue	支持。 允许编辑的字段由编辑操作所绑定的 Screen 决定。 <b>缺点：</b> 表单是固定的，对所有人相同，无字段权限控制。		支持。 根据用户的权限不同，所能编辑的字段也不同。（如测试只能编辑和 BUG 描述相关的字段。）
Issue 关联	支持 Issue Link 功能		支持设置“相关事务”功能。
移动 Issue	支持		不支持。（可以通过“复制到其他项目”实现类似的需求）
复制 Issue	支持。直接复制生成新的 Issue。 <b>缺点：</b> - 只能设置新 Issue 的摘要，不能调整其他字段的值。		支持。 <b>优点：</b> + 复制过程中允许更改字段的值，允许选择是否复制附件。 + 自动在复制和被复制的 Issue 间建立关联。
复制 Issue 到其他项目	不支持。		支持。 自动复制同名字段的值并允许调整。 自动在复制和被复制的 Issue 间建立关联。
发表评论或备注	支持“备注”功能。 只能填写纯文本信息，不能带格式。 不能上传附件。 不能手动选择通知对象。 可以设置“允许查看备注”的人。		支持“评论”功能。 可以输入格式文本，实现录入表格、进行贴图、调整字体颜色大小等操作。 可以上传附件。 可以手动选择通知对象。

工作记录	支持。 和 Time Tracking 功能在一起。如果 Time Tracking 功能禁用了，工作记录功能就没有了。		支持
模板打印	不支持		允许指定一个模板文件控制 Issue 信息的打印格式。
批量操作	支持批量编辑、批量删除等		支持批量处理功能
监视事务	支持		支持（订阅事务） 还可以“监视”项目，项目中有新建 issue 时，自动接收到通知。
事务投票功能	有		无
CVS 集成	有		无
Issue 更改历史	有		有
Issue 处理过程图示	无		有 
附件处理	全局附件选项：是否允许所有人上传附件；附件目录；附件大小限制；图片附件生成缩略图。 所有附件附加在 issue 上。		可以设置创建问题或处理问题的每个步骤是否允许上传附件。 附件可以附加在 issue 本身，也可以附加到某个处理步骤上。 可以控制每个工作组的成员的附件管理权限。
从 Excel 导入 Issue	支持，处理比较复杂，操作步骤比较多。		支持，先生成导入模板，再导入，不需要额外的处理，操作很简单。
导出 excel	支持，只能选择“所有字段”或“当前字段”，不能手动选择要导出的字段。		支持，可以选择要导出哪些字段，可以保存导出设置。
根据 ID/编码/关键词/关键词快速查找	支持		支持
字段组合查询（各条件 And 关系）	支持 		支持 

<p>快速过滤 issue</p>	<p>(1) 全部、突出的、未归划、分配给我的、我报告的、最近解决的、最近增加的、最近更新的、最重要的；</p> <p>(2) 各状态</p> <p>(3) 各优先级</p> <p>(4) 各开发人员</p> <p>(5) 模块</p> <p>(6) 版本</p> 		<p>(1) 相关：提交给我，我提交，我处理过，我订阅，我相关</p> <p>(2) 跟踪中，已关闭，所有</p> <p>(3) 各状态</p> <p>(4) 各优先级</p> <p>(5) 各分类</p> <p>(6) 各模块</p> <p>(7) 时间：今天创建、今天关闭、今天更新、昨天创建，昨天关闭，本周创建、本周关闭、上周创建、上周关闭、本月创建、本月关闭、上月关闭、即将到期、已过期等；</p> <p>(8) 当前待办人</p> <p>(9) 创建人</p> <p>(10) 各个“单选”或者“文本”类型的自定义字段</p> 
<p>高级查询 (复杂的 And、Or 条件组合)</p>	<p>不支持</p>		<p>支持</p> 
<p>根据 Issue 经历 Transition 的次数查询</p>	<p>不支持</p>		<p>支持。</p> <p>根据事务所经历的状态转换步骤查询事务</p> <p>从源状态: 已回复 到目标状态: 咨询中 经历转换次数: &gt; 1 查询</p> <p>(如用于如查询 Reopen 超过 1 次的 Issue。)</p>
<p>根据 Issue 在某个状态停留时间查询</p>	<p>不支持</p>		<p>支持</p> <p>根据事务在某个状态的停留时间</p> <p>状态: 咨询中 停留时间大于: 3d 查询</p>
<p>条件删选</p>	<p>不支持</p>		<p>可以按照“是否包含附件”“是否已关闭”“是否挂起”等条件组合过滤查询结果。</p> 
<p>在查询结果中导航，返回查询结果列表</p>	<p>支持</p> 		<p>支持</p> 

<p>在 Issue 列表中预览 Issue 信息</p>	<p>不支持</p>		<p>支持</p> 
<p>在 Issue 列表中预览 Issue 处理过程</p>	<p>不支持</p>		<p>支持</p>
<p>职务代理人功能</p>	<p>不支持</p>		<p>支持。代理人功能用于在出差时，请其他人代为处理提交给某个人的 Issue。</p>
<p>项目日历功能</p>	<p>不支持</p>		<p>支持</p> 
<p>统计和图形报表</p>	<p>需要插件 Plugin</p>		<p>支持。支持分布统计、分布图、趋势图、工作统计、状态停留时间统计等功能。</p>
<p>Plugin 插件功能</p>	<p>支持</p>		<p>不支持。</p>
<p>桌面助手</p>	<p>不支持</p>		<p>支持。 可以实现接收通知、截屏上传等功能。</p> 

收藏夹功能	不支持		支持
TotoList 邮件	不支持		支持设定一周的哪些天自动向每个人发送待处理 Issue 列表的邮件。 

## (7) 总结

两个软件都非常强大，可定制性很强。

Jira 通过 Scheme 机制实现配置共享，但是引入了非常多的难以理解的概念和非常高的复杂程度，加大了各种配置之间的关联，降低了对每个项目进行单独设置的灵活性。

Jira 的 Plugin 和 Dashboard 的功能很有特色。通过 Plugin 可以做不少功能的扩展。

Jira 的某些设计不利于将其应用于除了 bug 跟踪以外的其他流程。

Jira 的中文支持有待加强。

URTracker 各个项目的配置相互独立，大大简化了各种配置的复杂程度，使配置更简单，更直接，更灵活。流程的实现方式，也更适合于各种类型 Issue 的跟踪管理。

URTracker 可以通过状态时限、定时器等实现对流程步骤的时限控制，以及自动的告警通知、自动升级等功能。

URTracker 提供了强大的 Issue 分解功能，使其能够轻松的实现 WBS 任务分解，更好的应用于项目管理、计划任务管理等方面。

URTracker 还提供了一套非常完善的知识库系统，可以方便的将有价值的 Issue 转变成文章进行共享。URTracker 还提供了“文档列表”功能，方便在项目中共享文档资料。

总之，URTracker 通过更简单的方式提供了更灵活和强大的权限和流程控制能力、更多的实用的功能。JIRA 作为一个比较出名的软件，也有自己的特色。