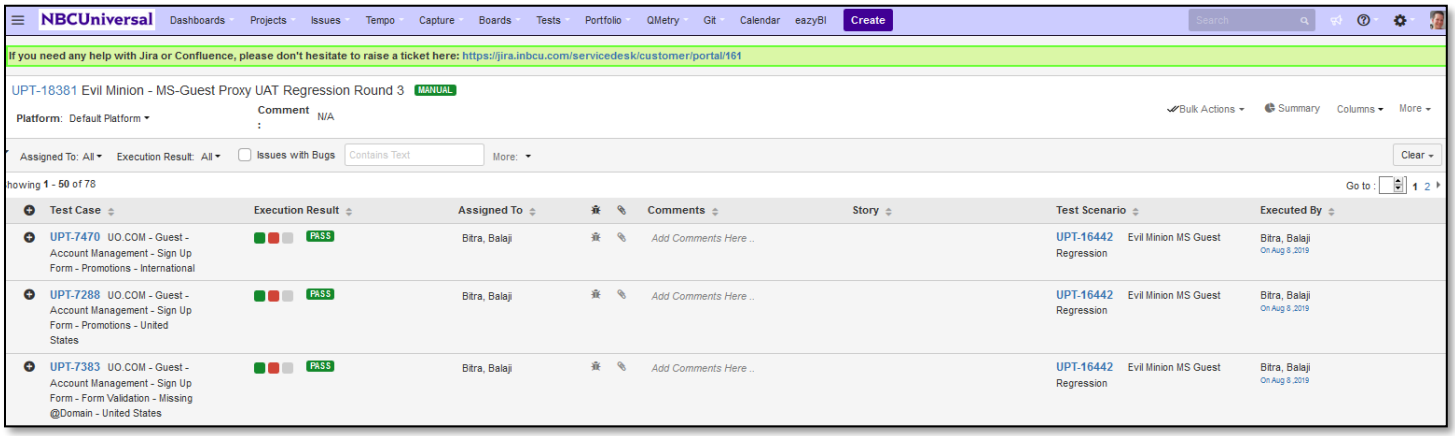
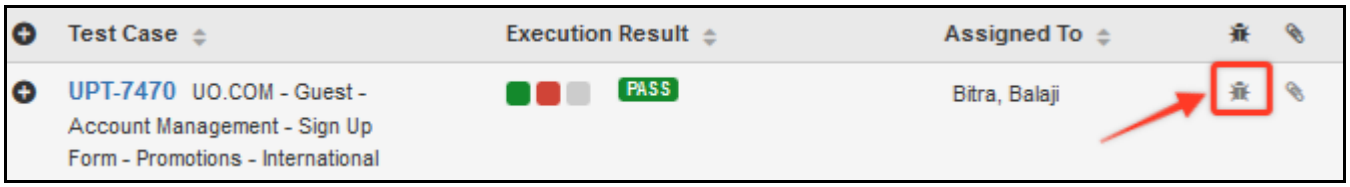
1. Documenting a Test Run

When executing a Test Run in Jira, all relevant field need to be filled out to provide detail of QE testing as well as testing evidence for CAB reviews. Below is a list of fields or functions within the Test Execution screen that will need to be filled out or populated for testing evidence.

