

Test Case Management

- 1. Introduction
- 2. Pre-requisites
- 3. Navigation To:
 - 3.1 Test Cases
 - 3.2 Test Cycles
 - 3.3 Test Plans
 - 3.4 Test Reports
- 4. Folder Structure for Test Cases
- 5. Creation of Test Cases under Folder Structure
 - 5.1 Add Test Cases/ Steps Features
 - 5.1.1 Add Step
 - 5.1.2 Call to Test
 - 5.1.3 Hide/Show Steps
 - 5.1.4 Add Parameter
 - 5.2 Version Control to Update Test Cases
 - 5.3 BDD Test Cases
 - 5.4 Data Driven to Parameterize Test Cases
- 6. Test Cycle/ Test Execution
 - 6.1 Folder Structure in Test Cycle
 - 6.2 Create a Test Cycle
 - 6.3 Add Test Cases to Test Cycle
 - 6.4 Clone Test Cycle
 - 6.5 Run Test Cycle / Execute Test Cases
- 7. Reporting and Metrics
- 8. Review and update test cases regularly.

1. Introduction

- **Purpose:** To define the process for creating and executing test cases using Zephyr in Jira.
- **Scope:** Applies to all QA team members involved in manual or automated testing.

2. Pre-requisites

- Jira access with Zephyr plugin installed. (Search Zephyr from App)
- Appropriate project permissions to create and execute test cases.
- Familiarise with Jira issue types and workflows.

3. Navigation To:

- Go to your Jira project.
- Select '**Zephyr**' from the left-hand panel under '**App**'.
- You can see 4 tabs below on right hand panel:
 - Test Cases
 - Test Cycles
 - Test Plans
 - Reports

The screenshot shows the Jira sidebar with the 'Apps' section open. The 'Zephyr' app is selected and highlighted with a red box. The main content area displays the Zephyr interface with the 'Test Cases' tab selected, also highlighted with a red box. Below the tabs, there is a search bar and a list of test cases categorized by application.

Below is high level information of all tabs, we see detailed information in next section.

3.1 Test Cases

- It is used to create new Test cases and by default ‘**Test Cases**’ tab is selected and visible.
- Under Test Cases tab, can see option as ‘**All Test Cases**’ to see all existing test cases.
- Can see ‘**New Test Cases**’ button to create new test cases.
- Under ‘**More**’ can see options ‘**Create test case in Bulk**’ to create multiple test cases at a time and option ‘**Import from file**’ to import test cases from file.

The screenshot shows the 'Test Cases' tab in the Zephyr app. The '+ New Test Case' button is highlighted with a red box. A dropdown menu is open, showing various actions such as 'Create test cases in bulk', 'Edit selected test cases in bulk', 'Import from file', and 'Export to XML'.

3.2 Test Cycles

- Can create Test Cycles for Test case execution.

3.3 Test Plans

- Can create Test Plan for our Project and Releases.

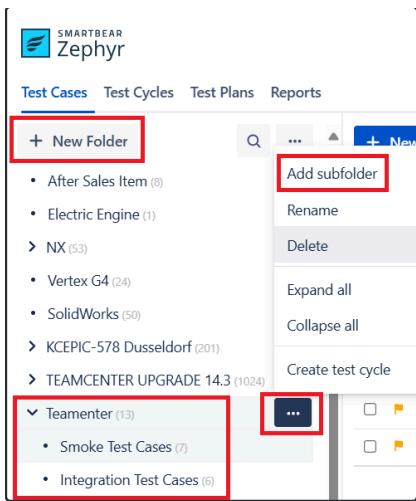
3.4 Test Reports

- Can see many types of useful report related to Testing result.

Note: Go to Project setting → App → Zephyr and check the Permission for Test Cases, Test Cycles, Test Plans, Test Executions, Reports and Configurations.

