

EVALUATION OF ISSUE TRACKING AND PROJECT MANAGEMENT TOOLS FOR USE ACROSS ALL CSIRO RADIO TELESCOPE FACILITIES

J.C. Guzman, CSIRO Astronomy and Space Science, Epping, Australia

Abstract

CSIRO's radio astronomy observatories are collectively known as the Australia Telescope National Facility, or ATNF. The ATNF is operated and managed by CSIRO Astronomy and Space Science, a division of CSIRO, Australia's national science agency. The facilities include the 64-metre dish at Parkes, the Australia Telescope Compact Array (ATCA) located in Narrabri, the Mopra 22-metre dish near Coonabarabran and the ASKAP telescope located in Western Australia and in early stages of commissioning. In early 2013 a new group named Software and Computing has been formed. This group, part of the ATNF Operations Program is responsible for the software development and maintenance for all ATNF facilities, from monitoring and control systems to science data processing and archiving. The new group brings all the expertise in software development, data processing, High-Performance Computing (HPC) and data archiving under one umbrella. One of the first tasks of the new group is to start "standardizing" the way software development is done across all radio telescopes. This paper presents the results of the evaluation of several issue tracking and project management tools, including Redmine and JIRA to be used as a common software development management tool across all ATNF facilities. This paper also describes how these tools can potentially be used for non-software type of applications such as fault reporting and tracking system.

MOTIVATION

The current ATNF Fault Report System has been in use since 1998 for the Parkes telescope and since 2001 for the Compact Array and Mopra. A single developer currently spending approximately 5 to 10 days per year maintains the system. The software is relatively simple providing basic fault tracking functionality and search capabilities. The client side is entirely via Web and accessed via ATNF website. The system supports multiple telescopes (or sites) and several categories (but no support for sub-categories). Despite its simplicity, the current system has many limitations raised by several users. These issues, and the need to support the new ASKAP telescope, triggered a review of the current system and evaluation of off-the-shelf alternatives. An internal draft report [1] was released in early September 2013 to the Operations management and users for consultation. An updated version of this report is currently in progress.

The ATNF also maintains two other project management tools (for historical reasons) used by different groups across the division:

- Trac [2] used by the Computing Infrastructure team to track UNIX system administration tasks. There are also several software projects for Parkes, ATCA and Mopra in Trac supporting internal and external users.
- Redmine [3] is used mainly by the ASKAP project, including the ASKAP software development project, system engineering and commissioning, and hardware development. There is also a Redmine instance supporting Science projects for internal and external users.

Since the merge of the two software development teams into a single group earlier in 2013, it is evident that using a single issue-tracking system to track all software maintenance and development tasks for all ATNF telescope systems is paramount. Homogenising processes and tools brings several benefits to the organisation, including easier maintenance costs of the tools, better management (assignment) of tasks across many projects and individuals located at different sites (distributed geographically), more visibility of the tasks to be completed and better management of software releases.

This paper describes the results of the analysis and evaluation of off-the-shelf issue-tracking tools that can be used as a Fault Report System (or Support System) and a software development management tool.

REQUIREMENTS

The Fault Tracking and Reporting System requirements captured in [1] are very similar to what is provided by a Helpdesk or Support tracking system. There are additional requirements to support software development and maintenance activities. A list of requirements for an issue-tracking tool that can be used as a Fault Tracking and Reporting system and as a software development and maintenance system are listed below. An **issue** can be a bug, system fault, task, and general or feature request.

- Track Issues, including creating issues by an internal or external user, assign issues to an individual or a group, add comments or work logs by internal or external users via provided UI (optionally via email), add attachments, link related issues, change issue states (open, in progress, etc.), change issue's priority
- Email notification
- Support for watchers
- Support for multiple telescopes (projects)
- Custom categories and sub-categories
- Custom fields
- Custom workflows
- Administration of user and groups

ISBN 978-3-95450-139-7

- Access security
- Application Programmable Interface (API) to create/modify existing web user interfaces
- Support for importing from other issue-tracking systems, at least the Comma-Separated Values (CSV) format
- Search issues using common queries, such as unassigned and open issues, unresolved issues per telescope, category and/or sub-category, by keywords in the description text or in the work log for comments
- Search issues using custom query
- Basic reporting, including: estimated observing time lost for a specified period of time, and issue statistics
- Provide custom reports
- Integration with revision control system, including subversion and git
- Code statistics and visualisation
- Group and track issues into milestones and versions
- Generate release notes
- Integration with wiki tool
- Web-based user interface
- Multi-platform (server side)
- Easy and intuitive to use
- Easy to install, administer and maintain

IMPLEMENTATION OPTIONS

There are many commercial and open source issue-tracking tools currently available as listed in [4]. Out of all the available commercial and open source issue-tracking tools, the author selected two candidates: Redmine and JIRA [5] for further evaluation. Redmine has been chosen because it has been already in used in our division mainly by the ASKAP project (including Software development). JIRA was chosen because it is used in other astronomical and high-energy physics facilities as well as in other divisions of CSIRO.