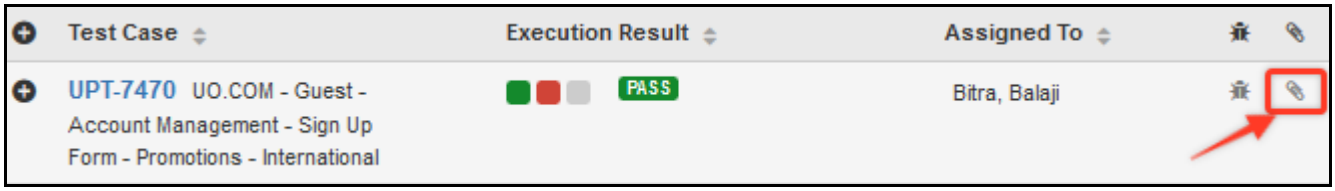
❖ Test Case Level



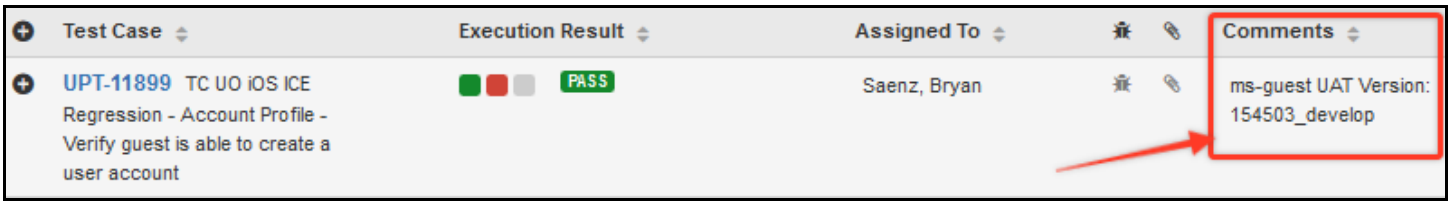
- **Test Case (Mandatory):** This field displays the name of the Test Case that is to be executed. This field should be 100% unique and never the same as another test case. This title should represent what you are testing at a high level. This is very important when exporting the Test Run results and presenting them to other teams or business stakeholders.
- **Execution Result (Mandatory)**: The default value for this field is Not Executed. Once the test has been executed the Result should be update to Pass, Failed or Blocked. This is a high level representation of what was found during the step by step test execution. A test will only Pass if ALL of the steps are Passed. If any step is Blocked or Failed, then the Test Case level Execution Results should represent the Failed or Blocked status as well.
- **Assigned To:** The Assigned To field will represent the user who is assigned to execute the test.
- **Bug (Icon):** This is a single-click button that will launch the 'Assign Bugs' pop-up screen. All Bugs that are assigned in individual steps should also be assigned here as well. Once the Bug(s) has been added, select the 'Close' button in the bottom right of the 'Assign Bugs' pop-up screen to dismiss the pop-up. At this time, you will see that the Bug icon is now red.
    - **Assign Bug Screen:** You can search for an existing bug in the search box or add (create) a new bug "on the fly" with the 'Add Bugs' button.



    - Once a Bug is added, the icon will turn red: 
- **Attachments (Paperclip icon):** Here you will attach any testing documentation that supports your testing. This will mainly be used for adding screen shots and testing evidence for In-Sprint Functional testing.

