

GOOD MORNING

"The more that you read/do, the more things you will know. The more that you learn the more places you'll go."

Welcome to My Presentation

Pushpala Monika

Certified Tosca Professional

TOSCA

DAY-1:

Following are some positive and negative points of Tosca:

Positive:

- **Test management and Automation Tool in one**
- **No Scripting**
- **Testing methodology**
- **Supports the most common technologies (HTML, Java, SOA, SAP, Oracle, ...)**
- **Good vendor support**

Negative:

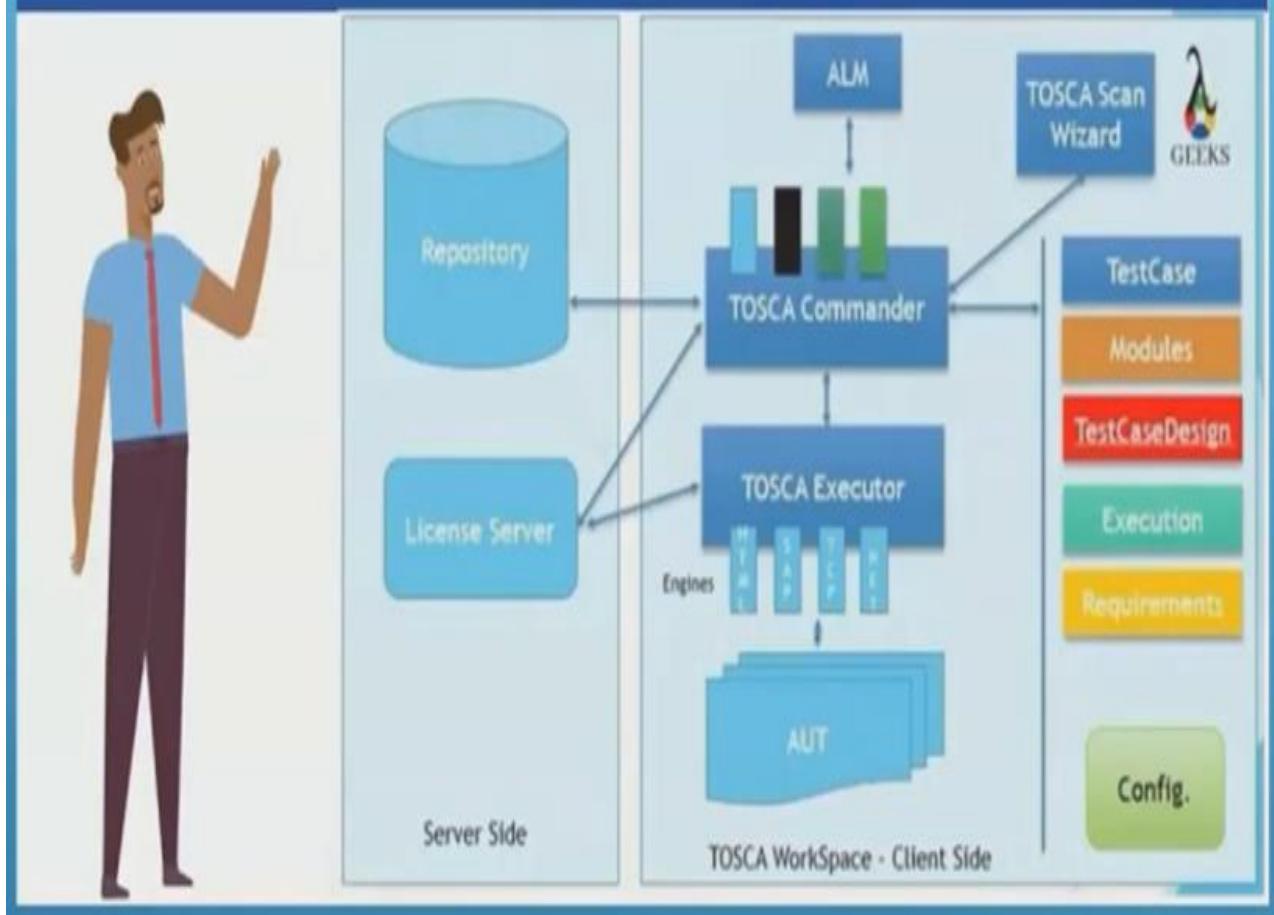
- **Not easy to learn and setup efficiently, especially in a large organization**
- **Coding (when necessary) can be complex**
- **Pricing**

It is a commercial tool, so it should be compared to other commercial tools like HP-Suite. If a freeware tool fits your needs its ok, no need to spend money. TOSCA provides action modes for this (Input, Verify, Buffer), which can be selected from a list for the control. Non-technical testers will prefer this approach to scripting.

What I really appreciate with Tricentis is that they not only provide a tool, but also propose a sophisticated testing strategy. The TCD Design (Test Case Design) module can guide you to the minimum number of necessary test cases giving a high coverage. It's quite easy to review the test design of different testers and to understand their ideas.

Tosca Architecture:

Architecture of Tosca



[ALM-Application lifecycle Management](#)

The key components on client side are:



SUMMARY

TOSCA SERVER-SIDE REPOSITORY: CENTRALIZED STORAGE FOR TEST ASSETS.

TOSCA COMMANDER: THE PRIMARY INTERFACE FOR CREATING, MANAGING, AND EXECUTING TESTS.

TOSCA EXECUTOR: RUNS THE TEST CASES AND STORES RESULTS.

TOSCA SCAN WIZARD: USED TO CREATE MODULES BY SCANNING APPLICATION UIS OR APIs.

MODULES, TEST CASES, TEST CASE DESIGN, EXECUTION, AND REQUIREMENTS ARE ALL SECTIONS WITHIN TOSCA COMMANDER THAT PROVIDE A COMPREHENSIVE FRAMEWORK FOR ORGANIZING, EXECUTING, AND ANALYZING AUTOMATED TESTS.

1) First link--Tricentis Tosca Fundamentals – Automating web application testing (AS1):--

In order to complete this training, you will need to have **Tricentis Tosca installed on your machine, as well as a valid license**. A training license to complete the courses in the Tricentis Academy is available through our Support Hub:https://support-hub.tricentis.com/open?id=training_license_request

Before getting started, please read the information below:

- » The training is made up of 39 lessons: The first 27 lessons cover the fundamentals of Automated Testing while the last 12 lessons cover more advanced material.
- » Each lesson contains **one video lesson**. Some lessons contain **one or more exercises from the Workbook**.
- » Each exercise is followed by an **exercise solution video**.
- » This training also contains extra learning materials, such as Cheat Sheets, Troubleshooting Guide, and infographics among others.
- » We recommend that you watch the video lesson to learn about each topic and then complete the exercise(s) to practice the topic. Finally, watch the solution video to see how we would complete the exercise(s).
- » The training is based upon our sample Web Shop application: <http://demowebshop.tricentis.com>
- » Ensure that you have installed the Tricentis browser extension: [Install browser extension](#)

AUTOMATION SPECIALIST LEVEL 1 ROADMAP

How to get started

Learning Path	What is needed to be successful?	Recommended Browser for Testing
<p>Start Course</p>  <p>Start your learning path with the introductory videos to get you ready for testing automation journey ahead. This course is your first step to becoming specialized in automation with Tricentis Tosca. By the end of the course, you will be able to fully execute automated GUI TestCases that you will create during the lessons.</p>	<p>Install Tosca and Get License</p>  <ul style="list-style-type: none"> Request an 8-week trial license Select Cloud (Recommended) or on-premise Select and submit license request Download and install Tosca on your device 	<p>Google Extension</p> <p>The course has been developed using Google Chrome, so we recommend using this browser while doing the exercises. Make sure to install the extension as instructed in the lessons.</p>
Curriculum	Practice Exercises	Learner Action
<p>From lesson 1-10 you will:</p>  <ul style="list-style-type: none"> Start by watching the videos and get familiar with the interface of Tosca, the System Under Test (SUT), Model-Based Test Automation (MBTA) and XScan You will also get to know how to identify controls by properties in XScan and controls by anchor in XScan 	<p>From exercise 1-4 you will:</p>  <ul style="list-style-type: none"> Create Workspace Navigate the Systems Under Test (SUT) Create first Module Identify by Anchor and create Control Group 	<p>Tosca Commander</p> <ul style="list-style-type: none"> Work on basic subset to get more comfortable with Tosca Commander. Subsets can be downloaded and imported directly to Tosca Do you have a query/doubt? Every section comes with exercises to practice your knowledge. Do you need help with it? Watch the solution video and see how we do it.
<p>From lesson 11-20 you will:</p>  <ul style="list-style-type: none"> Get introduced to TestCases Learn how to populate and modify TestCases using ActionModes and Dynamic Values Learn how to create a Library and a new TestCase using an existing Library and Reusable TestStepBlocks 	<p>From exercise 5-17 you will:</p>  <ul style="list-style-type: none"> Create TestCase Structure and TestSteps Enter TestStepValues Set Predefined Test Configuration Parameters and ActionMode WaitOn Record and use a buffer Use MATH functions, generate Dynamic Text and Dynamic Data Create a Library and a TestCase from a Library 	<p>Hands-on Exercises</p> <p>Try out the exercises in Tosca. Learn to create your own Libraries using TestSteps previously created and run them in the ExecutionList</p>
<p>From lesson 21-27 you will:</p>  <p>Progressing towards the advanced part of the learning path, learn some more advanced Module actions along with self-defined Test Configuration Parameters, Business Parameters, ExecutionList and Requirements</p>	<p>From exercise 18-22 you will:</p>  <ul style="list-style-type: none"> Apply Rescan, Value Range, and Module Merge Set self-defined Test Configuration Parameters Create Business Parameters, ExecutionList and Execute ExecutionList Link TestCases and ExecutionList to Requirements 	<p>Hands-on Exercises</p> <p>You have learned about discrete structures of TestCases. Apply these to optimize your previous TestCases and try to point out key differences in performance</p>

Description

Overview:

This Learning Path marks the start of your journey through test automation using Tricentis Tosca. The scope lies on automating testing for simple HTML applications without any coding involved. From exploring Tricentis Tosca's capabilities to practicing the newly gained skills, this Learning Path walks you step by step through the entire testing process of building automated tests quickly and efficiently.