Redmine

Redmine is one of the most popular open source flexible project management systems. Written using the Ruby on Rails framework, it is cross-platform and cross-database. Redmine is released under the terms of the General Public License (GPL) version 2. A detailed list of supported features can be found in [3].

Advantages of Redmine:

- In use in ASKAP for many years so ASKAP staff are very familiar with the tool.
- Supports multiple projects arranged in a hierarchical structure, i.e. projects, sub-projects, sub-sub-projects, etc. However highly nested project structures can be quite complex and perhaps confusing.
- Has an in-built wiki per project.
- Highly flexible without code development.
- Good search capability.

- Provides a REST API for developing custom web applications accessing the Redmine database.
- No license fee.

Disadvantages of Redmine:

- Some users complained about its “bland” and less intuitive user interface compared to JIRA.
- Smaller user base compared to JIRA.
- Although there is an external plugin available, there are not many documented cases where Redmine has been used as a helpdesk/support ticketing systems.
- Only basic reporting (time spent) available off-the-shelf. More complex reporting can be added via external plug-ins (if available) or by creating your own plugins (requires Ruby on Rails programming skills).
- There is a plug-in available to support sub-categories [4] but the author did not have time to test it.

Worth noting that in 2010 Redmine project was forked by a group of ex-Redmine developers and formed ChilliProject [6], a competitor of Redmine. This could potentially raise questions about the health and stability of the project. However there have been 38 releases since February 2011 going from 1.1.1 to 2.4.0 current. Hardly a sign that the Redmine project is somehow dying.

JIRA

JIRA is a commercial software product, developed by Atlassian Inc., and used for issue tracking and project management. The product name, JIRA, is not an acronym but rather a truncation of “Gojira”, the Japanese name for Godzilla. It has been developed since 2002. JIRA is written in Java. It integrates with source control programs such as Subversion and git. JIRA's flexible plugin architecture spawned a large number of plugins developed by the JIRA development community and third parties, including IDEs like Eclipse using the Atlassian IDE Connector. The JIRA API allows developers to integrate third-party applications into JIRA. There are hundreds of available plugins that extends JIRA in the Atlassian Marketplace website [7]. Plugins are available both for free and licensed.

JIRA has many similar features as an issue-tracking and project management tool as Redmine. The full list of JIRA features can be found in [5]. Some of the differences between JIRA and Redmine are listed below:

- JIRA supports multiple projects and provides grouping of projects in “project categories” but does not support nested hierarchical project structure like Redmine.
- Redmine has the wiki and some tools to assist software development out-of-the box. JIRA on the other hand only provides the issue management part. Wiki (Confluence [8]) and other software development add-ons are available for purchase.

- JIRA maintains the resolution of the issue (fixed, won't fix, duplicate, incomplete, cannot reproduce) as a separate in-built field.
- JIRA support voting of issues.
- JIRA has additional and more comprehensive time tracking fields such as updated, resolved, original estimate and remaining estimate times. These times are recorded separately and available for easy reporting.
- JIRA contains several pre-built reports. Redmine only offers time spent report out-of-the-box.

Advantages of JIRA:

- Very intuitive and easy to use user interface.
- Better user and developers documentation compared to Redmine.
- Better customization of the dashboard compared to Redmine.
- Larger user base compared to Redmine.
- Issue collector and/or user feedback plug-in (web embedded).
- Highly flexible without code development.
- REST API available for integrating with our existing Web user interface.
- Several built-in reports available.
- Lots of plugin extensions available (free and commercial).

Disadvantages of JIRA:

- Moderate license fee. JIRA license model charges per user, approximately \$40 per user if JIRA is running on your premises.
- Wiki (Confluence) and software tools have to be purchased separately.
- Additional cost to ATNF of migrating part (or entire) existing Redmine to JIRA.

COMPARISON RESULTS

Table 1 presents a comparison between JIRA (version 5.2) and Redmine (version 2.3.2). The scores are based on the author's analysis of the off-the-shelf documentation ([9], [10], [11]) and having "played" with trial versions of both. The meaning of each scoring value is listed below:

- Score 5 = fully supported and verified in the trial versions
- Score 4 = fully supported according to the documentation but not verified in trial version; or fully supported and verified but with some constraints
- Score 3 = supported but more effort in customisation or some code might be required to achieve the requirement in full
- Score 2 = not currently supported, but it is possible to extend with significant code effort
- Score 1 = not supported at all and it is not possible to extend via some coding effort

Table 1: Comparison between JIRA and Redmine

Requirement	JIRA 5.2	Redmine 2.3.2
Track issues	5	5
Email notification (a)	4	4
Support for watchers	5	5
Support for multiple projects	5	5
Custom categories and sub-categories (b)	5	4
Custom fields	5	5
Custom workflows (c)	5	4
Administration of user and groups	5	5
Access security	5	5
API	5	5
Search faults using common queries	5	5
Search faults using custom query	5	5
Basic reporting (d)	5	3
Provide custom reports (e)	4	3
Import from other issue tracking system (CVS format) (f)	4	4
Integration with revision control system, including subversion and git	4	4
Code statistics and visualisation (g)	4	3
Group and track issues into milestones and versions	5	5
Generate release notes (h)	4	4
Integration with wiki tool (i)	5	5
Web-based user interface	5	5
Multi-platform (server-side)	5	5
Easy and intuitive to use (j)	4	3
Easy to install, administer and maintain	5	5
User, administration and programmer's documentation (k)	5	3
TOTAL	118	109