4. Folder Structure for Test Cases

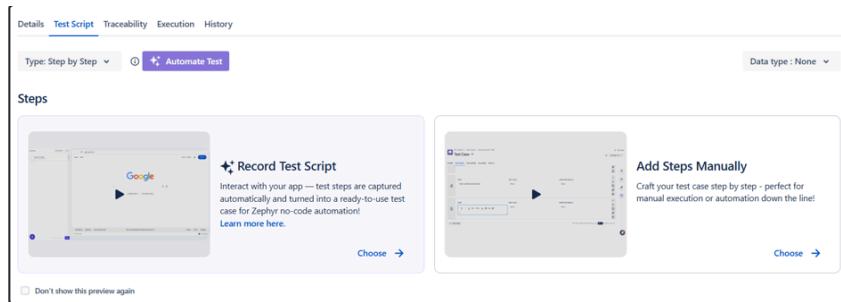
- Under ‘**Test Cases**’ tab, click on ‘**+ New Folder**’ option to create new folder.
- Add folders based on application. (e.g.: Teamcenter Folder)
- Add subfolder under application folder by clicking ‘...’ icon next to created folder.
- Add subfolder as ‘Smoke Test Cases’, ‘Regression Test Cases’, ‘Unit Test Cases’, ‘Integration Test Cases’, ...etc. based on application.



5. Creation of Test Cases under Folder Structure

- Select respective Folder, where we need to add test cases.
- Click '**+ New Test Case**' to create new test cases.
- Fill the following fields:
 - **Name:** Clear and descriptive title.
 - **Objective:** Purpose of the test.
 - **Preconditions:** Setup required before execution.
 - **Priority:** Based on business impact.
 - **Components:** Product name.
 - **Folder:** Test Case location (by default take selected folder, Change folder location if required)
 - **Labels:** Select based on application
- Click on '**Create and edit**' button to create test case and view it to add more details in test case.

- Select '**Add Steps Manually**' under '**Test Script**' tab to add Test steps.



- Fill the following fields under '**Test Scripts**' tab to add Test steps:
 - Step:** To add Step-by-step instructions.
 - Test Data:** If any specific data for step, needs to be added.
 - Expected Result:** To add Outcome for each step.
- Click on '**+ Add Step**' button below the step or '+' icon from right top to add more steps below it.

STEP	TEST DATA	EXPECTED RESULT
1	• Enter Teamcenter Active Workplace weblink in any browser https://teamcenter-qa.konecranes.com/iawc/	Should display login page

5.1 Add Test Cases/ Steps Features

- Add step:** Use it to add new step below.
- Call to test:** Use it to call existing test case.
- Clone:** Use it to clone same step below.
- Attach files:** Use it to attach any files for test step.
- Delete:** Use it to delete any Test step.
- Show/Hide steps:** When used 'Call to test' this option available to show/hide test steps.
- Add Parameter:** Use it to check test case with different Parameter.

The top screenshot shows a test step with a 'Show steps' link highlighted. The bottom screenshot shows the 'Parameters' dialog with a 'Data type: Parameters' dropdown highlighted.

Will see some of the above Features in detail:

5.1.1 Add Step

- This is used to add new steps in Test Cases.
- Click on '+ Add step' button or '+' icon from right vertical bar to add step below to existing step.
- e.g.: Step 1, 2, 3, so on

The screenshot shows the 'Test Script' tab selected in the top navigation bar. Below it, the 'Steps (2)' section displays two test steps. Step 1: 'Open any browser and enter Teamcenter Active Workspace weblink.' Step 2: 'On Login page enter 'Username', 'Password' and click on 'Login' button.' Each step has a 'TEST DATA' and 'EXPECTED RESULT' column. A sidebar on the right provides icons for managing steps, including a '+' sign for adding a new step.

5.1.2 Call to Test

- This function is used to **call existing test case in another test case**.
- Suppose we are creating any test case, and we realize that we require some test steps which have already been written in another test case, so we do not require to write same steps again we just use Call to Test option (use Arrow icon from right vertical bar) to include that test case into another Test case, then we can continue to write further steps.
- Once we add existing test case, then for added step showing heading as '**Call to Test**' with '**Show/Hide Steps**' option to expand steps as shown in below screenshot.
- Using '**Call to Test**' we can assemble small pieces of test case to large end to end scenarios.

