Terms	Definitions
Bugtraq	Internal bug DB used by Sun
BugDB	Internal bug DB used by Oracle
SubCRs	Ability to associate a defect with multiple releases and track the integration of the fixes through the development life-cycle. Sub-CRs can be owned by distinct engineers and have separate evaluations. Each Sub-CR field contains information that is specific to the fix for a particular release.

Requirements Areas

OpenJDK	required for the efficient and effective development of OpenJDK implementations
Release Management	required for the management of the Java SE releases across all release families
community	required for the end developers and system administrators who use Java SE
customization	necessary to develop the necessary tools on top of the defect system
internal	required for the effective support of Java SE for Java SE customers

Priorities

P0	Essential Features – required
P1	Highly Desirable
P2	Desirable – should be implemented if straightforward
P3	Non Essential – implemente only if it requires trivial effort

Role Types

View Bugs (by far the largest group):	Anyone in the world with a web browser
Vote, watch, add comments, submit bugs:	End users, Java Developers, OpenJDK developers
Update bugs:	Java Developers who submitted the bug, OpenJDK developers
Ownership of bugs:	Open JDK Developers
Admin:	Create/Change categories, manage users, manage data , Admins

#	Priority	Requirement Area	Requirement	Bugzilla	JIRA	Notes
1	P0	configuration	Supports 1-2k logins	Y*		
2	P0	configuration	Supports 30-40k users for votes, comments etc..	Customization [1]		33,316 users have voted, watched, or commented on bugs.sun.com
3	P0	configuration	Supports large volumes of bugs, 200K to 1M	Y*		
4	P0	Release Management	Supports Sub-CRs	Customization [2]		You can link related bugs or blocking bugs, but may not the same as Sub-CRs
5	P0	configuration	Ability to scale the system as load increases	Y		
6	P0	customization	Offers Web Services APIs	Y		
7	P0	OpenJDK	Supports rich query language	Y		e.g. http://netbeans.org/bugzilla/query.cgi?format=advanced
8	P0	OpenJDK	Provides web client	Y		
9	P0	Release Management	Provides command line querying and ability to update bugs	Customization[3]		can be implemented with web service APIs
10	P0	Release Management	Ability to update bugs in bulk	?	Y	
11	P0	Release Management	Provide custom reports and summaries	Y		
12	P0	OpenJDK	Email notification for new bugs	Y		
13	P0	OpenJDK	Email notification for bug state changes	Y	Y*	
14	P0	community	Ability to look up old bugtraq bug IDs	Customization		Need URLs to be redirect and implementations
15	P0	customization	All fields are accessible with web services APIs	Y		
16	P0	internal	Auto-sync between BugDB and this system	Customization[4]		
17	P1	OpenJDK	Rich standalone client	Customization		
18	P1	internal	Mark entire bug as private or public	Y		
19	P1	internal	Must be possible to restrict access to information (customer information, login info, etc)	Y		
20	P1	OpenJDK	Ensure security bugs are not generally visible	Customization[5]		
21	P1	internal	Bugs that were private in Bugtraq remain private	Y		
22	P1	OpenJDK	User account can be private if user chooses	Y		Pick an anonymous user name
23	P1	OpenJDK	Bugs can be closed as a duplicate with link to parent bug	Y		
24	P1	Release Management	Custom category/subcategory	Y		two levels
25	P1	OpenJDK	Allow attachments to a bug of arbitrary files	Y		
26	P1	OpenJDK	Easy-to-remember and easy-to-type bug IDs	Y	?	JIRA has the project name string in front of an issue ID.
27	P1	internal	Ability to move Bugtraq bugs to new system	Customization[6]		
28	P1	internal	Retain bug history of changes for bugs that are migrated from Bugtraq	Customization[10]		
29	P1	Release Management	Allow custom fields, keywords	Y		
30	P1	Release Management	Allow an email alias to be used for email notifications	Y	Y*	
31	P1	internal	Clear guidelines are presented to end users before submitting bugs	Customization[7]		See “Before You Start” section on http://bugreport.sun.com
32	P1	OpenJDK	Support other locales	Y		
33	P1	OpenJDK	Allow different types of bugs (defect/RFE)	Y		
34	P1	OpenJDK	Search capabilities no worse than Bugtraq, full text search	Y		
35	P1	internal	Bugs are easily accessible to support groups	Customization[8]		
36	P2	community	Comments, Votes, ability to notify commenter/ voters/ watches by email	Y		
37	P2	internal	One-click to populate a bug from this system to BugDB	Customization		Add customer field to store BugDB info and populate if no BugDB ID already.
38	P2	internal	Mark a security bug and execute proper settings to bug visibility	Customization[4]		
39	P2	OpenJDK	Meets accessibility standards (such as 508 in the USA)	Y		Uses ALT text instead of images and no flashing images
40	P2	Release Management	Provide flexible reporting capabilities	Y		
41	P2	Release Management	Ability to define custom bug states	Y		
42	P2	Release Management	Set rules for obligatory and optional fields	Y		
43	P2	community	Migrate existing bugs.sun.com votes, watches, and comments as possible	Customization[9]		
44	P2	customization	Customizable submission page/template for bug creation	Y		
45	P2	OpenJDK	Ability to go from a custom boundary system to an editable form of a bug entry	Y		
46	P2	Release Management	tags/keywords	Y		
47	P2	Release Management	Ability to define standard keywords (avoids misspelling etc.)	N		
48	P1	OpenJDK	Integration with Mercurial	Y		Through extensions: FishEye for JIRA and BugzillaExtension for Bugzilla
49	P2	OpenJDK	Ability to update bugs via email	Y		Will need to determine how this will be authorized
50	P2	OpenJDK	RSS feed for bugs based on query	Y*		