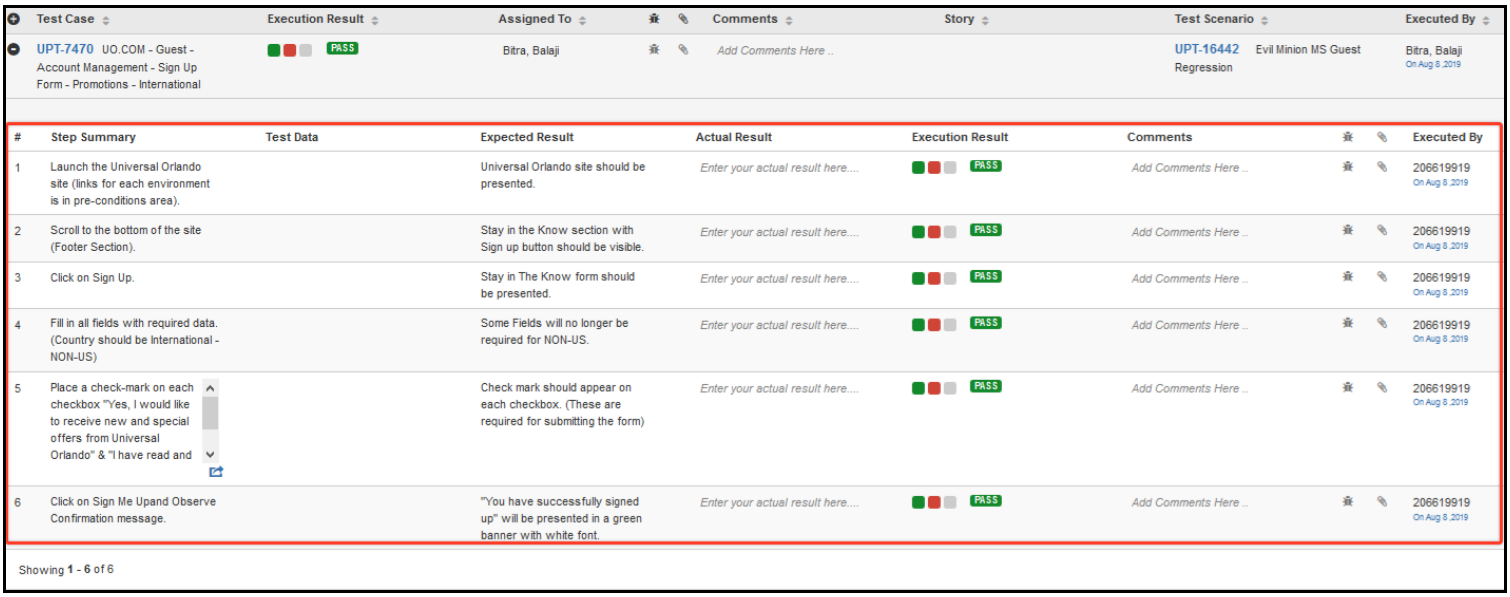


- **Comments (Mandatory):** The Comments box, at the Test Case level, is for important but brief information about that test case or testing status. Examples:
  - Why is the Test Case Blocked
  - What is the Version Number tested on the application; Mobile Builds, Micro Services Version, .NET Services, Web UI build number.
  - Brief statement as to why the test case failed: "Cannot create a new guest account"
  - When sending data to BOH for Financial Validation, enter all of your test execution data into the comment section (i.e. Account Credentials, Confirmation Numbers, and any additional information needed for the financial team).



- **Story:** This field will be pre-populated for Test Cases that have been assigned to a Story during In-Sprint testing.
- **Test Scenario:** This field will be pre-populated with the Test Scenario name, if the Test Case is being pulled into the Test Run via a Test Scenario.
  - In Jira, Test Scenarios is a set of test cases that are grouped together for re-usability and importing into a Test Run.
- **Executed By:** This field will be populated and updated automatically when a user makes a change on the Test Case or Test Step level.

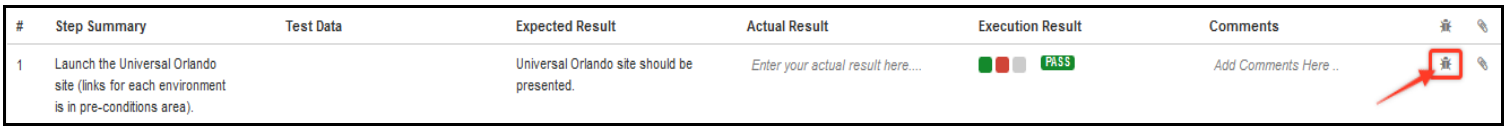
❖ Test Step Level



- **Step Summary (Mandatory):** This is a list of the Test Steps from the Test Case. All Test Cases should have Test Steps to execute.
- **Test Data:** This field will have a value if the Test Data was entered in the Test Case.

- **Examples:** Login credentials for a remote machine, Galaxy Node login, POS System.