The screenshot shows the 'Test Script' tab selected. The 'Steps (14)' section contains a step labeled 'CALL TO TEST'. Step 1: 'KCPLM-T2181 (1.0) TC_001 Login to Teamcenter Application'. Step 2: 'Navigate from Getting Started page -> My Teamcenter on bottom left corner of TC.'. The 'Show steps' link next to Step 1 is highlighted with a red box. A sidebar on the right includes a red box around the '+' sign for adding a new step.

5.1.3 Hide/Show Steps

- This option will display once we use **Call to test** function.
- It is used to show or hide the called test case.

This screenshot is identical to the one above, showing the 'Test Script' tab and the 'Steps (14)' section with the 'CALL TO TEST' step and its details. The 'Show steps' link is again highlighted with a red box.

5.1.4 Add Parameter

- Can add Multiple Parameter using Add Parameter as Parameter name, Default value pair and use Parameter name in {} in Steps or Test Data.
- **e.g.:** Enter app url as {Url}
- On entering open curly bracket ({} shows popup list with all Parameter available.
- Can edit default values, which can be editing different situations.
- It is used to check test cases with different data inputs

Follow step by step instructions below to Add Parameter for any Step.

1. Under Test Script, select Data type as '**Parameter**' to add data as parameter for reusability.

The screenshot shows the 'Test Script' tab selected in the top navigation bar. Below it, a table lists a single step. The 'Data type' dropdown is currently set to 'None'. A tooltip for 'Test data' is displayed, stating: 'Can be stored in one or more central data sources and reused across different test cases'.

2. Enter '**Parameter name**' and '**Default value**' as pair

The screenshot shows the 'Parameters' section. A new parameter named 'Url' has been added, with a default value of 'https://teamcenter-qa.konecranes.com/awc/'. The 'Data type' is set to 'Parameters'.

3. Refer below added parameter display in top Parameter section and in step showing suggestion as 'use {} to insert a parameter'.

The screenshot shows the 'Test Script' interface again. The 'Parameters' section displays the 'Url' parameter with its default value. In the 'Steps (1)' section, the first step's 'TEST DATA' field contains the placeholder 'Click to type • Use {} to insert a parameter'.

4. On entering open curly bracket ({} shows popup list with all Parameter available.

The screenshot shows the 'Test Script' interface once more. The 'Parameters' section is visible. In the 'Steps (1)' section, the 'TEST DATA' field for the first step shows the parameter name 'URL' followed by an open curly brace {}, suggesting the user can use it as a parameter.

5. Select from existing Parameter. Refer below added Parameter in Test Data.
6. Can add Parameter in ‘Step’ or ‘Test Data’ and can change ‘Default value’ of Parameter in top Parameter section to execute step with different parameters.

The screenshot shows the Zephyr Test Case editor interface. At the top, there are tabs for Details, Test Script, Traceability, Execution, and History. The Type is set to Step by Step. In the Parameters section, a parameter named 'URL' is defined with a default value of 'https://teamcenter-qa.konecranes.com/awc/'. Below this, the Steps section contains one step labeled '1'. The step details are: STEP 1 - Enter Teamcenter Active Workplace weblink in any browser. The TEST DATA column for this step also contains the 'URL' placeholder, which is highlighted with a red box. The EXPECTED RESULT is 'Should display/login page'. At the bottom, there are buttons for '+ Add step' and 'Record Steps'.

5.2 Version Control to Update Test Cases

- If we have test case as per user story and after some time there is change in requirement in that case, we have to modify test case as per change, but we have to keep track of older test case as well, so in Zephyr, we can do it by using Version Control.
- In test case screen, top right there is ‘+’ button next to ‘Version 1.0’ dropdown to create new version.
- Next to Version, we can see current version of open test case.