51 P2	OpenJDK	Inter bug tracking (so you can reference bugs from other projects), with possible notification when such a referenced bug gets updated and/or updating other bug trackers when this bug report gets updated		Y		If another bug is referenced in a bug report, both systems will cross out the bug ID if that bug is closed
52 P2		Favorite bugs	?		Y	A view that allows login users to have a list of favorite bugs for easy access. The community group can leverage this feature to monitor the bugs they are interested in.
53 P2	OpenJDK	Integrate with patch viewer	?		Y	
54 P1	Release Management	allow bug dependency relationships		Y		
55 P1	OpenJDK	Integrate with OpenJDK people database.		Y		The bug tracker should provide a programmatic interface for adding and modifying users, so that when a contributor is added to the OpenJDK people database, the system can automatically grant appropriate rights in the bug tracker, creating a new bug-tracker account if necessary or tying to an existing account. The bug tracker must also support whatever single-sign-on solution is eventually implemented for OpenJDK.
56 P2	customization	A stable export format for bug reports (XML using a defined DTD or schema) for easy digestion and repurposing by boundary systems.		Y		Exporting is possible from query results and for backups
57 P2	customization	Variable approval workflows	?		?	Not to confuse with customizable workflow. This req means variable workflows for different releases.
58 P1	customization	Extensible hooks for triggering other events (i.e. trigger events to occur in other boundary systems or to reduce boundary systems)	Y*		Y	

Y*	We think it's possible, but we cannot find any reference in the documentations
?	We don't know if this is possible
Customizations	
[1]	This is related to the architectural design and historial vote/comment stats.
[2]	This may require some changes in the Release Management workflow to adopt the new Sub-CR equivalent/similar feature
[3]	Both systems do not have command line support. This can however be implemented with the Web Service APIs
[4]	We need to define the relationship between this system and BugDB in order to design the implementation
[5]	Security bugs should only be shown to people who need to address the issue
[6]	We need to define the mapping between Bugtraq and the new system
[7]	Sample use cases: 1) Copyrighted material should not be submitted. 2) Bugs that are submitted are not guaranteed to be fixed.
[8]	We will need one boundary system that allows support group to view/search bugs from the new system and BugDB in order to simplify their process
[9]	This includes votes/ comments made by who and when, if possible
[10]	Need to archive current Bugtraq audit trail history.