- **Expected Results (Mandatory):** This is what the user is executing to see or experience when validating the action Test Step. All Test Steps must have an Expected Result.
  - Be explicit in what you are expected to see; think about a machine is testing.
  - Don't use assumptive words (i.e. should, would, if, or maybe).
- **Actual Results:** This field is comment box to explain any deviation from the Expected Result. If a Test Step fails or is Blocked, this field is **required** to be filled out and explain what happened.
- **Execution Results (Mandatory):** This field has the same features as the Execution Results at the Test Case level. Use this field to represent the status of each Test Step.
  - All Test Steps must have a value of Pass/Failed/Blocked. The only time a Test Step will have the Result of Not Executed (after execution) is if a step above it has Failed or Blocked progression in the test.
  - Use the 'Work In Progress' status for multi-day tests (Example: Testing ACI or account lockout).
- **Comments:** This field is an open text area for important information pertaining to the actual test step. Examples:
  - Order Confirmation Number
  - User Account Create for the test
  - If Blocked, what is blocking the step
  - If there is an acceptable deviation from the Test Step (i.e. available Test Data; validating Virtual Line on Fast and Furious instead of Jimmy Fallon).
- **Bug (Icon):** This is a single-click button that will launch the 'Assign Bugs' pop-up screen. All Bugs that are assigned to this individual step should also be assigned here.
  - Once the Bug(s) has been added, select the 'Close' button in the bottom right of the 'Assign Bugs' pop-up screen to dismiss the pop-up. At this time, you will see that the Bug icon is now red.
  - **Assign Bug Screen:** You can search for an existing bug in the search box or add (create) a new bug "on the fly" with the 'Add Bugs' button.

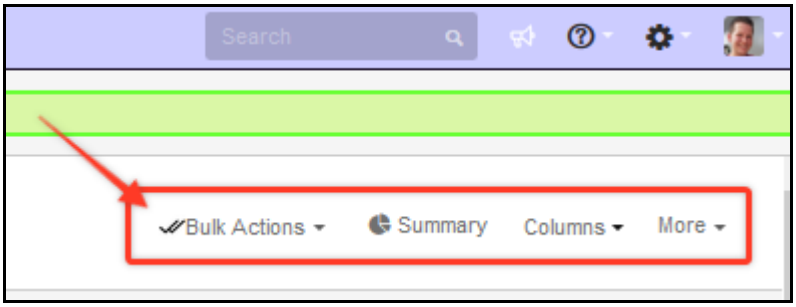


  - Once a Bug is added, the icon will turn red: 

- **Attachments (Paperclip icon):** Here you will attach any testing documentation that supports your testing. This will mainly be used for adding screen shots and testing evidence for In-Sprint Functional testing. The Test Step level attachments are only needed if the team feels that they are necessary at the step level. Most teams will use the Test Case level Attachment button to upload testing evidence documents.

