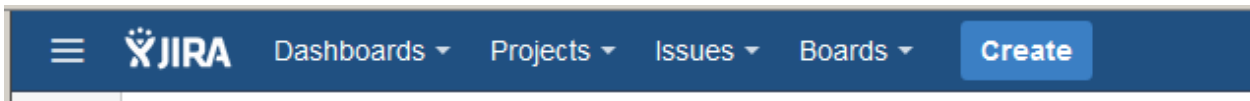


Change History

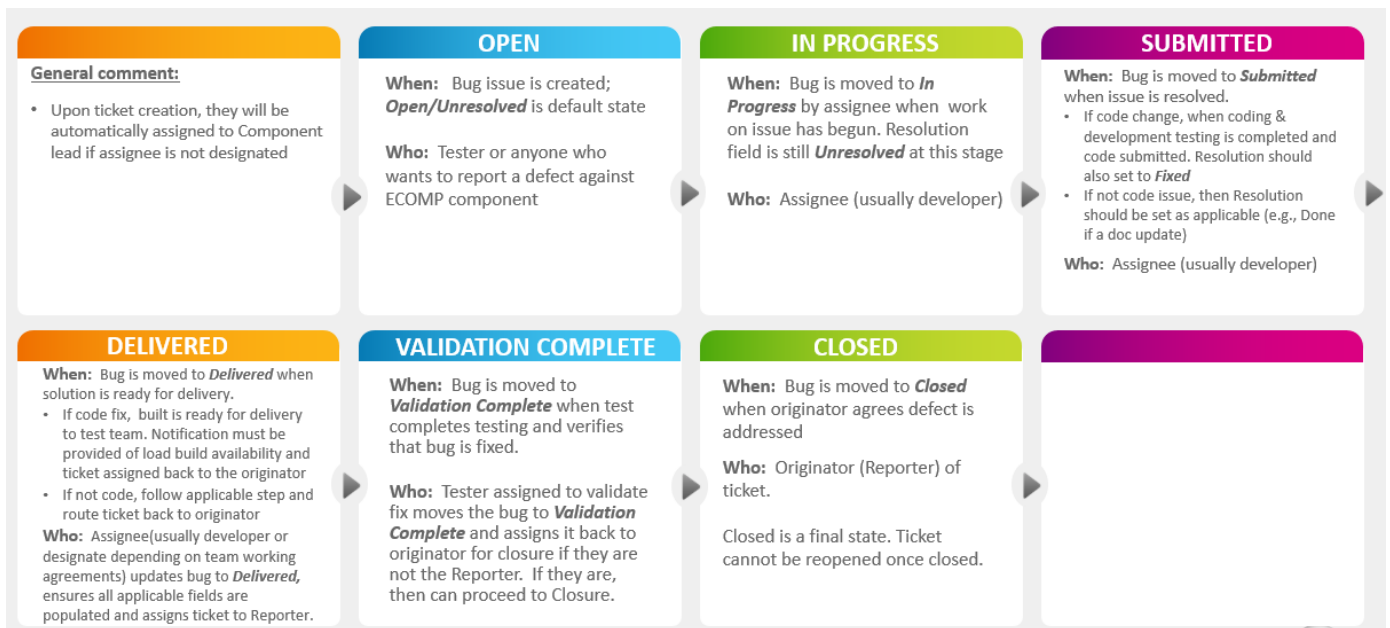
Version	Date	Prime	Description
1	12/12/16	Rx196w	Initial Release
2	12/14/16	Rx196w	Minor editorial corrections
3	12/19/16	Rx196w	Resolution field is now a required field in all screens; User will need to manually set it to Unresolved upon creation of ticket.