The screenshot shows the Zephyr Test Case editor interface. The 'Version 1.0' dropdown menu is open, revealing a red box around the '+' button used to create a new test case version. The dropdown also shows 'Version 1.0' and 'Version 2.0'.

- On Click of ' +' (New Test Case Version) button, it will open confirm popup to create new version of test case.

The screenshot shows a confirmation dialog box titled 'New Test Case Version' with the message 'Are you sure you want to create new version?'. It has 'Cancel' and 'Confirm' buttons. The background shows the Zephyr interface with the 'Version 1.0' dropdown open.

- On click of 'Confirm' button it will create new test case version.
- Do the modification in test case as per need and save the Test case, so it will save the changes in new version.

The screenshot shows the Zephyr Test Case editor interface. The 'Version 2.0' dropdown menu is open, showing the newly created version alongside 'Version 1.0'.

- On click of Version dropdown it will display list of available versions of test case.

The screenshot shows the Zephyr Test Case editor interface. The 'Version 2.0' dropdown menu is open, displaying the available versions: 2.0 and 1.0.

- Select the proper version of test case, while Test Case execution.

5.3 BDD Test Cases

Note: BDD test cases, we are not using now, this is just for information purpose.

- Test case written in **Behaviour Driven Development (BDD)** format are easily understood by both technical and non-technical stake holders.
- Test Case in BDD follow Given - When - Then structure.
- Given describe the Initial context or Precondition necessary for test.
- When specifies the Action taken by user or Event that triggers the behaviour been tested.
- Then describe the Expected Outcome or result of test case.
- Under '**Test Script**' tab, select type as '**BDD-Gherkin**' to write BDD test cases.

Type: BDD - Gherkin

BDD - Gherkin Script

```
1 Given I open the application url
2 And I click on "Login" button
3 Then I input user name "ahmed"
4 And I input password "password"
5 And I click "submit" button
6 Then I should see home page.
```

- Can export BDD test cases by selecting test case and click on 'More' option and click on 'Export feature files (BDD-Gherkin)' to export test cases.

5.4 Data Driven to Parameterize Test Cases

- Data driven test case is design to **run test case with multiple sets of data**.
- This approach is used test same functionality/ scenario with different inputs to validate behaviour of application under various condition.
- We can create Data driven test cases without writing duplicate test case for different sets of data.
- To add Data driven testcase, select option as '**Test Data**' under Test Script and add multiple columns depending on input fields for which need to execute test case and add different sets of data under each column to execute test case for all sets of data then use column name in test step like Parameter.
- Under '**Test Script**', select Data type as '**Test Data**' to add multiple test data.

Data type : None

None

Parameters
Create parameters to allow the reusability of the test cases.

Test data
Can be stored in one or more external data sources and shared/reused across different test cases

- On selecting Data type as '**Test Data**', it will display option to Add Columns for multiple test data.

- Add required column to check with multiple set of data.
- Click on '**Add row**' button to add multiple sets of test data.

- Enter '{' to see list of available Test data column in '**Step**' or '**Test Data**' sections to check Test Case with multiple sets of data.

Link to Requirements

- Use the Traceability tab to link test cases to Jira stories or requirements.

6. Test Cycle/ Test Execution

- Folder Structure in Test Cycle
- Create a Test Cycle
- Add Test Cases to Test Cycle
- Clone Test Cycle