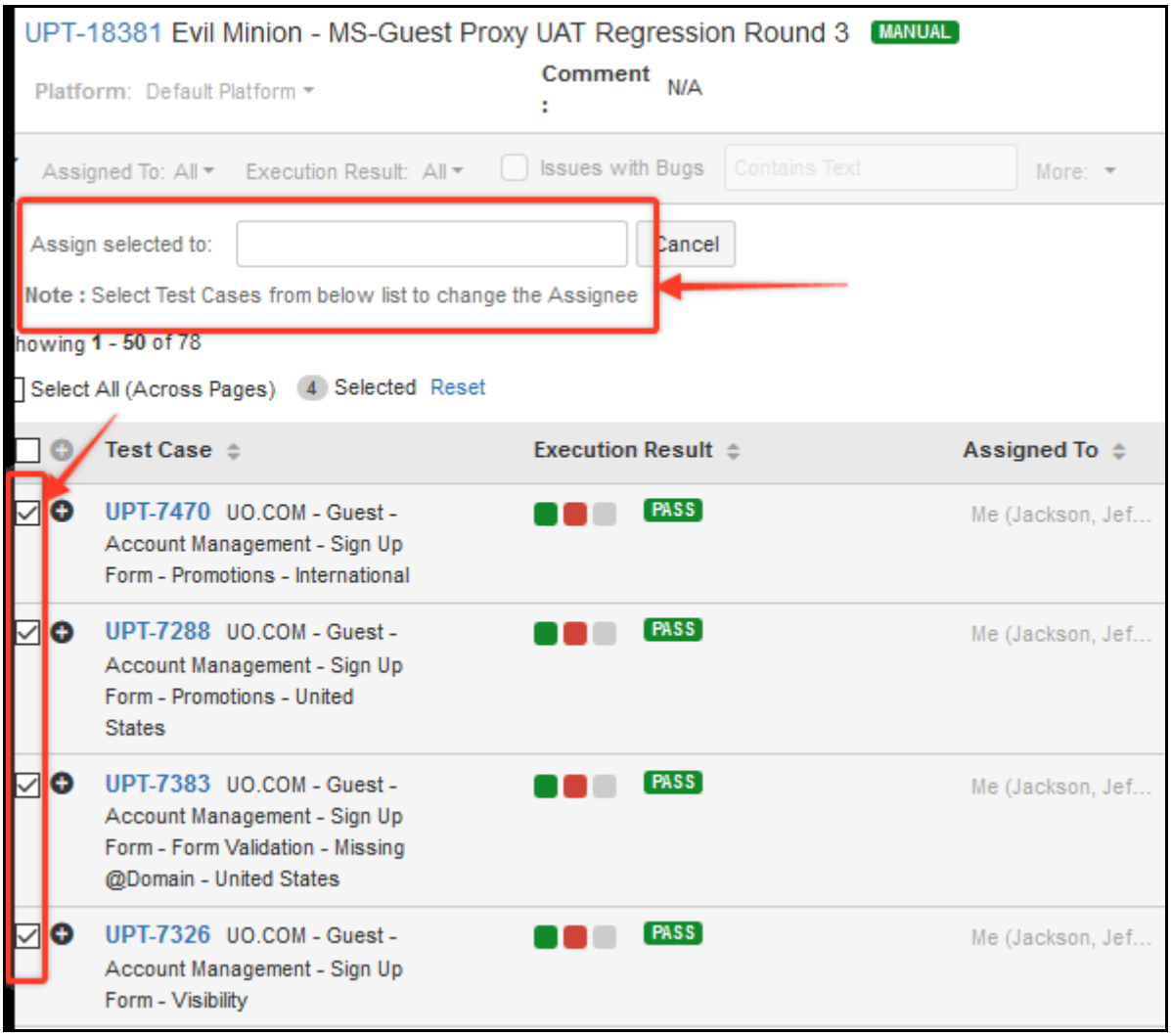
- **Executed By:** This field will be populated and updated automatically when a user makes a change on the individual Test Step.

2. Additional Test Run Controls



❖ **Bulk Actions Control**

- Assign Tester: Select this action if you would need to add a tester as the Assignee to multiple Test Cases.
  - One you select Bulk Actions > Assign Tester, check boxes will appear to all of the test cases.
  - Select the Test Cases that you want to reassign.
  - Then, in the 'Assign selected to:' text box, start entering a name. The users will start to auto-fill.
  - Select the desired name from the drop down and the test cases will be reassigned.



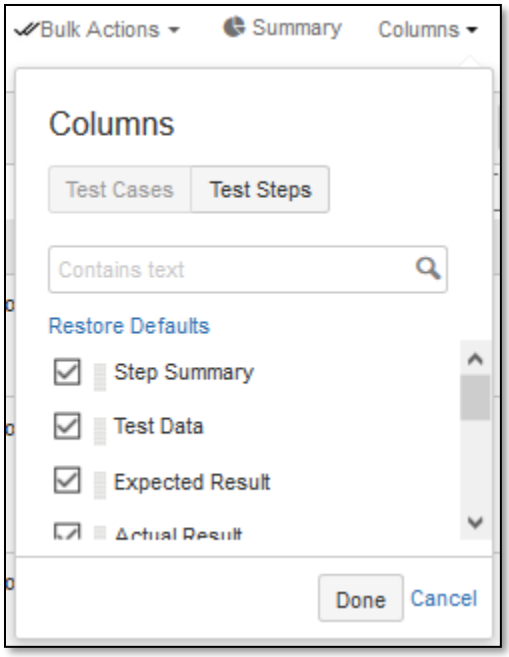
- Execution Result: This function is to bulk update the Execution Results on the Test Case level. This will not be used as all QEs will be manually marking Execution Results on the Test Case **and** Test Step level.
- Refresh Test Cases: This function allows the user to update a test case that has been edited after it was pulled into the Test Run.
  - To Execute this function, select the Bulk Actions > Refresh Test Cases option.
  - Upon selection, check boxes will appear to the left of all of the test cases.