Create Bug Ticket

To create a Bug ticket, once logged into Jira, click on the Create option at the top of your screen


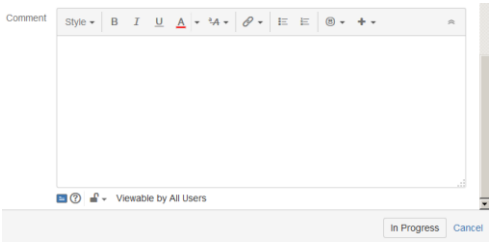









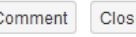



This will bring up the Create screen, it's pretty intuitive once you see this, but table below will guide you on what fields need to be populated. Fields with a red asterisk (*) are mandatory. If you don't see all the fields, check the Configure fields selected at the top right of the screen and ensure all are selected.

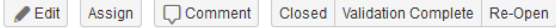


Workflow and fields to populate

Jira Status	What Does It Mean?	Who Updates It?	Fields that must be populate
Step 1: Open Bug via Create option, populate fields noted below, and confirm creation			
Open	This is the default status when a Bug is initially created in Jira, the originator is presented with Create Issue screen shown above.	Originator of ticket (Tester)	<p>The originator of the ticket must provide thorough details regarding the defect. This includes the following (fields with a red asterisk (*) are mandatory):</p> <ul style="list-style-type: none"> *Project – if you have access to multiple projects *Issue Type (Bug) – make sure you select correct Issue Type *Summary – provide concise/short descriptive summary of the issue Description – detailed description of the issue, with enough information needed by developer to be able to reproduce issue. *Component – identify where the defect is – this will allow proper assignment of the defect Severity and Priority – identify the urgency of the issue – make sure it makes sense, for example a Sev 1 issue with a priority of low or trivial does not make sense. Environment – identify the test environment in which the issue was found

Jira Status	What Does It Mean?	Who Updates It?	Fields that must be populate
			<ul style="list-style-type: none"> • Assignee – if you know who to assign the ticket to, go ahead and enter this person’s name. They must be a valid user in the Jira project. If you don’t know who to assign it to, leave this field blank; ticket will automatically be assigned to the component lead associated with the identified component on the ticket. • Affects Version – identifies the software version in which the bug was found in. Very important to identify this so that developer knows which release the defect is being reported against. • Component – identifies where the defect is – this will allow proper assignment of the defect • Attachment - Attach applicable logs, screen shot, etc... anything that would be helpful to the developer • Test Case ID – if the issue is found in the execution of a test case, enter test case ID • *Resolution – upon initial creation, this this field to Unresolved
Step 2: Bug is now assigned to a developer and work is starting on it.			
In Progress	<p>The Status field is updated from Open to In Progress when the developer has started work on the issue.</p>	<p>Assignee of Bug issue (Developer)</p>	<p>Developer moves issue from Open to In Progress when they have started working on it by selecting the In Progress option:</p>  <p>Developer will be presented with the In Progress screen. Developer can add comments if needed and confirm change in status at the bottom of the screen.</p> 
Step 3: Developer has identified the fix for the issue and is now ready to submit			

Jira Status	What Does It Mean?	Who Updates It?	Fields that must be populate
Submitted	The Status field is updated from In Progress to Submitted when the fix for the defect is submitted into the code repository and ready for a build	Assignee of Bug issue (Developer)	<p>Developer moves issue from In Progress to Submitted when they have finished their coding/unit testing and are ready to submit to code repository to be picked up in next build. Selecting the Submit option will present the Submit screen to the developer:</p> <p>    </p> <p>Ensure following fields are populated:</p> <ul style="list-style-type: none"> • *Resolution – the Resolution field is a required field in the Submit screen. Several options are available, select the one that is applicable to your situation. In the case of bug fixes, the Resolution would be Fixed • Fix Version/s – identify the release version of the software in which the fix is provided • Build Number – if known at this stage, enter the build number; otherwise, leave blank and enter at next step (when ticket is marked Delivered) • Comments - Ensure that appropriate details are entered in the Bug ticket about the nature of the fix. This will help the tester or anyone else looking at the issue to understand the nature of the problem and resolution; identify if additional regression is needed for retest, etc.... <p>Please note that many other scenarios may exist, but we are focusing on the code defect here.</p>
Step 4: Fixes have been merged and software build is ready for delivery to test team			
Delivered	The Status field is updated from Submitted to Delivered when the software build is ready for delivery to the test team	Assignee of Bug issue (Developer)	<p>Developer moves issue from Submitted to Delivered when the software build is available for testing teams to deploy to their lab environment. At this stage, the Jira bug should be properly updated with needed details and Bug Issue assigned back to test team (originator of ticket).</p> <p>Selecting the Deliver option will present the Deliver screen to the developer:</p> <p>     </p> <p>Ensure following fields are populated:</p>