6.1 Folder Structure in Test Cycle

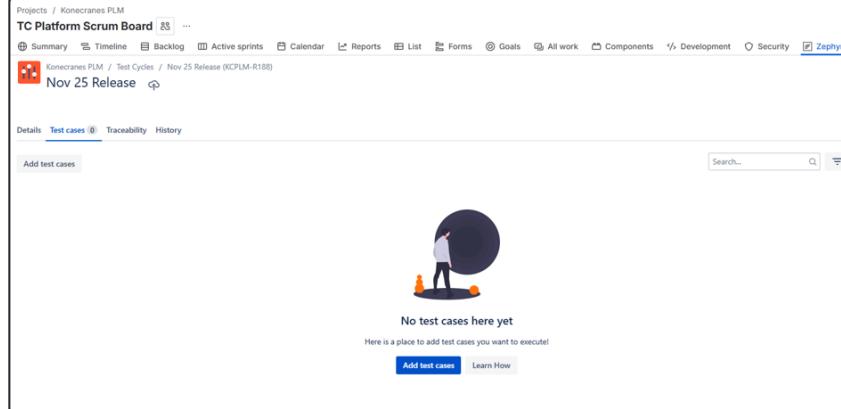
- Under '**Test Cycles**' tab, click on '**+ New Folder**' option to create new folder for Test Cycle same as we are creating folders in Test Cases.
- Add folders based on application. (e.g.: Teamcenter Folder)
- Add subfolder under application folder by clicking '...' icon next to created folder.
- Add subfolder based on Releases, Sprints, module wise, feature wise, ...etc. as per application.

- Note: Follow [Test Case Management | 4. Folder Structure for Test Cases](#) link/section for Folder Structure creation in detail.

6.2 Create a Test Cycle

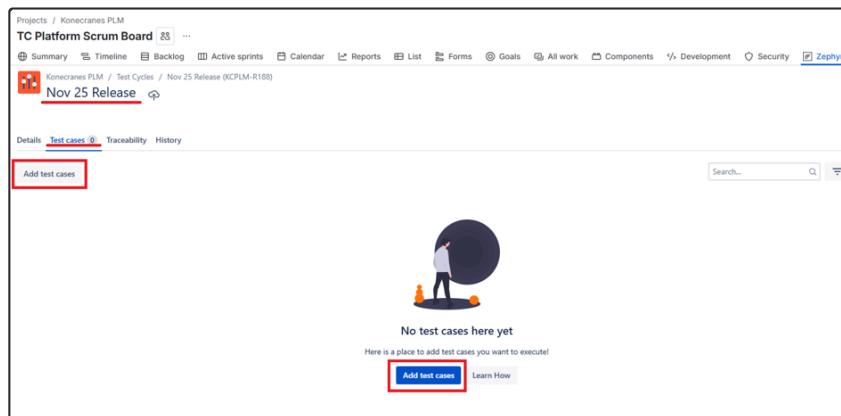
- Under ‘Test Cycle’ tab, select a Folder/Subfolder for which we have to create Test Cycle.
- Click on ‘+New Test Cycle’ button from right section.
- Add below metadata to create new Test Cycle:
 - **Name** - Test Cycle name
 - **Description** - More Info
 - **Status** - Initial/ By default Status will be Not Executed
 - **Release version** - Select from dropdown
 - **Iteration** - Select from dropdown
 - **Owner** - Select Owner from list
 - **Planned start/end date** - Select date
 - **Folder Location** - Select correct folder location

- Once enter all information and click on 'Create and edit' button to create new Test Cycle.
- On click of 'Create and edit' button, it will navigate to new Test Cycle page on ‘Test Cases’ tab to add Test Cases into Test Cycle.

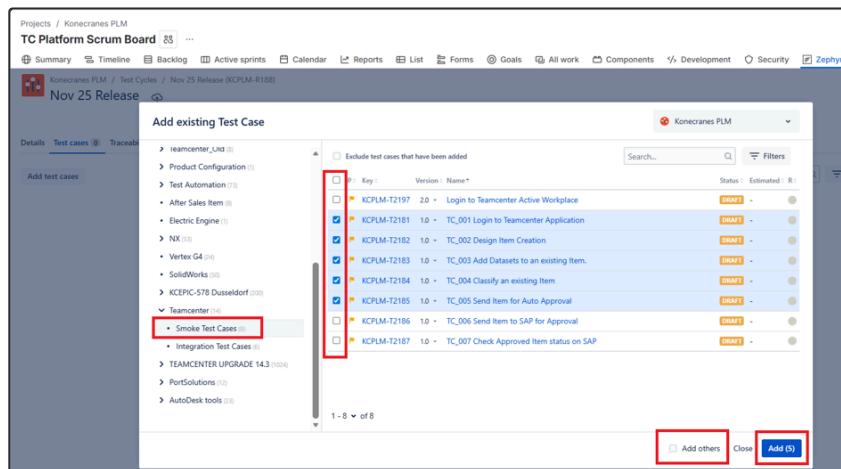


6.3 Add Test Cases to Test Cycle

- Click on any of 'Add test cases' button to add test cases in Test Cycle.