- Select the predetermined Test Case(s) that needs to be refreshed.
- Once the selections have been made, select the 'Apply' button above the Test Case column.
- Once applies, the selected Test Case(s) will be refreshed with the updated content.
- **Note**: Refreshing a Test Case **<u>will not</u>** clear or change the Execution Status.

❖ Summary

- Select the Summary button to expand the additional tabs that give a high level report of the Test Execution.
  - Summary Tab: This tab displays the total number of associated Stories, Test Cases, Test Scenarios, and Bugs.
  - Test Case Tab: This tab displays a percentage of Execution Status with a matching bar chart.
  - Bug Tab: This tab displays a count of the different Bug Status and Severity, as well as a matching bar chart.

❖ Columns



- The Columns drop down button gives the user the option to control the columns that are listed in the Test Case row and the Test Step rows.
- If you deselect a column it will be removed from the UI view as well as removed from the export, if the Test Execution Results are exported to an excel file.

❖ More

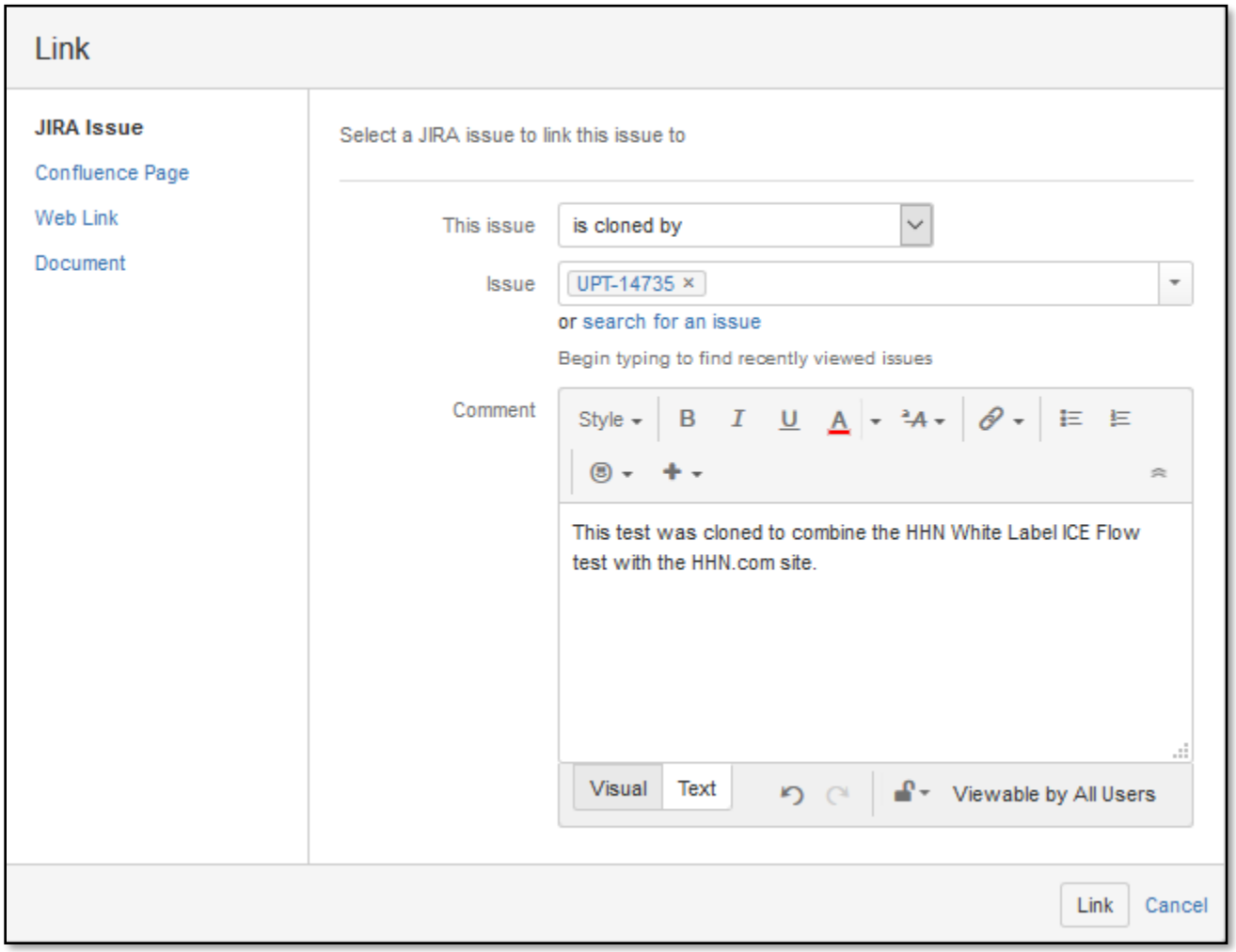
- Rebuild: This function is designed to "refresh" the whole Test Run.
  - When a test case has been added or removed from the Test Run, you will have to Rebuild the Test Run to be able to see the change reflected.
  - This is the same for adding/removing Test Cases to Test Scenarios, and adding/removing Test Scenarios to the Test Run.

