

OASIS Overview

Managing TC Work Using JIRA

March 10, 2014

Goal

- Demonstrate OASIS JIRA installation
- Show best practices for managing development projects
- Give TC a place to start

What is JIRA?

- Issue-tracking system used for bug & issue tracking, project management, etc.
- Components:
 - Projects – the overall collection of issues. At OASIS, we set up one project per TC (e.g. AMQP TC, Energy Interop TC)
 - Issues – the individual entries that the TC enters, tracks, resolves & eventually closes (e.g. ENERGYINTEROP-504, Add Reading Type as optional element in ei:Market)
 - Components – Classifications defined by the TC to tag issues into logical groupings (e.g. Spec, Schema, Examples)
 - Versions – Classifications defined by the TC to tag issues into specific lifecycle stages (e.g. wdo7, csprdo2)
 - Status– Indicators of the progress in resolving the issue

What is JIRA?



- Summary
- Issues**
- Road Map
- Change Log
- Reports
- Versions
- Components
- Labels

Issues

All issues
Unresolved

Added recently
Resolved recently
Updated recently

Assigned to me
Reported by me

Unscheduled
Outstanding

Unresolved: By Priority

Priority	Issues	Percentage
↑ Critical	2	2%
↑ Major	103	86%
↓ Minor	14	12%
↓ Trivial	1	1%

[View Issues](#)

Unresolved: By Assignee

Assignee	Issues	Percentage
Aaron Zhang	6	5%
Dale Moberg	3	3%
Derek Palma	28	23%
Douglas Neuse	1	1%
Frank Leymann	5	4%
Ken Zink	2	2%
Kevin Wilson	1	1%
Koert Struijk	1	1%
Krishna Raman	3	3%
Matthew Rutkowski	2	2%
Paul Lipton	1	1%
Paul Zhang	1	1%
Steve Fanshler	1	1%

Status Summary

Status	Issues	Percentage
→ Open	15	12%
→ Closed	62	50%
→ New	43	34%
→ Deferred	5	4%

[View Issues](#)

Unresolved: By Component

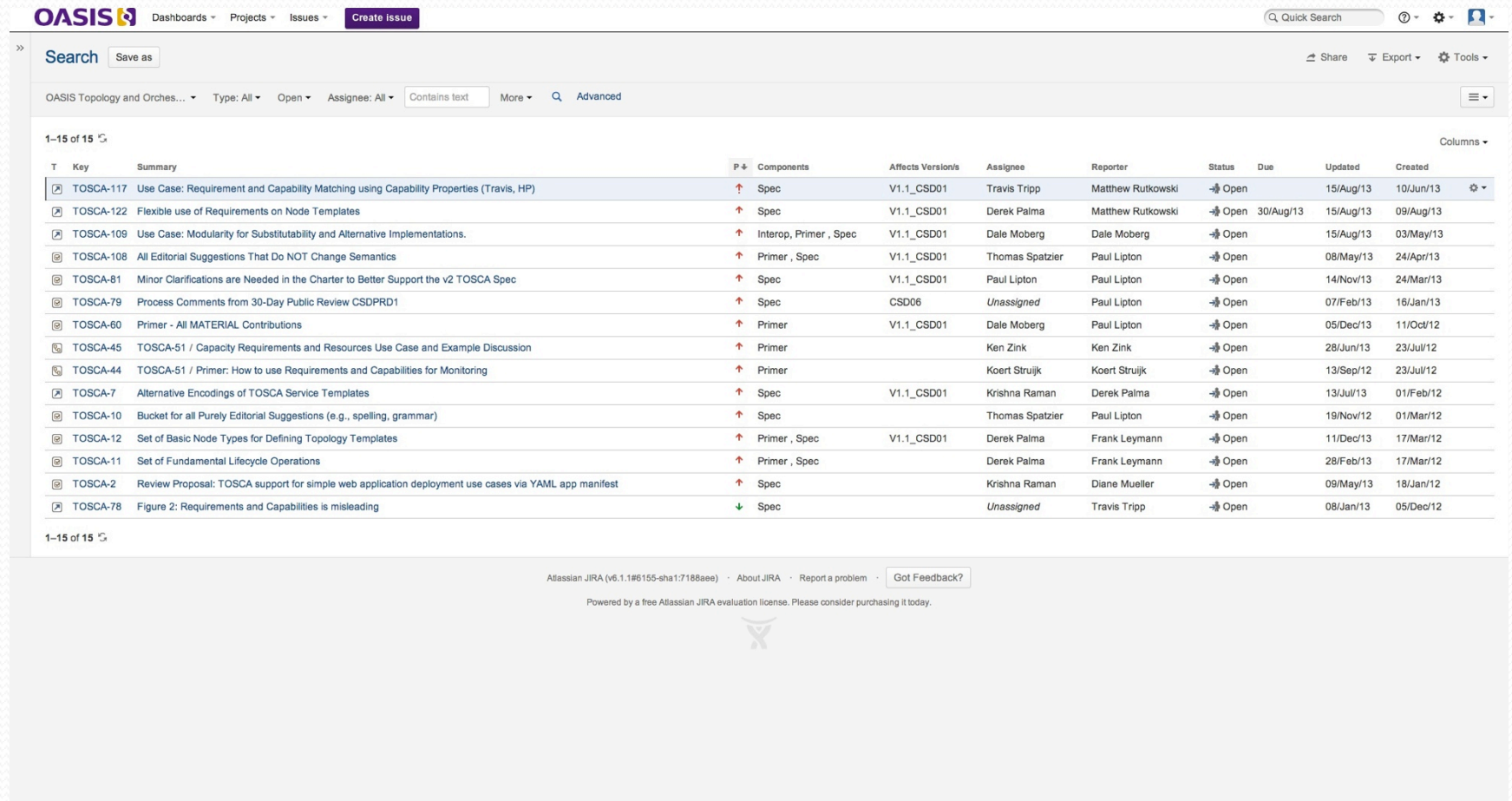
Component	Issues
→ Errata	2
→ Interop	28
→ Primer	28
→ Spec	65
→ No Component	5