Learning Objectives:

By the end of this Online Course, you will be able to:

- Capture the technical information from your System under Test needed for test automation.
- Create automated tasks to perform on your system under test.
- Run your first test with Tosca.
- Identify testing criteria for your system under test and reflect it in Tosca.

- Demonstrate how to enhance test data maintenance and reusability for a more efficient test automation.

Prerequisites

knowledge:Basic understanding of software testing concepts

Technical:Access to and valid license for Tricentis Tosca.

product version:This content was developed using Tosca 14.x.

You may notice some minor UI differences between your installed version and the learning materials.

TOSCA INSTALLATION:Please follow the below link to install:

<https://www.tricentis.com/software-testing-tool-trial-demo/tosca-trial>

Then Browser Extension:Use any one preferable Manually--

https://support-hub.tricentis.com/open?id=show_search_result&lang=en&path=%2Ftosca%2F1520%2Fen%2Fcontent%2Fengines_3.0%2Fxbro

wser%2Fxbrowser_scan.htm&product=tosca&sessionRotationTrigger=true&type=product_manual&version=16.0

The screenshot shows a web browser window with the Tricentis logo at the top. The page title is "Create XBrowser Modules". The main content area has a heading "Create XBrowser Modules" and a paragraph explaining that Tosca XScan allows creating XBrowser Modules to test web applications. It then describes how to install the Tricentis Automation Extension for the browser, listing three methods: "During the installation of Tricentis Tosca", "Directly in the browser", and "Manually". The "Manually" option is circled in green. To the right, there is a sidebar titled "In this topic" with links to "Create XBrowser Modules", "Install extension in the browser", "Install extension manually", and "What's next?". At the bottom left, there is a link to "Install extension in the browser".

Install extension in the browser

You can install the Tricentis Automation Extension for the following browsers:

- [Google Chrome](#)

Section 1:

INTRODUCTION:lesson 1



<https://mindmajix.com/tosca-tutorial>

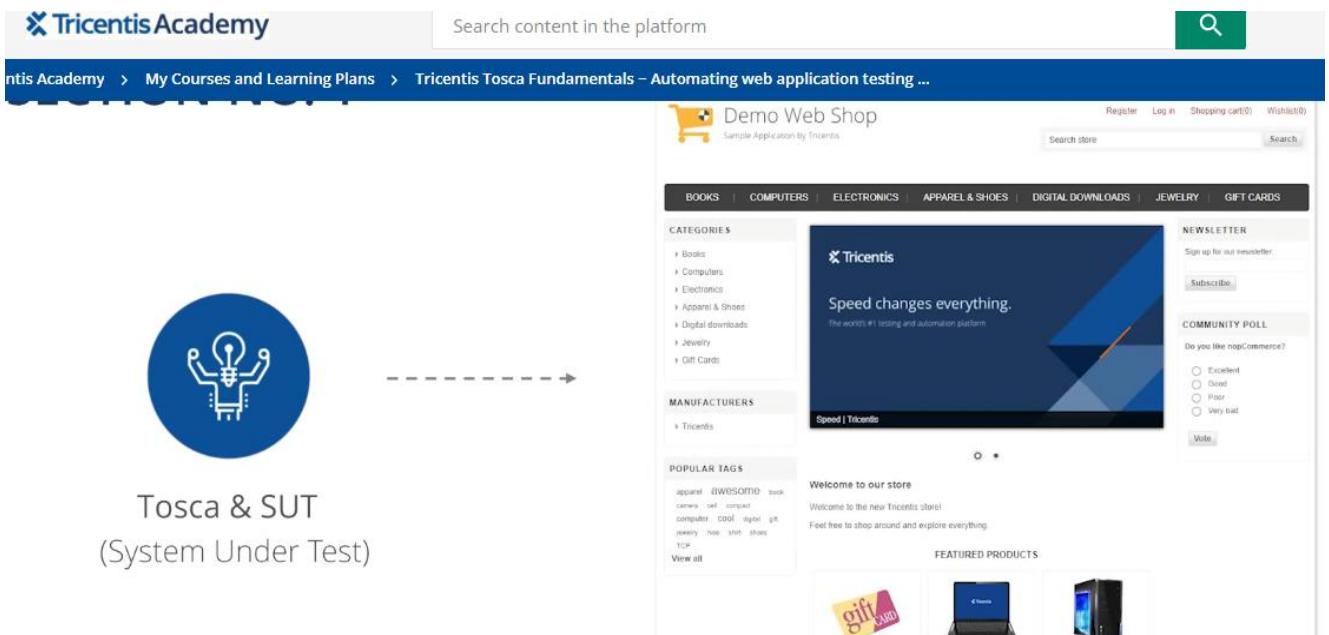
====Contents From this link:

TOSCA Contains 5 Sections:--

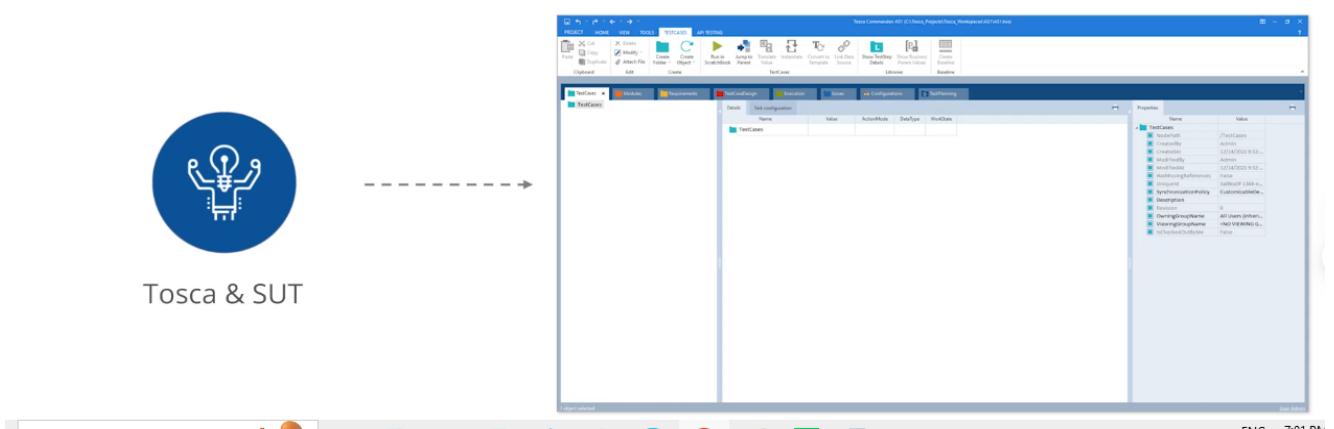


SUT-->System under TEST

Here we are going to see TOSCA &SUT with Demo Web Shop Application:

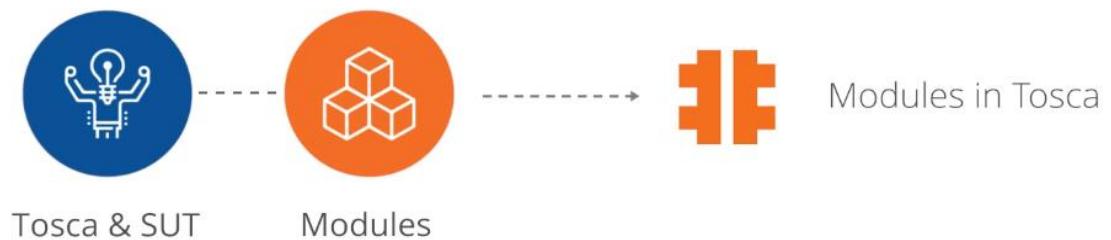


SECTION NO. 1



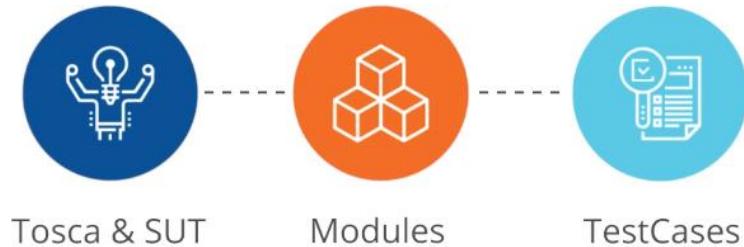
Technical information of SUT by simple scanning it and creating Modules in order to store the Scanned elements in Tosca as it is Code free:

SECTION NO. 2



SECTION NO. 3

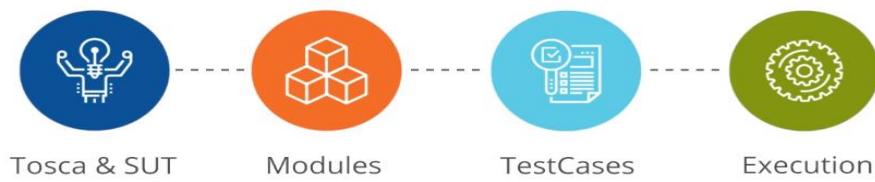
Then we create Test Cases



TestCasescontains:



SECTION NO. 4:

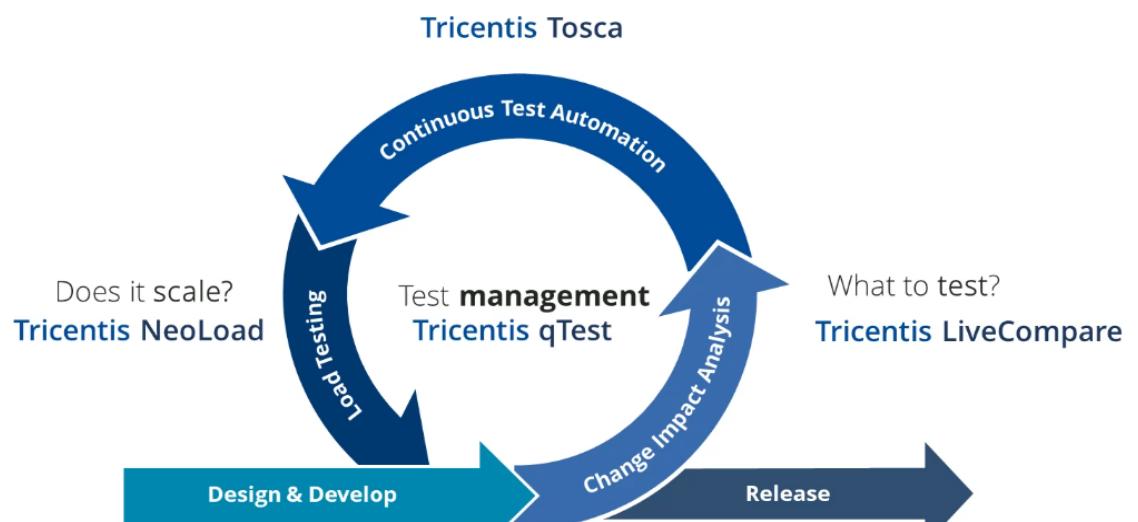


SECTION NO. 5:

We need to link the Test back to Requirements at the end:

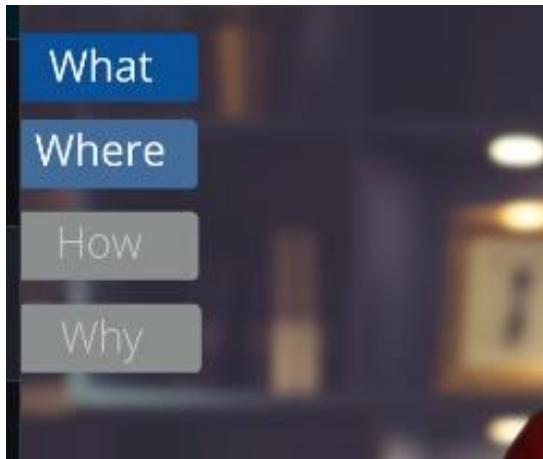


We AIM for :



ABOUT TOSCA :--lesson 2





What, Where, How and Why --Workspace is needed

Create a NEW WorkSPACE:

The screenshot shows the Tricentis Academy homepage. At the top, there is a navigation bar with the Tricentis Academy logo, a search bar, and a trophy icon. Below the navigation bar, a breadcrumb trail shows the user is in 'Tricentis Academy > My Courses and Learning Plans > Tricentis Tosca Fundamentals – Automating web application testing ...'. On the left, there is a sidebar titled 'Recent Projects' with sections for 'First Steps' and 'Advanced Examples'. In the center, there is a large blue banner with the text 'Know of these Tricentis testing capabilities?' and a link 'Discover now >'. At the top right of the page, there are links for 'Online Tutorials', 'Watch Videos', 'Online Help', 'Knowledge Base', and 'Contact Support'. A green circle highlights the 'Create new' button in the top right corner of the main content area.

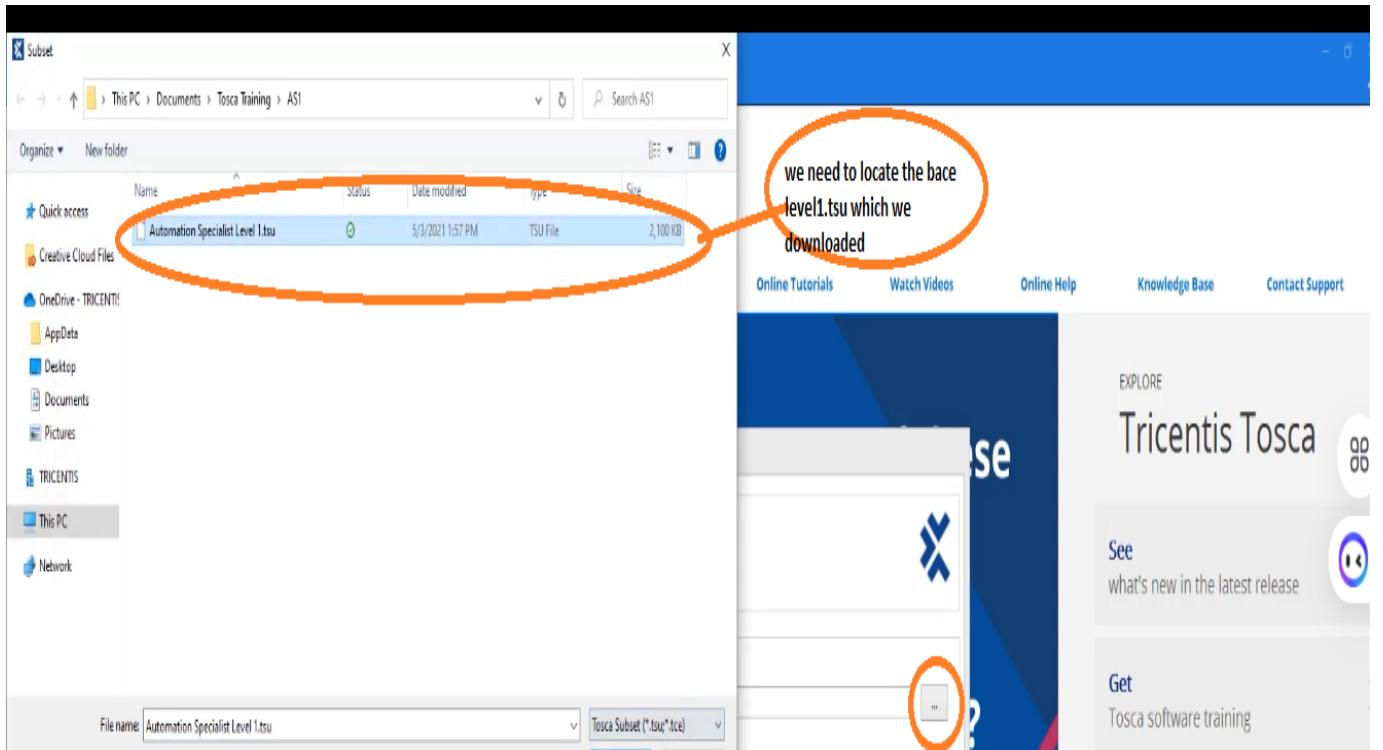
By clicking on Create new we can create work space: Explain each one in detail

The screenshot shows the 'Tosca Commander: Create new workspace' dialog box. The dialog has several fields highlighted with orange circles:

- A dropdown menu labeled 'Select type of Repository:' with the option 'None (creates single user workspace)' selected.
- A field labeled 'Create new workspace in:' containing the path 'C:\Tosca_Projects\Tosca_Workspaces'.
- A field labeled 'Select name for new workspace:' with the placeholder 'Use workspace template'.
- A dropdown menu labeled 'Use workspace template:' containing the path 'C:\TOSCA_PROJECTS\ToscaCommander\Standard.tsu'.

At the bottom right of the dialog, there are 'OK' and 'Cancel' buttons. To the right of the dialog, there is a message: 'In Real time we have multiuser workspace.' with a Tricentis logo.

IF it does select this then it would be **BLANK WORKSPACE** first
....>browse and select the one which we downloaded from
TOSCA tool:



This “.tsu”file contains the following---Explain

Note:

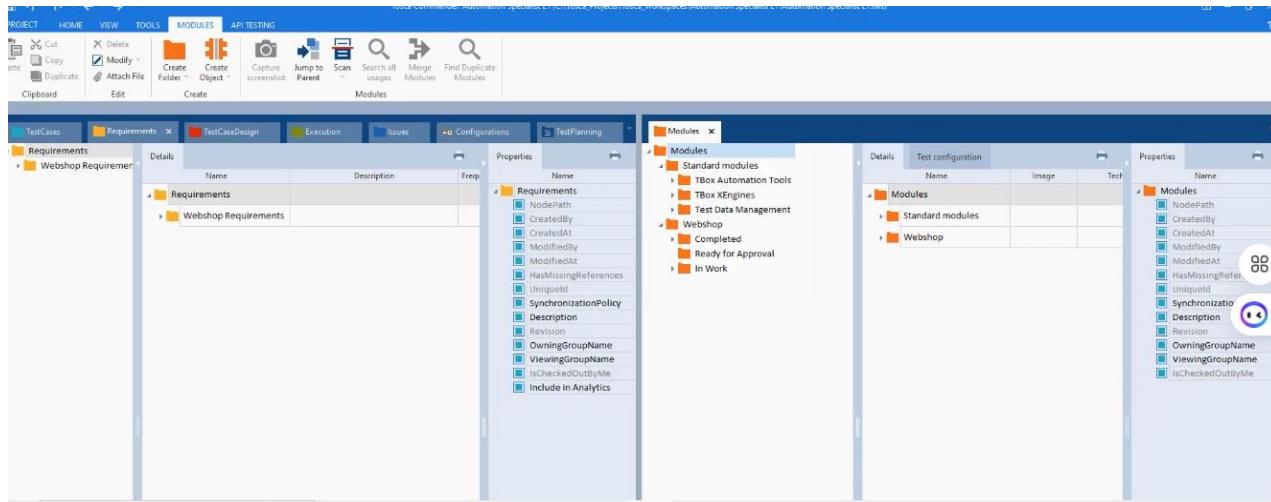
1) This is an Example of Demo Web shop.