Jira Status	What Does It Mean?	Who Updates It?	Fields that must be populate
			<ul style="list-style-type: none"> • *Resolution – the Resolution field is a required field in the Submit screen. You have an opportunity to update it again in the Deliver screen if applicable; otherwise, no further action is needed. • Fix Version/s – identify the release version of the software in which the fix is provided if not already populated • Build Number – enter the build number. If one was entered when ticket was moved to Submit, validate that it's correct. • Assignee – assign ticket back to the originator or to the person that must test the fix.
Step 5: Test team has tested the fix and confirm that it's working as expected			
Validation Complete	The Status field is updated from Delivered to Validation Complete when testing has successfully completed.	Originator of ticket (Tester)	<p>Tester moves issue from Delivered to Validation Complete after they successfully validate the bug fix.</p> <p>Selecting the Validation Complete option will present the Validation Complete screen to the tester:</p>  <p>Ensure following fields are populated/updated as applicable.</p> <ul style="list-style-type: none"> • Comments – Update comments with details of testing and results. • Test Case ID – Update test case ID if applicable • Assignee – assign ticket back to the originator if different from current assignee
Step 5: All done, ticket is closed			
Closed	The Status field is updated from Validation Complete to Closed .	Originator of ticket (Tester)	<p>Tester moves issue from Validation Complete to Closed after they successfully validate the bug fix.</p> <p>Closed is a final state; one ticket is closed, it cannot be moved back to any previous state.</p> <p>If the person who validated the ticket is different than the person who originated the ticket, then the originator should close the ticket after they review the test results.</p>

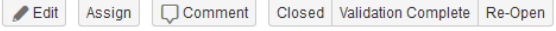
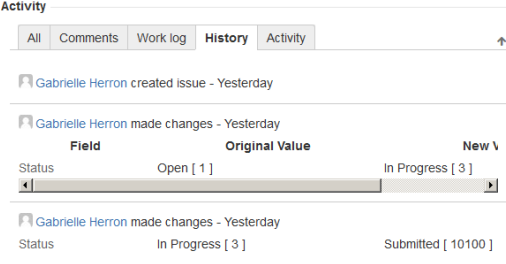
Bug: Use Case 2: Test originated bug, Code fix provided, Retest Failed

The general workflow for Use Case will be:

- **Open -> In Progress -> Submitted -> Delivered -> Re-Open (Status Open, Resolution Fix Failed)**

In this use case, the workflow is the same as above, everything proceeds the same up through the Delivered state. At this point, the ticket is Re-opened with Resolution of Fixed Failed and assigned back to the developer and the cycle starts back from Open state.

General Flow:

Jira Status	What Does It Mean?	Who Updates It?	Fields that must be populate
Steps 1 – 4 : Same as for Use Case 2			
Step 5: Test team has tested the fix and the fix Failed			
Open	The Status field is updated from Delivered back to Open and Resolution is marked as Fix Failed.	Originator of ticket (Tester)	<p>Tester moves issue from Delivered back to Open by selecting the Re-Open option:</p>  <p>Ensure following fields are populated/updated as applicable.</p> <ul style="list-style-type: none"> • Comments – Update comments with details of testing and results. • Attachment - Attach applicable logs, screen shot, etc... anything that would be helpful to the developer • Test Case ID – Update test case ID if applicable • Assignee – assign ticket back to the Developer. If you're not sure who that is, look in the History of the ticket and see who move ticket to Submitted state. For example: <p>Activity</p> 
Jira Issue goes through same flow starting with Step 1 again.			