[View Issues](#)

Unresolved: By Issue Type

Issue Type	Issues	Percentage
→ Bug	7	6%
→ Improvement	40	33%
→ New Feature	7	6%
→		

What is JIRA?



The screenshot displays the JIRA web interface. At the top, there is a navigation bar with 'OASIS' branding, 'Dashboards', 'Projects', 'Issues', and a 'Create Issue' button. A search bar is located on the right. Below the navigation bar, there is a search filter section with 'Search' and 'Save as' buttons, and a 'Share' button. The main content area shows a list of 15 issues, with the first 15 displayed. The issues are listed in a table with columns for Key, Summary, P (Priority), Components, Affects Version/s, Assignee, Reporter, Status, Due, Updated, and Created. The issues are sorted by priority, with 'Open' issues at the top and 'Spec' issues at the bottom. The issues are: TOSCA-117, TOSCA-122, TOSCA-109, TOSCA-108, TOSCA-81, TOSCA-79, TOSCA-60, TOSCA-45, TOSCA-44, TOSCA-7, TOSCA-10, TOSCA-12, TOSCA-11, TOSCA-2, and TOSCA-78. The footer of the interface includes the version 'Atlassian JIRA (v6.1.1#6155-sha1:7188aee)', 'About JIRA', 'Report a problem', 'Got Feedback?', and a note about the evaluation license.