2) Google.com

3) Yahoo.com/Gmail—signing and login

4) Zoho.com

5) Yatra.com



**Workspace can help you to PLAN, CREATE AND EXECUTE
YOUR TEST :**

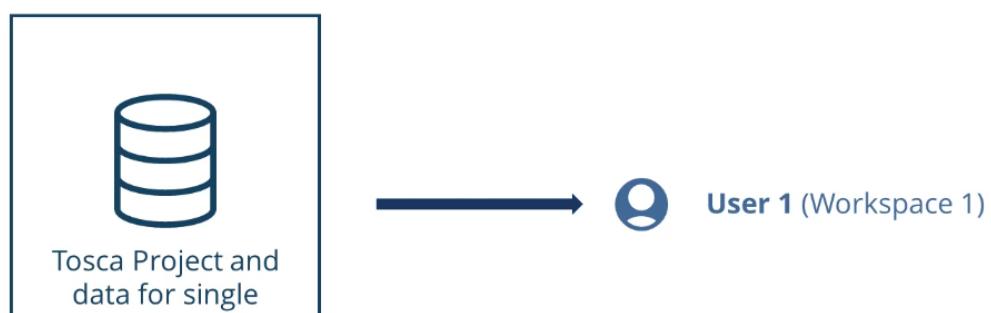
Sections that help you:

Plan
Create
Execute

We Talk about Multiuser Workspace:

Firstly about single user and then multiuser Explain

REPOSITORY MULTIUSER WORKSPACE



REPOSITORY MULTIUSER WORKSPACE



But how ever TOSCA recommend Single user workspace--Best Practice



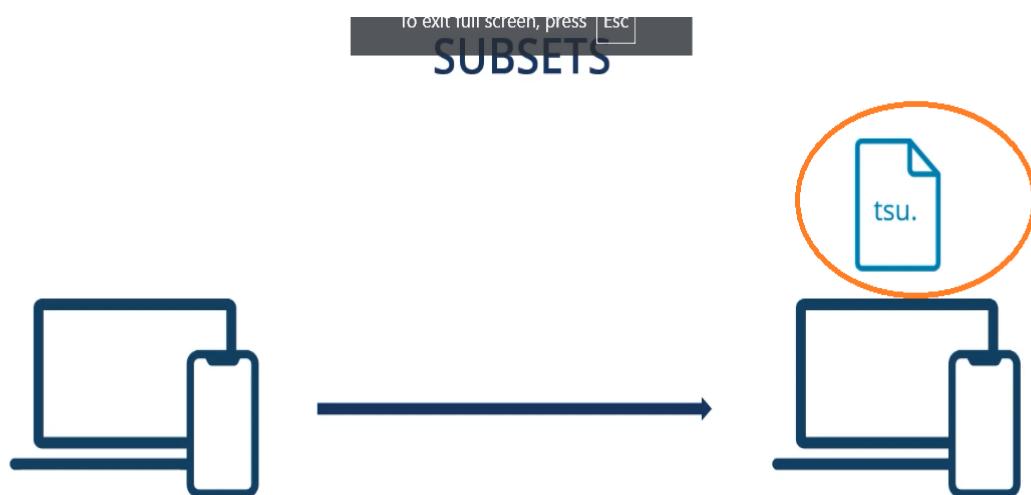
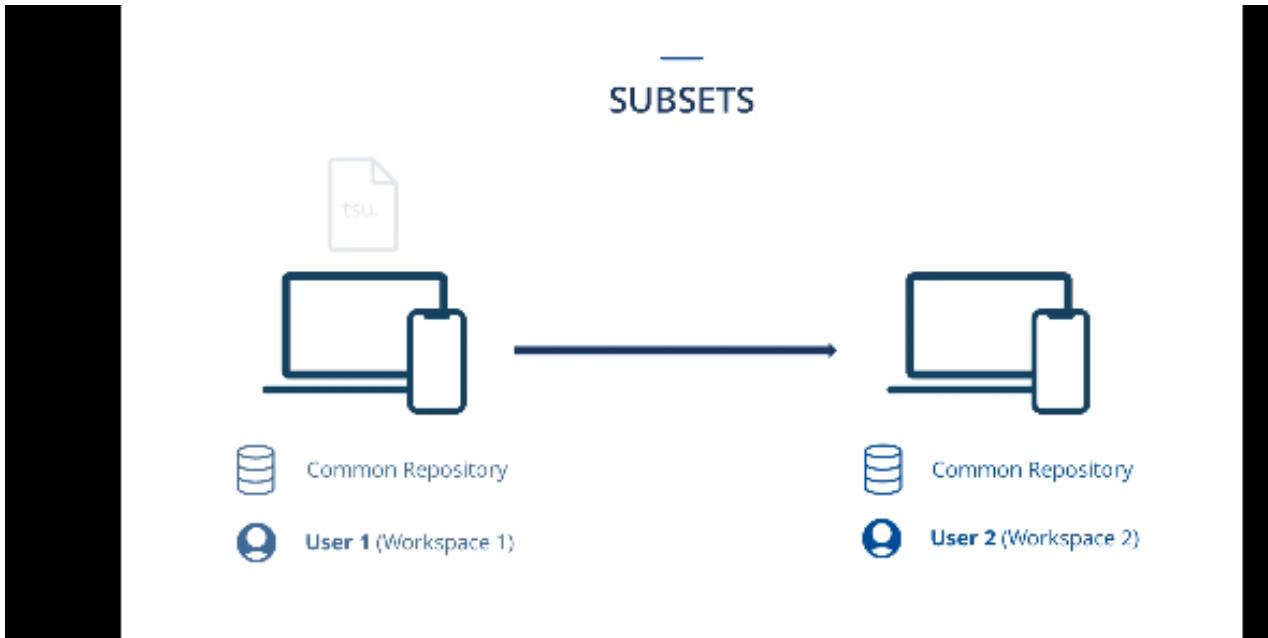
Assign only one user per Workspace

It is best practice to assign maximum one user per Workspace

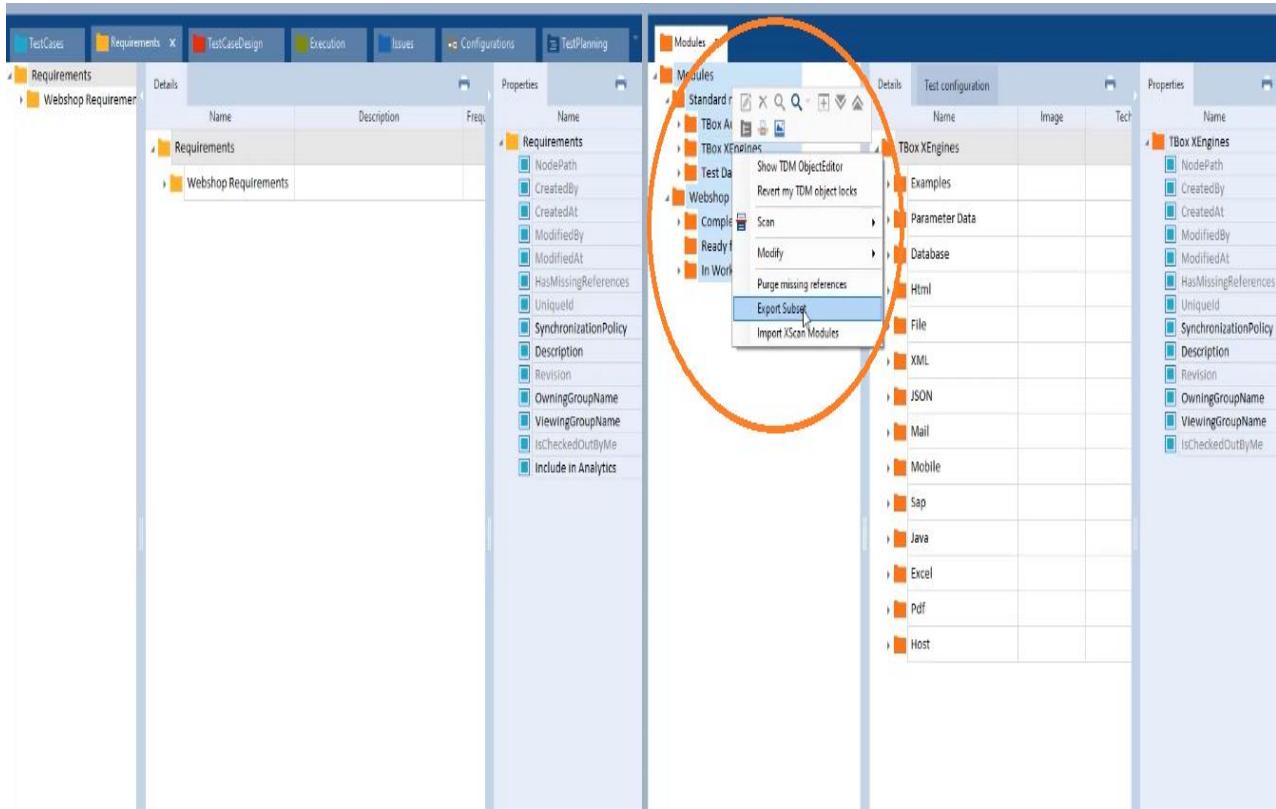
But however You can Export and Import form Project to Another with the help of SUBSETS:

In case you want to transfer some elements form one common Repository to another can be

Done easily:

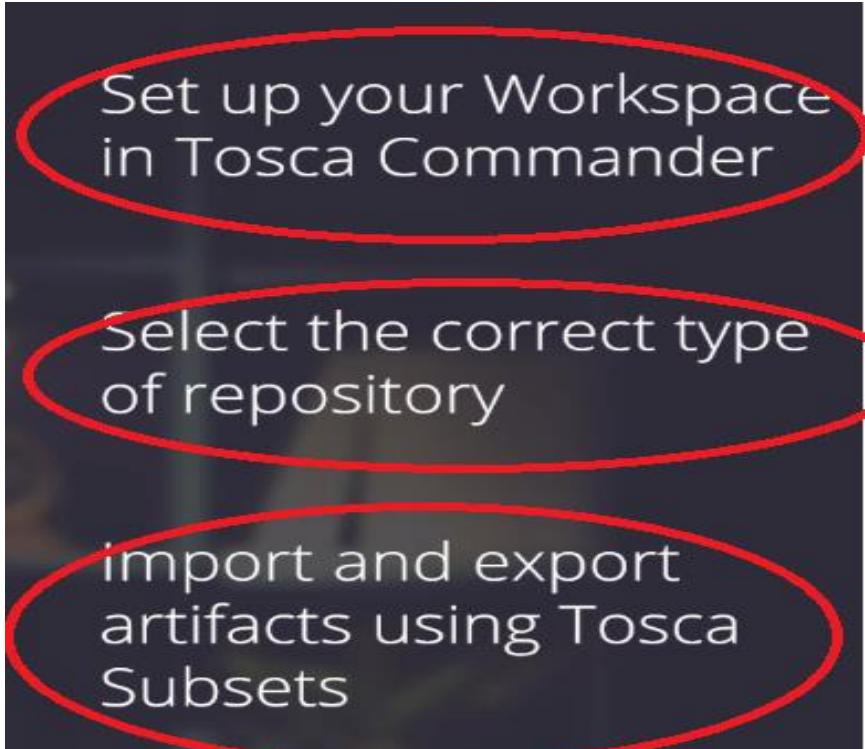


You can Export the subset by selecting all of them:



In case of Import :

Hence do the following correctly:

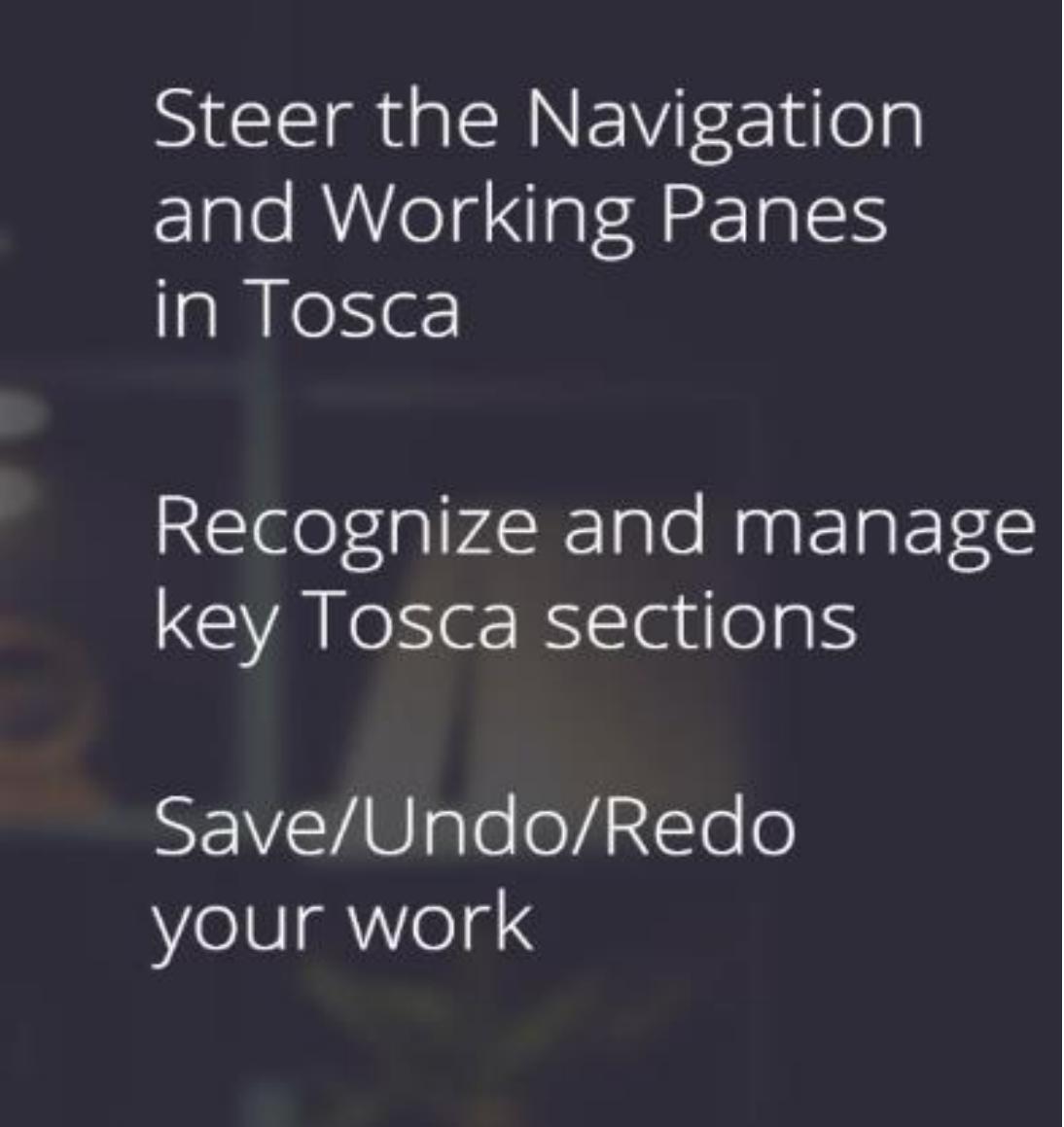


Set up your Workspace
in Tosca Commander

Select the correct type
of repository

import and export
artifacts using Tosca
Subsets

Now we will navigate you to this section:



Steer the Navigation
and Working Panes
in Tosca

Recognize and manage
key Tosca sections

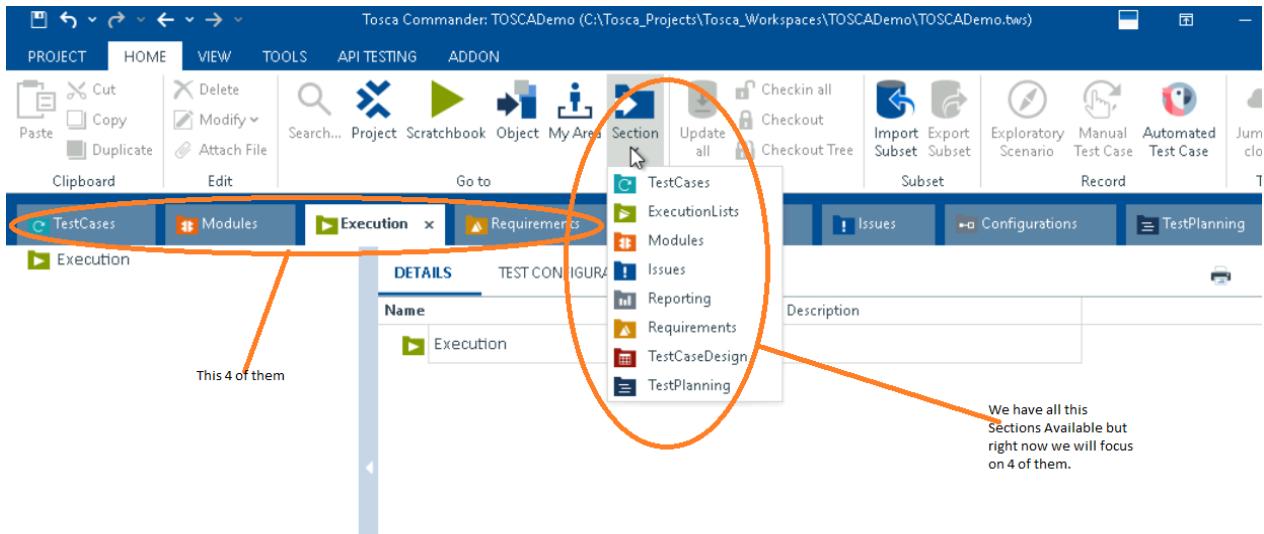
Save/Undo/Redo
your work

Now you can add columns by right clicking on column: Choose Column chooser and add as many columns you want and show them how to delete it to by dragging column down and release as shown next image:

The screenshot shows the Tricentis Academy interface with the URL [academy.tricentis.com/learn/courses/567/tricentis-tosca-fundamentals-automating-web-application-testing-as1/lessons/...](https://academy.tricentis.com/learn/courses/567/tricentis-tosca-fundamentals-automating-web-application-testing-as1/lessons/). The page title is "Tricentis Academy > My Courses and Learning Plans > Tricentis Tosca Fundamentals – Automating web application ...". The navigation bar includes "Modules", "TestCases X", "Execution", "Requirements", "TestCaseDesign", "Issues", "Configurations", "TestPlanning", and "Data Integrity Testing". The main content area displays a table with columns: Details, Test configuration, Name, Value, ActionMode, DataType, and WorkState. A context menu is open over the "ActionMode" column header, with the option "Column Chooser" highlighted and circled in orange. To the right, a "Properties" panel shows various configuration settings for the selected item.