Item #	Requirements	Reference
1	Supports 1-2k logins	
2	Supports 30-40k users for votes, comments etc..	
3	Supports large volumes of bugs, 200K to 1M	http://blogs.atlassian.com/news/2007/11/introducing_wan.html
4	Supports Sub-CRs	
5	Ability to scale the system as load increases	http://confluence.atlassian.com/display/JIRA/Increasing+JIRA+Memory
6	Offers Web Services APIs	http://confluence.atlassian.com/display/REST/REST+API+Development
7	Supports rich query language	http://confluence.atlassian.com/display/JIRA/Advanced+Searching
8	Provides web client	Y
9	Provides command line querying and ability to update bugs	
10	Ability to update bugs in bulk	http://confluence.atlassian.com/display/JIRA/Modifying+Multiple+%28%27Bulk%27%29+Issues
11	Provide custom reports and summaries	http://confluence.atlassian.com/display/JIRA/Generating+Reports
12	Email notification for new bugs	http://confluence.atlassian.com/display/JIRA/Defining+a+Project
13	Email notification for bug state changes	
14	Ability to look up old bugtraq bug IDs	
15	All fields are accessible with web services APIs	Bug info is accessible through WS APIs and can be extended with REST Plugin Modules
16	Auto-sync between BugDB and this system	
17	Rich standalone client	N
18	Mark entire bug as private or public	http://confluence.atlassian.com/display/JIRA/Configuring+Issue+Level+Security
19	Must be possible to restrict access to information (customer information, login info, etc)	http://confluence.atlassian.com/display/JIRA/How+to+Set+Default+Comment+Security+Level
20	Ensure security bugs are not generally visible	
21	Bugs that were private in Bugtraq remain private	
22	User account can be private if user chooses	
23	Bugs can be closed as a duplicate with link to parent bug	http://confluence.atlassian.com/display/JIRA/Linking+Issues
24	Custom category/subcategory	http://confluence.atlassian.com/display/JIRA/Defining+a+Component
25	Allow attachments to a bug of arbitrary files	http://confluence.atlassian.com/display/JIRA/Attaching+a+File
26	Easy-to-remember and easy-to-type bug IDs	All numbers
27	Ability to move Bugtraq bugs to new system	
28	Retain bug history of changes for bugs that are migrated from Bugtraq	
29	Allow custom fields, keywords	http://confluence.atlassian.com/display/JIRA/Adding+a+Custom+Field http://confluence.atlassian.com/display/JIRA/Labelling+an+Issue
30	Allow an email alias to be used for email notifications	
31	Clear guidelines are presented to end users before submitting bugs	
32	Support other locales	Depends on the database unicode support
33	Allow different types of bugs (defect/RFE)	http://confluence.atlassian.com/display/JIRA/Creating+an+Issue http://confluence.atlassian.com/display/JIRA/Defining+%27Issue+Type%27+Field+Values
34	Search capabilities no worse than Bugtraq, full text search	Lucene http://blogs.atlassian.com/rebelutionary/downloads/tssjs2007-lucene-generic-data-indexing.pdf
35	Bugs are easily accessible to support groups	
36	Comments, Votes, ability to notify commenter/ voters/ watches by email	http://confluence.atlassian.com/display/JIRA/Watching+and+Voting+on+an+Issue
37	One-click to populate a bug from this system to BugDB	
38	Mark a security bug and execute proper settings to bug visibility	
39	Meets accessibility standards (such as 508 in the USA)	Use alt text instead of images, no flashing images
40	Provide flexible reporting capabilities	http://confluence.atlassian.com/display/JIRA/Generating+Reports
41	Ability to define custom bug states	http://confluence.atlassian.com/display/JIRA/Activating+Workflow
42	Set rules for obligatory and optional fields	http://confluence.atlassian.com/display/JIRA/Specifying+Field+Behaviour#SpecifyingFieldBehaviour-Required%2FOptionalfields
43	Migrate existing bugs.sun.com votes, watches, and comments as possible	
44	Customizable submission page/template for bug creation	http://confluence.atlassian.com/display/JIRA/Customising+the+Look+and+Feel
45	Ability to go from a custom boundary system to an editable form of a bug entry	
46	tags/keywords	http://confluence.atlassian.com/display/JIRA/Labelling+an+Issue
47	Ability to define standard keywords (avoids misspelling etc.)	http://confluence.atlassian.com/display/JIRA/Labelling+an+Issue
48	Integration with Mercurial	http://confluence.atlassian.com/display/JIRA/Integrating+with+a+Source+Control+System
49	Ability to update bugs via email	http://confluence.atlassian.com/display/JIRA/Creating+Issues+and+Comments+from+Email
50	RSS feed for bugs based on query	http://forums.atlassian.com/thread.jspa?messageID=257253550&#257253550