- (a) Both JIRA and Redmine support creating and updating issues via email. However it requires setting up your email server, which was not done during the trial period.
- (b) Redmine supports sub-categories via the following plugin https://github.com/bpat1434/redmine_category_tree. The author did not verify this plugin.
- (c) JIRA provides a graphical workflow editor.
- (d) JIRA has many built-in reports compared to Redmine. Redmine only offers time-spent reports.
- (e) JIRA provides a tutorial how to create a report but the author did not verify in the trial version. Redmine does not provide specific

- tutorial on custom reporting, but there are plenty of documentation how to create a plugin.
- (f) Both tools provide external tools that import data from existing issues tracking systems, including from CVS format files. The author did not try these tools and cannot confirm how good they are.
 - (g) JIRA requires the additional product FishEye [12]. FishEye has a very nice UI compared to Redmine built-in code statistics.
 - (h) For Redmine, there is a plugin that provides this functionality:
https://github.com/hdgarrood/redmine_release_notes.
 - (i) To complement JIRA, the Atlassian product Confluence provides enterprise wiki. It has a WYSIWYG editor and provides content structure. Redmine has a built-in wiki, but it is rather limited, in particular when structured content is required. There is a plugin that provides a WYSIWYG editor.
 - (j) The author undertook a survey amongst several users (20 respondents) and these are the averaged scores.
 - (k) JIRA documentation is quite comprehensive and easy to navigate compared to Redmine.

CONCLUSIONS

As seen in the Table 1, both JIRA and Redmine provide all the essential requirements to be used as a Fault Tracking and Reporting system for ATNF telescopes. The internal report recommends that the existing Fault Report system should be replaced by one of these off-the-shelf alternatives, because maintaining in-house code is usually expensive in the long run if an off-the-shelf can be used instead. The difficult question is which one.

The results from Table 1 shows that JIRA is slightly better than Redmine, especially in the area of reporting, look and feel, user documentation, more case studies as a Helpdesk system and more experience in the astronomical and high-energy physics community. JIRA is also supported by CSIRO corporate IT division and used in some of their projects. For these reasons, the draft version of the internal report recommended JIRA to be used as the ATNF Fault Tracking and Report System. However, Redmine is currently being used in our division, mainly in the ASKAP project and there will be costs associated in migrating existing Redmine projects into JIRA, although it is not yet clear if all the Redmine projects have to be moved.

A decision has not been made yet at the time of writing this paper. A round of staff consultation, especially the ones that will be affected by the change are currently on going. The report is also being expanded to include an analysis of the impact of making JIRA or Redmine the single issue-tracking tool for the whole division, and

possible implementation options. It is envisage that a decision will be made by the end of October 2013.

Both JIRA and Redmine provide all the essential requirements for an issue-tracking and project management tool. Despite a decision in the ATNF division not been made yet, these are some of the author's thoughts and suggestions for people looking or evaluating issue-tracking tools:

- Use one; it will make your life as a developer or manager easier.
- If you are a single developer or a small software development team starting a new project that prefers an open source alternative consider evaluating or using Redmine, especially if you are looking after a tool that supports multiple projects and software development or maintenance workflows.
- For larger projects or teams and if you don't mind spending a modest license fee, consider using or evaluating JIRA. JIRA has a nicer UI, better reporting features (appeals more to managers) compared to Redmine. It has a larger user base and supports many different workflows from Helpdesk-type system to software development projects. Bear in mind that if you need a wiki, you will have to consider Confluence to complement JIRA (additional cost). For software development projects, it is also recommended to add some of the tools that complements JIRA, especially FishEye.

The analysis and comparison results described in this paper were done with JIRA 5.2 and Redmine 2.3.2, so bear in mind that features provided by both tools might have changed by the time you read this paper, so consult the corresponding vendors for up-to-date information.

REFERENCES

- [1] J.C.Guzman, *ATNF Fault Report and Tracking System (FRTS) Requirements Specifications and Implementation Options*, Draft version 0.1, Sep 2013 (CSIRO internal only).
- [2] <http://trac.edgewall.org>
- [3] <http://www.redmine.org>
- [4] Comparison of issue tracking systems; <http://en.wikipedia.org>
- [5] <https://www.atlassian.com/software/jira>
- [6] <https://www.chiliproject.org>
- [7] <https://marketplace.atlassian.com>
- [8] <https://www.atlassian.com/software/confluence>
- [9] JIRA 5.2 Online Documentation, <https://confluence.atlassian.com/display/JIRA052/JIRA+Documentation>.
- [10] Andrew Lesyuk. *Mastering Redmine*. Packt Publishing, 2013.
- [11] Patrick Li. *JIRA 5.2 Essentials*. Packt Publishing, 2013.
- [12] <https://www.atlassian.com/software/fisheye/overview>