This screenshot shows the same interface as the previous one, but with a second context menu open over the "ActionMode" column header. The "Delete" option is highlighted and circled in orange. The rest of the interface and the properties panel are identical to the first screenshot.

We have 8/9 different section (Data Integrating Section-older version) of which we will talk on 4 section now:



1) Modules: Show them you can arrange modules as you like ---just drag and release

The screenshot shows the 'Modules' section in Tosca Commander. On the left, there's a tree view of modules under 'Standard modules' and 'Webshop'. A large orange circle highlights this tree view. In the center, there's a table view with columns: Name, Image, TechnicalID, TypeInfoDescription, and ValueRange. A specific row in the table is highlighted with a blue selection bar. A callout points to this row with the text 'Modules'. Another callout points to the entire table area with the text 'Folders with technical information of your SUT'. On the right, there's a 'Properties' pane showing various properties for the selected module.

Modules are Maintain Technical stuff—means It is more like object Repository more advance one compare with some other tool.

2)Test cases: Test cases where you manage input values, your business logic, Test data and even the flow of Test case are managed here.

The screenshot shows a software application window with a dark blue header bar containing several tabs: Modules, TestCases (which is highlighted in light blue), Requirements, TestCaseDesign, Issues, Configurations, TestPlanning, and Data integrity Testing. Below the header is a sidebar with a tree structure showing 'TestCases' expanded, with 'AS1' selected. Under 'AS1', there are nodes for 'Additional Material', 'Library', 'Lessons', 'Constraint', 'Dynamic Buffer', 'ResultCount', 'Repetition', and 'Exercises'. To the right of the tree, there is a table titled 'Test configuration' with columns for Name, Value, ActionMode, DataType, and WorkState. One row in the table is highlighted in light blue, corresponding to the 'AS1' node in the tree. A callout box with a yellow border and black text points to the 'AS1' node in the tree, containing the word 'TestCases'. Another callout box with a blue border and white text points to the right of the tree, containing the instruction: 'Input your Business information to instruct Tosca to automate your SUT'.

3)Execution: You can Execute your testcases here.

The screenshot shows the Tricentis Academy interface with the following navigation path: Tricentis Academy > My Courses and Learning Plans > Tricentis Tosca Fundamentals - Automating web application ...

The main menu includes: Modules, TestCases, Execution (highlighted), Requirements, TestCaseDesign, Issues, Configurations, TestPlanning, and Data Integrity Testing.

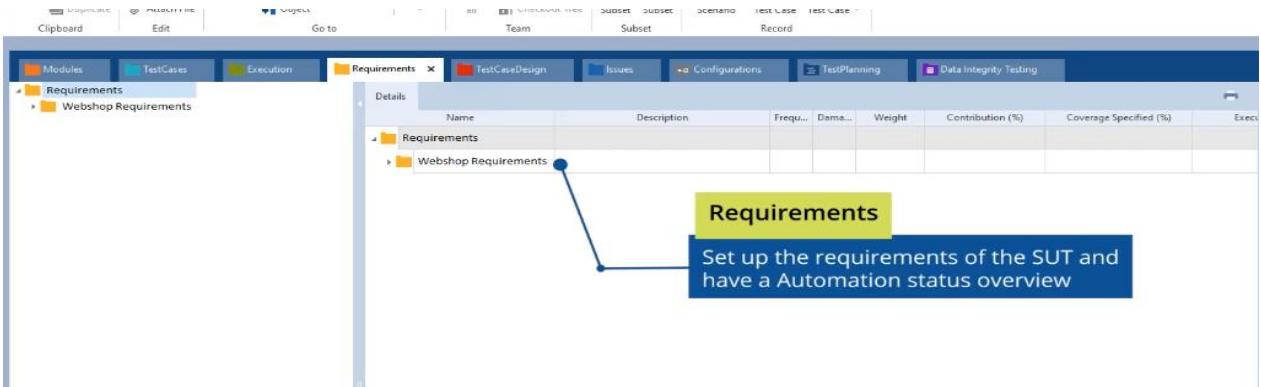
The Requirements section is active, displaying a table titled "Test configuration". The table has columns: Name, Number of Agents, and Description.

The table data includes:

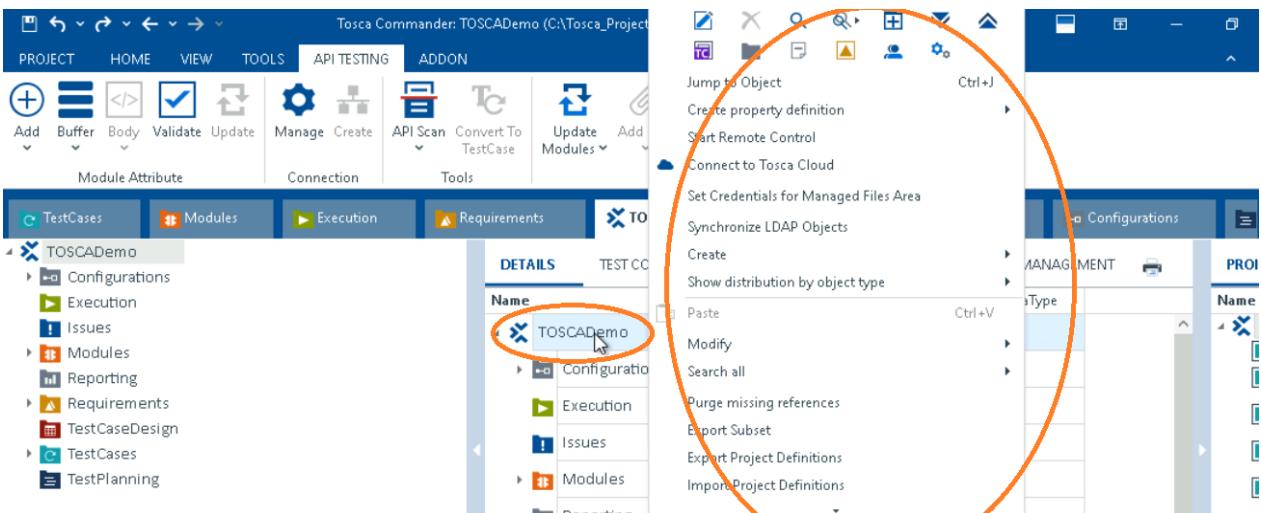
Name	Number of Agents	Description
Execution		
Execution_Import2111...		
ExecutionLists		
Exploratory Testing		
Interactive Testing		

An annotation highlights the "ExecutionLists" row with a yellow box labeled "ExecutionList". A callout box below it contains the text: "Run your test, log the results over time and identify bugs".

4) Requirements: Manage your Business Requirements both you Testcases and Execution suites/Sections can me mapped in Requirement section so that it give you the traceability whenever they are Executed. It says what % they are executed etc.



Now select the Project --Say DEMOPRJ the one To Create the workspaceand then Right click on it and Explain all of them in detail:



Explain about undo and redo works only when the project is not saved.

Tosca Automation Extension in Browser:

1. As with Tosca Xsan you can create X Browser modules in order to test Web applications.
2. In order to work with Browser Engine.
3. All Browser are available—Chrome, Edge, Firefox etc.

Use the link below to do so:

https://support-hub.tricentis.com/open?id=show_search_result&lang=en&pah=%2Ftosca%2F1520%2Fen%2Fcontent%2Fengines_3.0%2Fxbrowser%2Fxbrowser_scan.htm&product=tosca&sessionRotationTrigger=true&type=product_manual&version=16.0

lesson 3:

Best Practice:

- 1) Naming conversation and Description later like for example below:



Keep your project readable and make it easier for everyone:

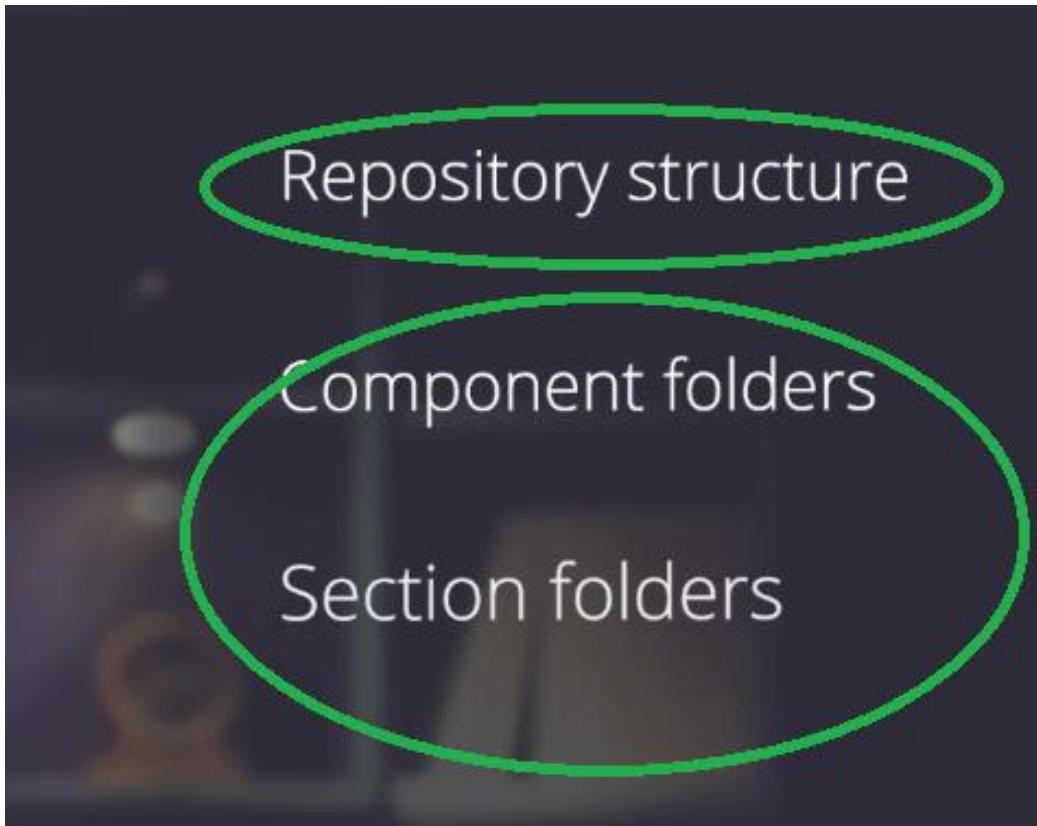
✗
Modules
 Insurance Calculator
 General
 Header, Insurance Calc.
 Navigation
 Insurance Calculator Footer



✓
Modules
 Insurance Calculator
 General
 Insurance Calculator | Header
 Insurance Calculator | Navigation
 Insurance Calculator | Footer



2) Repository structure which contains Component folder and Section folders under that:



Like Example:



3)

Four eyes principle:

In Work

Ready for Approval

Approved / Completed

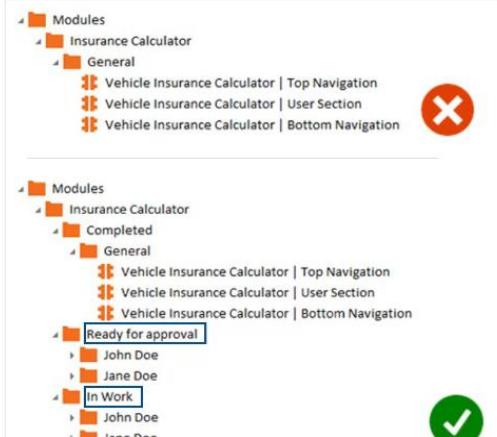
====Explain Ready for

Approval and Approval Briefly



FOUR - EYES PRINCIPLE

Use the three approval stages for artifact creation:



**By doing so what we gain ---Easier maintenance,
Greater efficiency and Higher readability in Agile or water fall**

model but in differs company to company and project to project.

Lesson 4:

Use and navigate
the System Under Test

Explaining about
Shopping web site-
Amazon/flipcard/Demo
shopping.

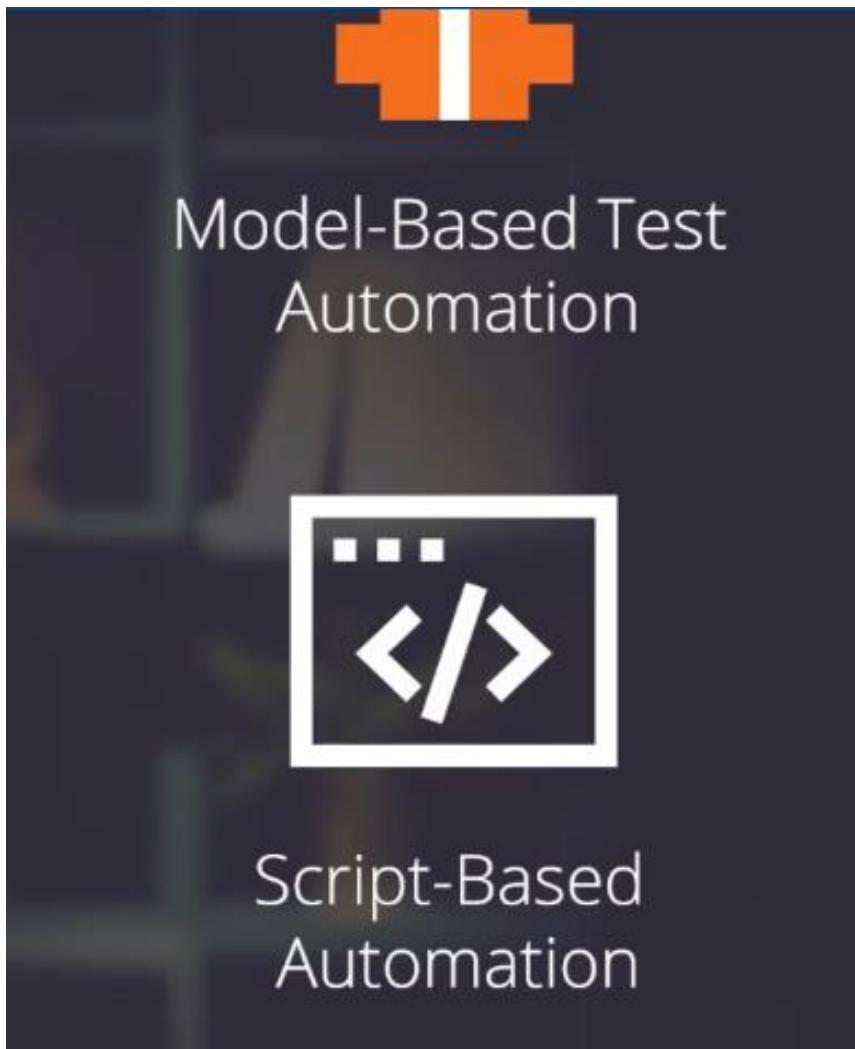
Install the correct
browser plug-in

All of them must install
Tricentic Browser plug-in

SECTION-2:

Lesson-5:

Tosca uses Model -Bases Test Automation instead of Script Based Automation.



But Model-Based Test Automation ---Explain in Detail using below images

MODEL-BASED TEST AUTOMATION

Extract Interface Model



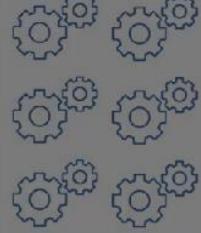
Feed Data

Technical Information



Automation Model

Business Information



Test Cases

Automate Interfaces (UI, API)