51	Inter bug tracking (so you can reference bugs from other projects), with possible notification when such a referenced bug gets updated and/or updating other bug trackers when this bug report gets updated	
52	Favorite bugs	http://confluence.atlassian.com/display/JIRA/Customising+the+Dashboard
53	Integrate with patch viewer	http://confluence.atlassian.com/display/JIRA/Viewing+an+Issue%27s+FishEye+Changesets
54	allow bug dependency relationships	http://confluence.atlassian.com/display/JIRA/Configuring+Issue+Linking
55	Integrate with OpenJDK people database.	
56	A stable export format for bug reports (XML using a defined DTD or schema) for easy digestion and repurposing by boundary systems.	http://confluence.atlassian.com/display/JIRA/Automating+JIRA+Backups http://confluence.atlassian.com/display/JIRA/Displaying+Search+Results+in+XML
57	Variable approval workflows	http://confluence.atlassian.com/display/JIRA/Configuring+Workflow
58	Extensible hooks for triggering other events (i.e. trigger events to occur in other boundary systems or to reduce boundary systems)	http://confluence.atlassian.com/display/JIRA/Adding+a+Custom+Event http://confluence.atlassian.com/display/JIRA/Adding+a+Custom+Event

Item #	Requirements	Reference
1	Supports 1-2k logins	
2	Supports 30-40k users for votes, comments etc..	
3	Supports large volumes of bugs, 200K to 1M	
4	Supports Sub-CRs	
5	Ability to scale the system as load increases	http://www.bugzilla.org/docs/4.0/en/html/multiple-bz-dbs.html http://web.archiveorange.com/archive/v/lRMFny1fCPtm9DvJ2OJx
6	Offers Web Services APIs	https://wiki.mozilla.org/Bugzilla:WebService
7	Supports rich query language	http://www.bugzilla.org/docs/4.0/en/html/query.html https://landfill.bugzilla.org/bugzilla-tip/query.cgi?format=advanced
8	Provides web client	
9	Provides command line querying and ability to update bugs	
10	Ability to update bugs in bulk	
11	Provide custom reports and summaries	http://www.bugzilla.org/docs/4.0/en/html/reporting.html
12	Email notification for new bugs	http://www.bugzilla.org/docs/tip/en/html/parameters.html
13	Email notification for bug state changes	http://www.bugzilla.org/docs/tip/en/html/parameters.html
14	Ability to look up old bugtraq bug IDs	
15	All fields are accessible with web services APIs	XML & JSON
16	Auto-sync between BugDB and this system	
17	Rich standalone client	
18	Mark entire bug as private or public	http://www.bugzilla.org/features/#groups
19	Must be possible to restrict access to information (customer information, login info, etc)	
20	Ensure security bugs are not generally visible	
21	Bugs that were private in Bugtraq remain private	
22	User account can be private if user chooses	
23	Bugs can be closed as a duplicate with link to parent bug	http://www.bugzilla.org/docs/tip/en/html/bug_status_workflow.html
24	Custom category/subcategory	http://www.bugzilla.org/docs/tip/en/html/components.html
25	Allow attachments to a bug of arbitrary files	http://www.bugzilla.org/docs/tip/en/html/attachments.html
26	Easy-to-remember and easy-to-type bug IDs	All numbers
27	Ability to move Bugtraq bugs to new system	