- On clicking 'Add test cases' button, it will open a popup window as below to select test cases.
- Select respective Folder from left navigation, from which we need to select test cases.
- Once select specific folder, it will display all Test cases specific to that folder on right section.
- Select checkbox to add relevant test cases into it.
- Can click on 'Add others' checkbox from bottom to select test cases from another folder for same Test Cycle.



- Once select test cases for specific folder, click on 'Add' button to add it in test cycle.
- Can select other Folder to add Test cases at a same time to same cycle if 'Add others' checkbox selected.

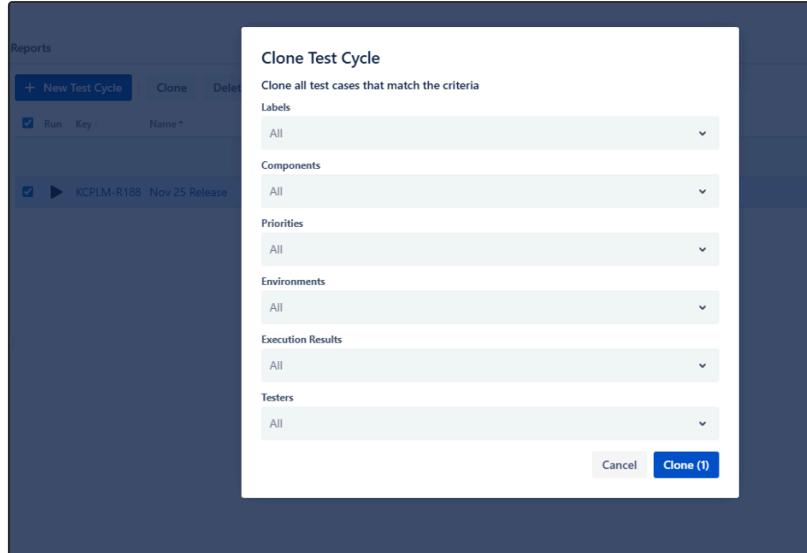
- Click on '**Add test cases**' button from top to add more test cases in same test cycle.
- All selected Test Cases will be displayed under '**Test Cases**' tab for Test Cycle'.
- Click on '**Test Cycle**' from top breadcrumb to see created Test Cycle.
- Once test cycle is created, can see Test Cycle for Specific Folder/Feature and can see initial progress as **0%** in main list with status as **Not Executed**.

- Click on '**All test cycles**' from left navigation to see all available Test Cycles on right section with its 'Progress' and 'Status'.
- On left navigation count showing next to each folder says that the number of Test Cycles created/available for specific folder.

6.4 Clone Test Cycle

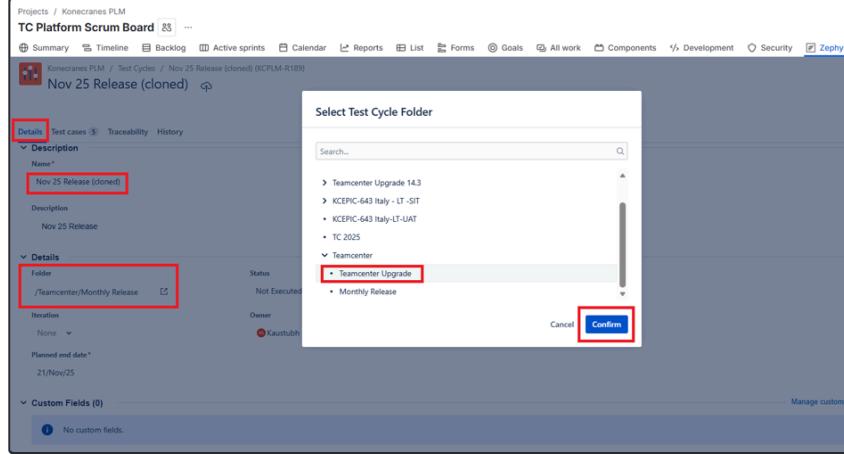
- Can create multiple Test Cycle under Folder, also can Clone Test Cycle to use same Test cases in another Test Cycle.