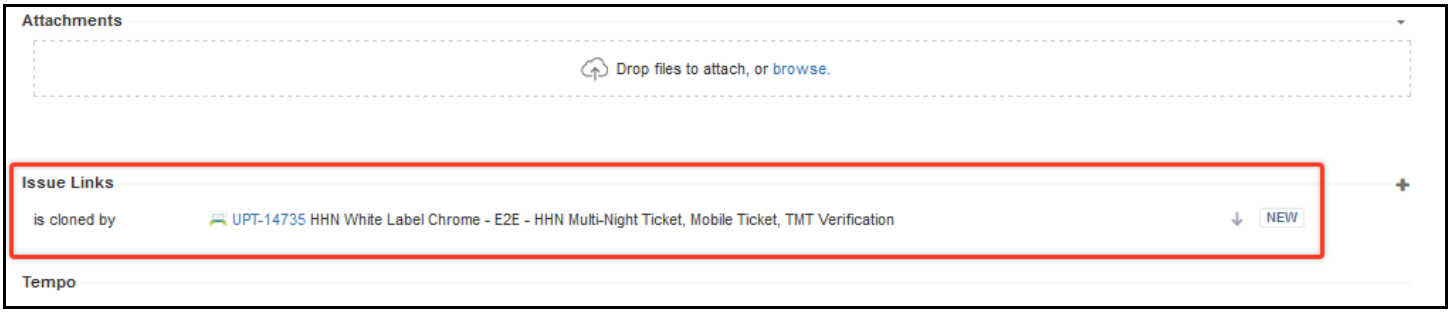
- The Rebuild function can also be used to Refresh all associated test cases the same as the 'Refresh' function from the Bulk Actions drop down.
- To execute this function, select More > Rebuild > Confirmation pop-up displays > Select Rebuild.
- **Note**: Rebuilding a Test Run **will not** clear or change the Execution Status.

➢ Exporting Test Run Results: This is used to provide testing evidence to CAB, as well as passing testing results to the BOH Finance team to continue specific testing.

- To Export the Test Run Results, navigate to the Test Execution page in Jira.
- Once the page is loaded, select the More drop down in the top right corner, and select the 'Export' option.
- You will then be prompted to Open or Save the excel file of the export.
- This export will display all of the columns, fields, and data that are shown in the Jira Test Execution screen.

3. Cloning Test Cases





- When cloning a test case to be shared with another team, you will need to Link the two test cases to show the association. In the Link window of the cloned Test Case, select the value of "This issue **is cloned by.**" Then select the Issue number of the original Test Case.
- This association is particular useful for teams like Digital Underground (App 2.0) and Mobile Core.
    - Once the new functionality of "App 2.0" goes to Production, the Test Cases that would normally be Regression are passed too and consumed by the Mobile Core team.
    - The Digital Underground team will keep an original copy of their Test Case as a history of effort on their Stories, as well as a history of functional requirements.
    - Additionally, the Mobile Core team would Clone a copy of the Regression-eligible Test Cases and move them to the Mobile Core Regression bed.