

## QUALITY CENTER (BASIC)

### Introduction

QC (Quality Center) is a web based Test Management Tool. It is based on Client Server technology and has four main modules/tabs:

Requirements, Test Plan, Test Lab and Defects for management of testing processes.

There can be additional modules as well depending on the various Add-ins installed like BPT (Business Process Testing) etc.

**Releases:** This module helps to manage releases and cycles (iterations) more effectively. User can easily plan and track application release progress using this module.

**Requirements:** This module is used for Requirement Management and Requirements Traceability of various Test cases stored in the QC Repository.

**Test Plan:** This tab is used for creating or updating different Test Cases. The Test Cases are contained in different folders which are displayed in a tree like structure. It can store both Manual as well as Automated test cases. Manual Test Cases can be written locally or imported from Excel Sheets.

With each 'Test Step' having Expected Result and Actual Result section. QC supports automated script developed for different Automation Tools like QTP, LoadRunner, WinRunner etc. These scripts can be saved directly from the Tool into the Test Plan tab of QC. However, prior to this, appropriate QC Add-in needs to be installed to support an Automation Tool.

**Test Lab:** This tab is for execution of the test cases stored in the Test Plan module which can be imported locally to the Test Lab screen and Run. When Manual Test case is executed, it opens up a pop up listing all the Test Steps and the user is supposed to update status of each step with Passed, Failed or Not Complete. When automated test case is run, QC invokes the Automation Tool which in turn executes the script and stores back the result into QC repository and displays on the UI.

**Defects:** All the defects are logged in this tab of Quality Center. These defects can be mapped to the corresponding test cases which failed and hence to the Requirements tab.

### Course Contents

<b>Working With Releases</b>	<b>Test Execution</b>
<ul style="list-style-type: none"> <li>Creating a release tree.</li> <li>Tying requirements to releases and cycles.</li> <li>Tying tests to releases and cycles.</li> </ul>	<ul style="list-style-type: none"> <li>Building a test sets tree.</li> <li>Creating test sets.</li> <li>Organizing tests in a test set.</li> <li>Defining and scheduling test execution flows.</li> <li>Configure automated test rerun and cleanup rules.</li> <li>Executing manual and automated tests.</li> <li>Recording and reviewing test execution results.</li> <li>Monitoring the status of test sets.</li> </ul>
<b>Defining Test Requirements</b>	<b>Defect Tracking</b>
<ul style="list-style-type: none"> <li>Creating and defining requirements.</li> <li>Building a requirements tree.</li> </ul>	<ul style="list-style-type: none"> <li>Logging defects.</li> <li>Searching and reviewing defects.</li> <li>Associating defects to other entities.</li> <li>Updating a Defect.</li> <li>Tracking the status of defects.</li> </ul>
<b>Analyze Requirement Risks</b>	<b>Reporting &amp; Analysis</b>
<ul style="list-style-type: none"> <li>Add traceability links between requirements.</li> <li>Perform risk analysis for requirements.</li> </ul>	<ul style="list-style-type: none"> <li>Generate analysis reports and graphs.</li> <li>Generate Microsoft Excel reports.</li> </ul>
<b>Test Planning</b>	
<ul style="list-style-type: none"> <li>Building a test plan tree.</li> <li>Creating tests.</li> <li>Linking tests and requirements.</li> <li>Designing test steps.</li> <li>Using parameters in tests.</li> <li>Configuring a test to call other tests.</li> <li>Generating test scripts.</li> <li>Monitoring the status of test plans.</li> </ul>	

Lab Exercise(s)/ Practice sessions