I'm not a robot

Requirments qtest test case test scenario example upload media

qTest Manager provides a feature to store Requirements, allowing testers to easily access information for test coverage. Users can either create new Requirements directly within qTest Manager or import them from external Application Lifecycle Management (ALM) tools like Jira or VersionOne. Additionally, Microsoft Excel can be used to make changes to existing Requirements and import new ones linked to Test Cases. If you have already created Requirements in an external ALM tool, you can import them into qTest Manager for further management. However, separate management of Requirements not integrated with the external tool is still possible. For more information on importing Requirements from an external ALM tool, refer to Import new Requirements using Excel. Importing Requirements using Microsoft Excel allows users to create multiple Requirements at once and then map the fields properly in qTest Manager. The sample import template is recommended for this process. When importing Requirements using Excel, note that these Requirements will not be integrated with an external ALM tool. Changes made on-site won't affect project-level custom fields, so those can't be tracked with qTest Insights. Site-level field updates are explained in Site Fields. For combo boxes or check boxes that accept multiple values, separate them with commas, not spaces. Think of formatting like this: 1,2,3,4 instead of 1, 2, 3, 4. If needed, add custom fields for Requirements in qTest Manager and map them so they can be used. Add Custom Fields in qTest Manager has a procedure on how to do that. Before importing an Excel file into qTest Manager, decide if you want to import Requirements into an existing folder or a new one. Then prepare the Excel sheet name according to the format MD-# followed by the folder name. If you're importing into an existing folder, update the sheet name to match the folder's name, like MD-2 Homepage. The sheet name can't be longer than 31 characters. To import Requirements into a new folder, simply rename the sheet tab with the new folder name. qTest will create a new folder based on this name. If you don't rename the sheet tab, the new folder will have a default name that can be changed later. To import an Excel file: Click the Requirements tab and select your project in the tree panel. Then click the Import Excel icon in the tree toolbar to start the import process. The Import Requirement Wizard will guide you through this step. Drag and drop your file into the blue upload box or click the link to pick a file from your computer. Once uploaded, map the fields from the Excel spreadsheet to qTest fields by dragging them over. Don't map the Requirement Id field – qTest Manager will auto-generate it. You don't need to match the Excel column name with the qTest field name. Click Next and then Import. A progress bar will show you how the import is going. When it's finished, a message will appear confirming that the Requirements have been imported. Click OK & Refresh to see them on the Requirements tab. To make changes to existing Requirements using Excel: Download the Requirement Details report and update the details as needed in the Excel file. Then import this updated file into qTest Manager. To import Requirements from an Excel spreadsheet into qTest, you need to accurately map existing Requirement IDs. Moreover, rename the Excel sheet tab to match the current folder name using the MD-# format followed by the folder name. For instance, for importing Requirements into the MD-2 Homepage folder, retitle the Excel sheet tab as MD-2 Homepage; however, ensure that it does not exceed 31 characters in length. qTest is a widely used test management tool utilized by QA professionals to streamline their testing processes. It offers a centralized platform for creating, managing, and executing test cases, tracking defects, and generating reports. This comprehensive guide will walk you through the process of creating and managing test cases in qTest, providing detailed explanations and examples to help maximize its capabilities. What constitutes a Test Case? Before delving into qTest specifics, let's define what a test case entails. A test case is a set of conditions or actions executed on a software application to verify its functionality and ensure it meets specified Requirements. It generally includes: - Test Case ID: A unique identifier for the test case. - Test Case Name: A descriptive name that summarizes the objective of the test case. - Description: A detailed explanation of the test case's purpose and scope. - Preconditions: Conditions that must be met before executing the test case. - Test Steps: A sequence of actions to be performed during the test. - Expected Results: The anticipated outcome of each test step. - Actual Results: The observed outcome of each test step during execution. - Status: The current state of the test case (e.g., Passed, Failed, Blocked). Why Choose qTest for Test Management? qTest provides a robust and user-friendly platform for managing your testing efforts. Some key benefits include: - Centralized Test Repository: Store all your test cases in a single location, making them easily accessible to the entire team. - Improved Collaboration: Facilitate collaboration among testers, developers, and stakeholders by providing a shared platform for communication and feedback. - Increased Efficiency: Streamline your testing process with features like test case reusability, test execution scheduling, and automated reporting. - Enhanced Traceability: Link test cases to Requirements and defects to gain better visibility into the testing progress and identify areas of improvement. - Data-Driven Insights: Generate comprehensive reports and dashboards to track key metrics and make informed decisions about your testing strategy. Creating Test Cases in qTest Now, let's explore how to create test cases in qTest. 1. Accessing the Test Design Module To begin, log in to your qTest instance and navigate to the "Test Design" module. This is where you'll create and manage your test cases. 