28	Retain bug history of changes for bugs that are migrated from Bugtraq	
29	Allow custom fields, keywords	http://www.bugzilla.org/docs/tip/en/html/custom-fields.html
30	Allow an email alias to be used for email notifications	http://www.bugzilla.org/docs/tip/en/html/parameters.html#param-email
31	Clear guidelines are presented to end users before submitting bugs	
32	Support other locales	http://www.bugzilla.org/features/#unicode
33	Allow different types of bugs (defect/RFE)	http://www.bugzilla.org/docs/tip/en/html/bugreports.html
34	Search capabilities no worse than Bugtraq, full text search	Boolean Charts http://www.bugzilla.org/docs/2.22/html/query.html
35	Bugs are easily accessible to support groups	

		http://www.bugzilla.org/docs/tip/en/html/userpreferences.html http://www.bugzilla.org/docs/tip/en/html/voting.html
36	Comments, Votes, ability to notify commenter/ voters/ watches by email	http://www.bugzilla.org/docs/tip/en/html/hintsandtips.html#commenting
37	One-click to populate a bug from this system to BugDB	
38	Mark a security bug and execute proper settings to bug visibility	
39	Meets accessibility standards (such as 508 in the USA)	use alternative text instead of images; no flashing images
40	Provide flexible reporting capabilities	http://www.bugzilla.org/docs/4.0/en/html/query.html
41	Ability to define custom bug states	http://www.bugzilla.org/docs/tip/en/html/bug_status_workflow.html
42	Set rules for obligatory and optional fields	http://www.bugzilla.org/docs/tip/en/html/products.html#group-control-examples
43	Migrate existing bugs.sun.com votes, watches, and comments as possible	
44	Customizable submission page/template for bug creation	http://www.bugzilla.org/docs/tip/en/html/cust-templates.html
45	Ability to go from a custom boundary system to an editable form of a bug entry	
46	tags/keywords	http://www.bugzilla.org/docs/tip/en/html/keywords.html
47	Ability to define standard keywords (avoids misspelling etc.)	N
48	Integration with Mercurial	http://mercurial.selenic.com/wiki/BugzillaExtension
49	Ability to update bugs via email	http://www.bugzilla.org/features/#email-in
50	RSS feed for bugs based on query	http://www.bugzilla.org/releases/2.20/new-features.html#rss
51	Inter bug tracking (so you can reference bugs from other projects), with possible notification when such a referenced bug gets updated and/or updating other bug trackers when this bug report gets updated	
52	Favorite bugs	
53	Integrate with patch viewer	http://www.bugzilla.org/docs/4.0/en/html/attachments.html
54	allow bug dependency relationships	http://www.bugzilla.org/docs/4.0/en/html/hintsandtips.html
55	Integrate with OpenJDK people database.	
56	A stable export format for bug reports (XML using a defined DTD or schema) for easy digestion and repurposing by boundary systems.	http://www.bugzilla.org/docs/2.16/html/faq.html#faq-phb-data
57	Variable approval workflows	http://www.bugzilla.org/docs/4.0/en/html/lifecycle.html
58	Extensible hooks for triggering other events (i.e. trigger events to occur in other boundary systems or to reduce boundary systems)	