T	Key	Summary	P	Components	Affects Version/s	Assignee	Reporter	Status	Due	Updated	Created
	TOSCA-117	Use Case: Requirement and Capability Matching using Capability Properties (Travis, HP)	↑	Spec	V1.1_CSD01	Travis Tripp	Matthew Rutkowski	→ Open		15/Aug/13	10/Jun/13
	TOSCA-122	Flexible use of Requirements on Node Templates	↑	Spec	V1.1_CSD01	Derek Palma	Matthew Rutkowski	→ Open	30/Aug/13	15/Aug/13	09/Aug/13
	TOSCA-109	Use Case: Modularity for Substitutability and Alternative Implementations.	↑	Interop, Primer, Spec	V1.1_CSD01	Dale Moberg	Dale Moberg	→ Open		15/Aug/13	03/May/13
	TOSCA-108	All Editorial Suggestions That Do NOT Change Semantics	↑	Primer, Spec	V1.1_CSD01	Thomas Spatzier	Paul Lipton	→ Open		08/May/13	24/Apr/13
	TOSCA-81	Minor Clarifications are Needed in the Charter to Better Support the v2 TOSCA Spec	↑	Spec	V1.1_CSD01	Paul Lipton	Paul Lipton	→ Open		14/Nov/13	24/Mar/13
	TOSCA-79	Process Comments from 30-Day Public Review CSDPRD1	↑	Spec	CSD06	Unassigned	Paul Lipton	→ Open		07/Feb/13	16/Jan/13
	TOSCA-60	Primer - All MATERIAL Contributions	↑	Primer	V1.1_CSD01	Dale Moberg	Paul Lipton	→ Open		05/Dec/13	11/Oct/12
	TOSCA-45	TOSCA-51 / Capacity Requirements and Resources Use Case and Example Discussion	↑	Primer		Ken Zink	Ken Zink	→ Open		28/Jun/13	23/Jul/12
	TOSCA-44	TOSCA-51 / Primer: How to use Requirements and Capabilities for Monitoring	↑	Primer		Koert Struijk	Koert Struijk	→ Open		13/Sep/12	23/Jul/12
	TOSCA-7	Alternative Encodings of TOSCA Service Templates	↑	Spec	V1.1_CSD01	Krishna Raman	Derek Palma	→ Open		13/Jul/13	01/Feb/12
	TOSCA-10	Bucket for all Purely Editorial Suggestions (e.g., spelling, grammar)	↑	Spec		Thomas Spatzier	Paul Lipton	→ Open		19/Nov/12	01/Mar/12
	TOSCA-12	Set of Basic Node Types for Defining Topology Templates	↑	Primer, Spec	V1.1_CSD01	Derek Palma	Frank Leymann	→ Open		11/Dec/13	17/Mar/12
	TOSCA-11	Set of Fundamental Lifecycle Operations	↑	Primer, Spec		Derek Palma	Frank Leymann	→ Open		28/Feb/13	17/Mar/12
	TOSCA-2	Review Proposal: TOSCA support for simple web application deployment use cases via YAML app manifest	↑	Spec		Krishna Raman	Diane Mueller	→ Open		09/May/13	18/Jan/12
	TOSCA-78	Figure 2: Requirements and Capabilities is misleading	↓	Spec		Unassigned	Travis Tripp	→ Open		08/Jan/13	05/Dec/12

Why use JIRA?

- Easier to track & manage than via emails
 - Issues don't get lost or forgotten
 - All information on issue can be kept in one place
 - Issues can be collected, managed, and tracked in groups using Components and Versions
- Workflow offers a structured approach to addressing issues (i.e. New, Open, Resolved, Applied, Closed)
- Provides an audit trail for TC decisions and actions
- Comment resolution logs can be produced from JIRA

Organizing Your Project

- OASIS creates your project at TC level
 - E.g. OASIS eTMF TC, OASIS AMQP TC
- Within your project:
 - Use Components to group issues into logical categories (e.g. 'spec,' 'schema,' 'terminology')
 - Use Versions to organize issues by work progress stage or other meaningful breakdown (e.g. 'wdo1,' 'csprd01')
 - Use Status to manage work on each issue

Organizing Your Project



- Summary
- Issue Types
- Workflows
- Screens
- Fields
- Versions
- Components**
- Roles
- Permissions
- Issue Security
- Notifications
- Issue Collectors
- Development Tools

Components

Projects can be broken down into components, e.g. "Database", "User Interface". Issues can then be categorised against different components.

Name	Description	Component Lead	Default Assignee
<input type="text"/>	<input type="text"/>	<input type="text"/>	Project Default (Unassign) <input type="button" value="Add"/>
Errata			Project Default <input type="button" value="Delete"/>
Interop	This components is used to track TOSCA Interop SC member use cases and issues related to deploying or managing sample TOSCA CSAR files and/or service templates.	Matthew Rutkowski	Project Default <input type="button" value="Delete"/>
Primer	TOSCA primer provides non-normative overview, examples, value statements, and more.		Project Default <input type="button" value="Delete"/>
Spec	All the specification documents, but not XML files		Project Default <input type="button" value="Delete"/>



Organizing Your Project



- Summary
- Issue Types
- Workflows
- Screens
- Fields
- Versions**
- Components
- Roles
- Permissions
- Issue Security
- Notifications
- Issue Collectors
- Development Tools



Versions

Merge

For software projects, JIRA allows you to track different versions, e.g. 1.0, 2.0. Issues can be assigned to versions.