- Select Test Cycle which needs to be clone and click on '**Clone**' button from top.

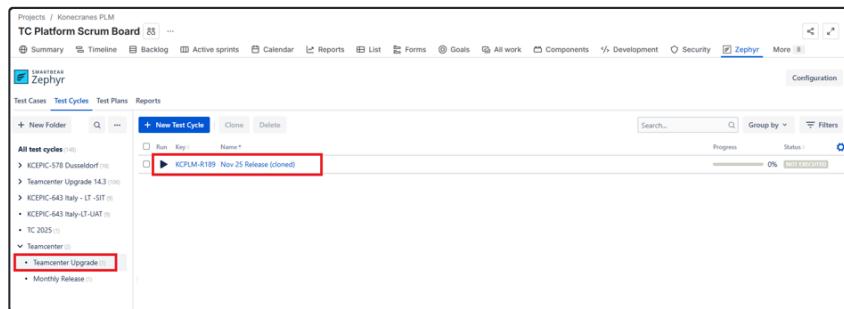


- On click of 'Clone', Test Cycle will be clone and it will display in list as below.

- Click on clone test cycle or any Test Cycle, so it will open in Edit mode.
- It will display 'Details' tab. Here we can update Test Cycle Name, Location, Release version, Date, etc... so same Test Cycle can reuse.

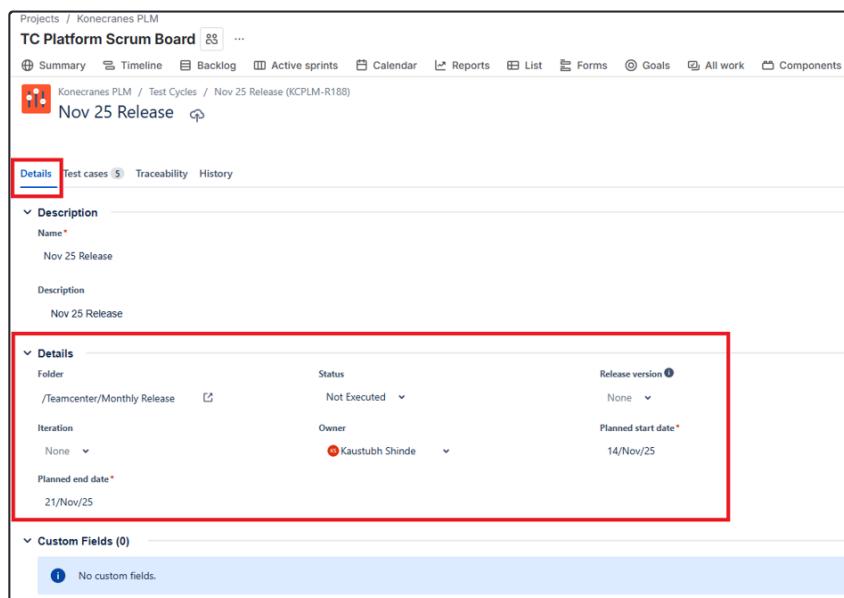


- On update Test cycle location, it will move to updated location folder, so accordingly we can reuse test cycle for different Release/ updates.



6.5 Run Test Cycle / Execute Test Cases

- To Run/Execute any Test Cycle, click on same test cycle in edit mode.
- On 'Details' tab, update Owner, Planned Start/End Date, Release Version, Iteration.



