JIRA vs Bugzilla

JIRA vs. Bugzilla

Mozilla's Bugzilla and Atlassian JIRA are among the most popular issue trackers. While they share features common to issue trackers, the systems are very different.

This article highlights some differences that may be crucial if you are migrating from one system to another. It is not a comprehensive comparison.

	Bugzilla	JIRA
License	Open-Source (MPL)	Commercial (Source code available to commercial license owners)
Server-Side Architecture	Perl MySQL / PostgreSQL / Oracle	Java (J2EE) MySQL / PostgreSQL / Oracle / MS SQL Server Tomcat Lucene
Server Load	Low – Perl scripts act as simple CGIs and can be highly sped up with mod_perl. On large databases (>100,000 bugs) the database may become the bottleneck when doing search queries. You can also keep multiple Bugzilla instances running on one server without much overhead.	Considerable – JIRA is a more complex system and typically executes a whole lot more of the server-side code per web request, so the server load is considerably higher (*but search in JIRA may be faster than in Bugzilla due to the Lucene index*). Although you'd need more powerful hardware for JIRA, it will likely do well on a reasonably modern CPU and 1GB of memory. As a general rule, Atlassian recommends not to store more than 200,000 issues on a single JIRA instance.
Access Control	Security groups: quite flexible, but a bit mind-bending feature for grouping users & issues and granting permissions.	JIRA has more simple permissions model, more conventional and arguably more convenient. Migrating from Bugzilla security groups might be not an easy task.
Flags /Requests	This is a quite unique Bugzilla feature.	Labels in JIRA can be used as basic flags (without the assignment of a "?" flag). Excellent extensibility leaves room for a plugin that mimics Bugzilla's flags feature more precisely.
Search Power	Bugzilla's advanced search is quite powerful, especially when it comes to all the options available in the Boolean Charts and high-precision searches like regexp matching. Mastering all the options and understanding how Boolean charts work could be a challenge though.	JIRA has flexible JQL language (JIRA Query Language) that allows you to build arbitrary boolean expressions. The interface for writing a query is a text box with auto-completion and error highlighting, and is generally more usable than the large Bugzilla's form. JIRA lacks some of the expert-level search conditions that Bugzilla can do and searching for text in JIRA issues may be limited by how Lucene index works. Third-party plugins can extend JQL functionality through custom functions.
Security	Due to full open-source code exposure and usage by Mozilla and some other big players, Bugzilla's security should theoretically be very high. Bugzilla team publishes security advisories and releases security patches in a very reasonable time.	Atlassian takes security seriously and publishes security advisories and patches when security threats become known. Compared to Bugzilla, JIRA security risks are somewhat higher due to the larger overall complexity, a lot of client-side JavaScript code, and additional functionality provided by 3rd-party plugins.
User Interface	Bugzilla user interface hasn't changed much over the years. Some usability improvements have been added, but overall, nothing fancy. The HTML is generated from templates, and some companies have modified those templates to make Bugzilla look a lot better.	JIRA user interface by far better than Bugzilla, out of the box. A lot of effort has been put into it, and it's being continuously improved.
Custom Field Types	Types available in Bugzilla: Text fields, Multi-selection, Drop-down, Date/time, Bug ID	Custom field types available in JIRA are too many to list, and even more custom field types available from plugins.
Custom Field Values / Conditions	Bugzilla lets you show/hide the whole custom field or specific values based on the value of some other field.	JIRA allows conditional configuration based only on Project and Type fields.
Workflow	Bugzilla lets the admin define a global workflow for all Products by editing transition matrix (each cell allowing transition from status A to status B). Unlike JIRA, Bugzilla can let users select the initial status of a new issue and lets the admin configure which transitions require comments.	Unlike Bugzilla, JIRA allows to define multiple workflows which are applied based on the issue's Project and Type. Each workflow may transition issue through a subset of statuses, with each transition possibly having a separate set of fields (Screen) that the user can fill out when doing the transition. Transitions may have pluggable, configurable conditions and post-functions. Overall, JIRA workflows are far more powerful albeit missing some features available in Bugzilla.
Issue Linking	Bugzilla has only one link type: blocks/depends, plus a Bug ID custom field.	JIRA has configurable link types with user-defined semantics. JIRA also has pluggable remote issue links, that allow to link an issue to any other entity outside JIRA.

| Plugins, Add ons, Customizations | Bugzilla has a few Bugzilla Add-Ons. Unofficial Bugzilla patches may be found on the Internet. | JIRA functionality can be extended and altered by plug-ins to a very large extent. Atlassian has a thriving developer ecosystem with their own Plugin SDK, and JIRA has rich set of APIs that make it easy to create powerful plugins. Lots of JIRA plugins are listed on Plugin Exchange. |