2. Creating a Test Case Once you're in the Test Design module, follow these steps to create a new test case: - Select a Module: Test cases in qTest are organized into modules, which are essentially folders that help you categorize and manage your test cases. - Click "New Test Case": Click the "New Test Case" button to open the test case creation form. - Fill in the Test Case Details: Provide the necessary information for your test case, including the test case name, description, preconditions, test steps, and expected results. You can also assign the test case to a specific module during this process. By following these guidelines, you'll be able to efficiently create and manage test cases in qTest, effectively utilizing its features to enhance your testing processes. To create a test case in qTest, you need to fill out its details including setting priority and status, adding attachments like screenshots or documents, and saving it. For instance, let's say you're testing a login feature for a web app. You could make a test case with these specs: the name 'Verify successful login with valid credentials,' a description explaining what it does, preconditions like having a registered account, steps to open the login page, enter username and password, and click log in, and expected results like being redirected to the home page. To ensure you're testing the right things, link your test cases to requirements in qTest by opening the test case, going to the requirements tab, adding linked requirements, and saving it. To manage your test cases effectively, use qTest's features like organizing them with modules and folders, editing existing ones, cloning similar ones, tracking versions, filtering and searching based on criteria like status or assigned tester, importing and exporting in formats like Excel or CSV. Also, make sure to follow a consistent naming convention for your test cases and write clear and concise steps. This will help you keep track of which requirements have been tested and identify any gaps in your test coverage. 1. Define Specific Expected Results and Establish Traceability Clearly outline the expected outcome of each test step to eliminate confusion and ensure accurate testing execution. 2. Organize Test Cases for Easy Navigation Utilize modules and folders to structure your test cases and maintain an easily accessible test repository. 3. Regularly Review and Update Test Cases Periodically review and update test cases to ensure they remain relevant and accurate. 4. Leverage qTest's Features for Streamlined Management Take advantage of qTest's features, including cloning, versioning, filtering, and searching, to optimize your test management process. 5. Link Requirements to Test Cases with Traceability Establish a comprehensive connection between requirements and test cases using qTest Manager, providing essential traceability for QA teams. To link a test case to multiple requirements, press SHIFT and/or CTRL keys to select multiple requirements in the dialog. To enable Automated Code Coverage, the Feature File needs modification. The Test Engineer creates an automated code stub that identifies failing step definitions in the Git Repository. A Trigger Jenkins Pulse Rule is triggered, which informs the Developer by initiating a Jenkins Build and sending results to Jira. A Color Code Scenario Results in Jira Pulse Rule provides visual indicators for failed scenario results. This allows stakeholders (PO/BA, Test Engineer, Developer) to view latest execution outcomes from Jira. Next, automated step definitions using Selenium with Cucumber are implemented to move the feature into production. Refactoring code is necessary to make features and scenarios pass. Committing and pushing the code initiates a Build in Jenkins, validating test scenario success. If successful, a TriggerJenkins rule will notify again. Implement Feature and Add Step Definitions are required actions. Moreover, the Automation Code must be included within each Step Definition Block, supported by a Selenium library for browser UI automation. A developer can view versioned tests in the Git Repository. Validating results in Jenkins and turning Scenario results green (color coding Pulse rule) is necessary to move the Jira issue into a Done status. Additionally, the Link and Report Scenario Results in qTest Manager Pulse Rule allows linking Scenario requirements to qTest Manager for reporting and viewing test progress in BDD along with other projects. Navigation to Test Execution tab, selecting the Test Cycle or Suite, clicking a Test Run ID link, and accessing the Execution History tab enable viewing latest test scenario results. Furthermore, Create Additional Charts with qTest Insights View Execution Results facilitate navigating to qTest Manager from Product Navigator, selecting Quality Analysis, filtering by Project name, refreshing for results in Quality dashboard. Saving or Embedding the Script allows sharing reports on another website. Similarly, View Coverage Results involve navigating to qTest Insights, selecting Coverage Analysis, filtering by Project name, and refreshing for results in Coverage dashboard. Finally, qTest Manager provides essential traceability between requirements, test cases, and defects for efficient project management. Establishing Traceability for Requirements and Test Cases To create a traceability matrix, you can link requirements or test cases. Tricentis recommends linking requirements to test cases via API or within the UI. Maximum thresholds: - One requirement linked to 2,500 test cases: expect difficult navigation within the UI and increased load time. - One requirement linked to 60,001 or more test cases: not recommended as the system will become unstable. Best Practice is to link up to 100 test cases to a requirement in a single operation. Alternatively, administrators can set a new limit in Administration > Settings. Linking Requirements to Existing Test Cases: 1. Select Requirements in qTest Manager. 2. Expand your module and select the requirement. 3. In the Resources section, select Add. 4. The Test Cases dialog displays; select the test case to link to the requirement. 5. Click Add, then Save. A requirement can be linked to more than one test case. To quickly link multiple test cases, use SHIFT and/or CTRL key selection in the dialog. Linking Requirements to New Test Cases: 1. On the Requirement page, expand the Create Associated Test Cases section. 2. Enter a name, type, description, and precondition. 3. Click Create; the created test case is automatically linked to the requirement and displays in the Linked Test Cases section. Linking Test Cases to Requirements: 1. Select the applicable Test Case in the Test Design tab's left navigation panel. 2. On the Requirements tab of the Test Case page, click Add. 3. The Add Requirements dialog displays; select the proper Requirement in the dialog. 4. Click Add; a test case can be linked to more than one requirement. To quickly link multiple requirements, press SHIFT and/or CTRL key selection in the dialog.