Name	Description	Start date	Release date	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Add"/>
OASIS Standard v1.0	<i>Enter value</i>			
V1.1_CSD01	First CSD of V.next			
CSD07	Changes made after 30-day public review CSDPRD1			
CSD06				
CSD05	Most likely final issues			
CSD04	Requirements/Capabilities and Other Enhancements			
CSD03	Packaging and Use Case-Driven Changes to Define Relationships and Operations and other items			
CSD2				
CSD1	First Milestone - Reflects TOSCA-1 & TOSCA-6			



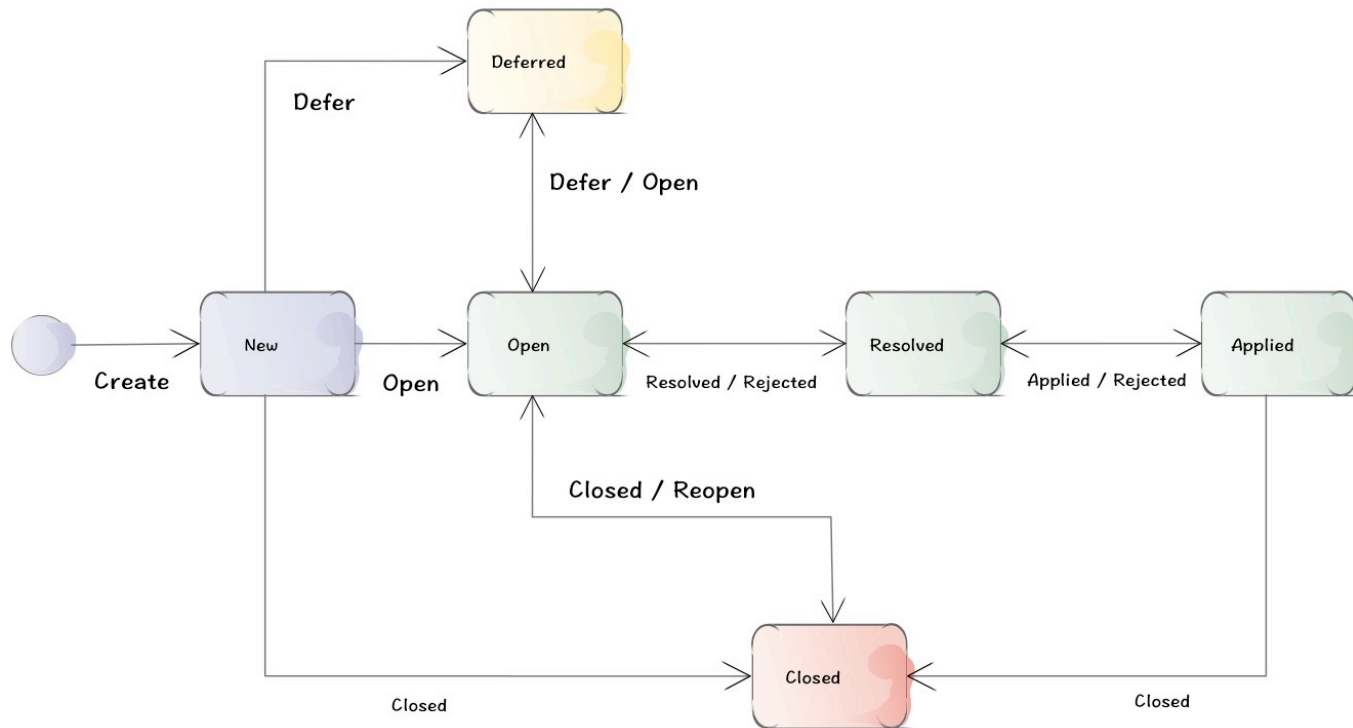
How to Get Started

- Decide what you want to track
 - Public review comments only?
 - Issues raised in committee meetings?
 - Issues from Subcommittees?

How to Get Started

- Decide how you will use status
 - JIRA status codes are: New, Open, Resolved, Applied, Closed, Deferred
 - Typical workflow:
 - New = raised but not accepted by TC
 - Deferred = not critical; awaiting review by TC
 - Open = accepted by TC but not yet addressed
 - Resolved = solution proposed by not applied
 - Applied = solution applied in work product
 - Closed = Issued addressed; no more work required
 - Issues can move back and forth between status

How to Get Started



How to Get Started

- Decide how you will assign / approve / close issues
 - E.g. one TC member to manage versus all members
 - E.g. issues are not 'real' until entered in JIRA
 - E.g. issues only closed by vote of the TC

How to Get Started

- TC Admin sets up your project & users
- TC sets up versions and components
 - Versions and components are defined by clicking “Administration” at the top of the page
- TC members add issues, descriptions, comments
 - Issues are entered by clicking “Issues / Create issue” at the top of the page

Questions?