- Navigate to 'Test Cases' tab and Assign Environment, Tester, Release version, etc...
 - Environment** - Select it basis on which environment test case needs to run.
 - Tester** - Select User/assignee who will execute the test cases.

- To run/execute test cycle, navigate to '**Test Cycle**' tab and select specific folder to run Test Cycle of that folder.
- Click on '**Play icon**' for Test Cycle to begin execution.

- On click of '**Play icon**' for Test Cycle, test case execution page will open as below.
- By default, 1st Test Case is selected.
- From left navigation, select '**Show assigned to me**' from top, to display assigned test cases to you.
- By default, Test Case status will display as 'Not Executed' (Can see at right top).
- Click on **Start Timer** icon from right top to see total execution time require to run test case, Timer will automatically stop when all steps are Pass or any step fails or any step blocked.

- Execute each step and update Test step status.
- Here in all test step on right there are options to select result as below:
 - Pass
 - Fail
 - Blocked
 - Skipped
 - In Progress
- Click any of this right option/icon to update test step status.

The screenshot shows a 'Test Script' section within a Jira Test Management board. It contains two steps:

- Step 1:** STEP: Double click on TC application. TEST DATA: [empty]. EXPECTED RESULT: Should open Teamcenter login popup. Status: 13/Nov/25 6:59 PM PASS.
- Step 2:** STEP: Enter valid username and password and click on 'OK' button to login. TEST DATA: Username: [empty], Password: [empty]. EXPECTED RESULT: User should be able to successfully log in. Status: 13/Nov/25 6:59 PM FAIL.

On the right, there is a legend for status colors: Not Executed (grey), In Progress (orange), Pass (green), Fail (red), Blocked (blue), and Skipped (dark blue).

- If all test steps are Passed, then Test case status will change automatically as Pass.
- If any test step is Fail the Test Case status will display as Fail.
- For Pass step display in green vertical bar and for fail test step showing red vertical bar at left.
- If any test step failed, then can 'Create new issue' or 'Add existing issue' as shown below.

The screenshot shows a 'Test Cases' section within a Jira Test Management board. It contains four steps:

- Step 1:** Click on Home and Navigate to Menu then click on 'File -> New' and click on 'Dataset'. Verify 'New Dataset' popup window is opened. ACTUAL RESULT: Click to type the actual result.
- Step 2:** STEP: Select File Type which want to upload on New Dataset popup and click on '...' next to 'Import to upload dataset'. TEST DATA: [image file]. EXPECTED RESULT: Selected file type should be highlighted. Import popup should display document file type only. Status: 13/Nov/25 7:09 PM FAIL.
- Step 3:** STEP: Upload file of selected file type and Verify name of dataset. TEST DATA: None. EXPECTED RESULT: Should display file name as ID with revision suffix (70414400-1). Status: 13/Nov/25 7:09 PM FAIL.
- Step 4:** Click to type the actual result.

On the right, there is a legend for status colors: Not Executed (grey), In Progress (orange), Pass (green), Fail (red), Blocked (blue), and Skipped (dark blue). A red box highlights the 'Create new issue' button.

- Once all Test cases executed then we can see number of test cases Pass, Fail, Blocked, Not Executed in progress bar on mouse hover on it.

The screenshot shows the 'Test Cycles' tab within a Jira Test Management board. It lists several test cycles:

- All test cycles: KCEPIC-570 Dusseldorf, Teamcenter Upgrade 14.3, KCEPIC-643 Italy - LT-SIT, KCEPIC-643 Italy-LT-UAT, TC 2025, Teamcenter, Teamcenter Upgrade, Monthly Release.
- Selected cycle: KCPUM-R188 Nov 25 Release.

A progress bar at the bottom indicates the cycle is 60% complete. A red box highlights the selected cycle.

7. Reporting and Metrics

- Use Test Execution Reports to track progress.
- Monitor:
- Pass/Fail rates
- Defect density
- Coverage by requirement

8. Review and update test cases regularly.

- Maintain traceability with requirements and defects.