Create Test Cases



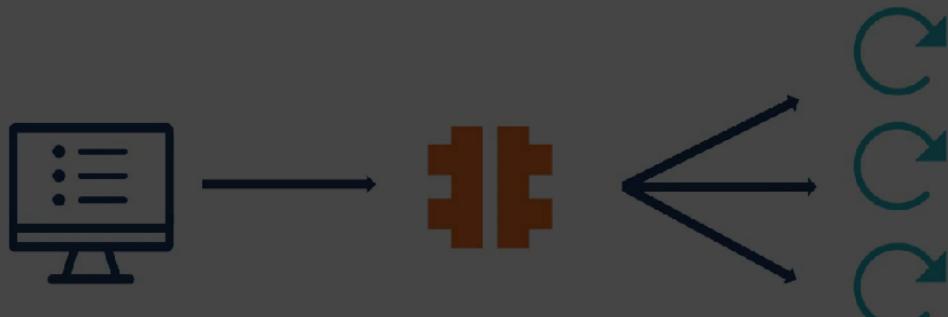
1:20 / 4:26



Developers: Provide information to the system and identify focus area

Business: View, control, and update test data

— REDUCED MAINTENANCE

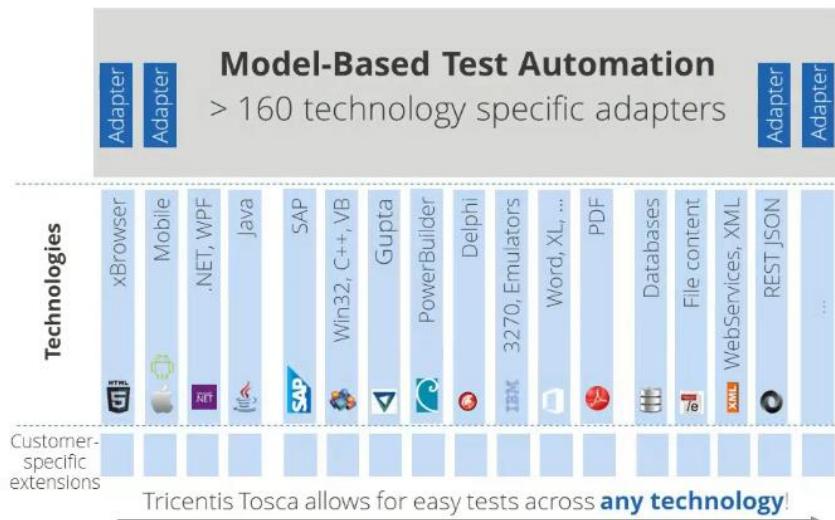


System

Module

Test Cases

TRICENTIS MODEL-BASED TEST AUTOMATION



Example this site can be easily Automated including UI as well as there API's

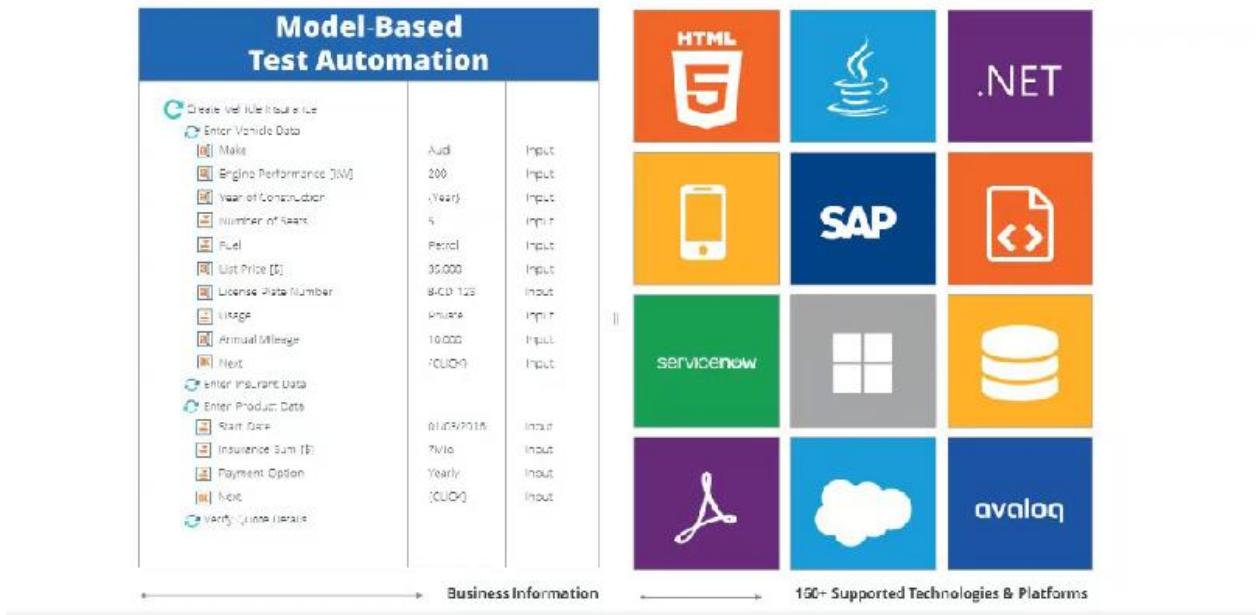
The screenshot shows two windows side-by-side. On the left is a Windows application titled 'Demo Web Shop - iScan' with a 'BASIC VIEW' tab selected. It displays a hierarchical tree view under 'Live Selection' with categories like 'Books', 'Computers', 'Electronics', and 'Apparel & Shoes'. On the right is a web browser window titled 'DEMO WEB SHOP Sample Application by Tricentis'. The page features a navigation bar with links for 'BOOKS', 'COMPUTERS', 'ELECTRONICS', 'APPAREL & SHOES', 'DIGITAL DOWNLOADS', 'JEWELRY', and 'GIFT CARDS'. A sidebar on the left lists 'CATEGORIES' such as Books, Computers, Electronics, Apparel & Shoes, Digital downloads, Jewelry, and Gift Cards. Another sidebar lists 'MANUFACTURERS' with a single entry for 'Tricentis'. The main content area features a large banner with the text 'Speed changes everything.' and 'Tricentis A...', along with a 'Subscribe' button. Below the banner, a section titled 'POPULAR TAGS' lists terms like apparel, book, camera, compact, computer, cool, digital, gift, jewelry, nice, shirt, shoes, TCP, and View all. A 'FEATURED PRODUCTS' section shows three items: a yellow 'gift CARD', a black laptop, and a black desktop computer. A 'NEWSLETTER' section on the right encourages users to sign up for the newsletter and provides a 'Subscribe' button. A 'COMMUNITY POLL' section asks if users like nopCommerce, with options for 'Excellent', 'Good', 'Poor', and 'Very bad', and a 'Vote' button.

ADVANTAGES OF MODEL-BASED TESTS

Model-Based			HTML 5 Browser	Script-Based	HTML 5 Browser
Enter Vehicle Data	Audi	Input			
Make	200	Input			
Engine Performance [kW]	(Year)	Input			
Year of Construction	5	Input			
Number of Seats	Petrol	Input			
Fuel	35.000	Input			
List Price [\$]	9-CD-123	Input			
License Plate Number	Private	Input			
Usage	10.000	Input			
Annual Mileage	(DD/MM)	Input			
Next					
Enter Insurance Data					
Enter Product Data	01/03/2016	Input			
Start Date	7MIS	Input			
Insurance Sum [\$]	Yearly	Input			
Payment Option	(DD/MM)	Input			
Next					
Verify Quote Details					
Price List	Select	Select			
Insurance Sum	1.535.22	Verify			
Gross Premium [\$]					
Business Information:					
<pre>Assert.AreEqual("1.535.22", driver.FindElement(By.XPath("//table[@id='PriceList']/tbody/tr[10]/td[2]/b")) .GetAttribute("Value"));</pre>					
Do you see the business information?					

TOSCA is Applicable to all Technologies like HTML, JAVA, .NET etc.

TRICENTIS MODEL BASED TEST AUTOMATION



Hence using Model-Based Test Automation --We can Test Application Effectively and Efficiently.

Lesson 6:

Explain about Standard Modules in TOSCA and how to Use them: WindowsOperations, Process operations and SAP and other Specialization modules.

Standard modules: Explain

STANDARD MODULES

- Standard modules
 - TBox Automation Tools
 - ▶ Basic Windows Operations
 - ▶ Buffer Operations
 - ▶ Expression Evaluation
 - ▶ File Operations
 - ▶ Numeric Operations
 - ▶ Process Operations
 - ▶ Resource Handling
 - ▶ Selenium
 - ▶ Timing
 - TBox KEngines
 - ▶ Examples
 - ▶ Parameter Data
 - ▶ API
 - ▶ Database
 - ▶ Html
 - ▶ File
 - ▶ XML
 - ▶ JSON
 - ▶ Mobile
 - ▶ Sep
 - ▶ Java
 - CSV
 - BaseModules
 - ▶ Specialization
 - ▶ Remote Control

As TOSCA come up with more technical information as by adding Description column and show:

DETAILS

Name	Image	TechnicalID	TypeInfoDescription	ValueRange	Description
OpenUrl	Icon				Opens a URL with a browser. If the 'Browser' value is empty then the default default browser is taken.
Url	Icon				The URL which should be 'Browser' TestConfiguration is set then the default default browser is used.
UseActiveTab	Icon				The URL which should be 'Browser' TestConfiguration is set then the default default browser is used.

Show them how to drag and place it in Testcase (As you can drag anywhere you want):

TestCases

Name	Value	ActionMode	DataType	WorkState
AS1				
1 - Approved TestCases				
2 - Ready for Approval				
3 - In Work				
Open and Close the application				

Best Practice:



Use Standard Module When Possible

Standard Modules are known for their resilience and reliability.

Additionally, Tricentis updates and expands the repository of Standard Modules with each subsequent Tosca release.

Lesson 7, 8,9 &10:

Xscan:



In XScan we will see the below:

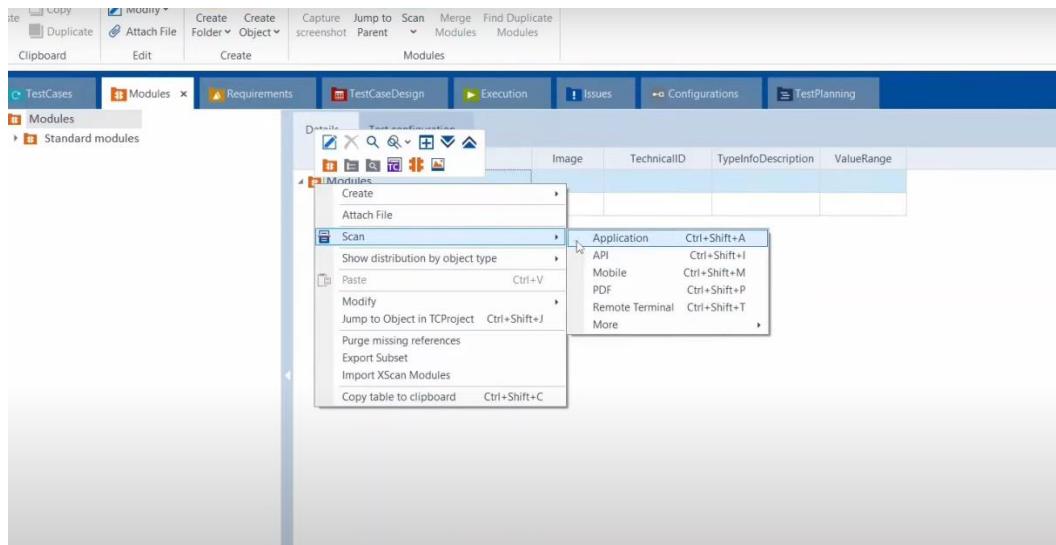
- Navigate XScan
- Use the different XScan Views and Buttons
- Identify control by Properties
- Identify control by Anchor
- Basic View, Advanced View, 'Selection on Screen' view option & Condensed View of XScan
- Save vs Finish Scan
- Module Creation

From the Above full sentence: Use the different XScan views and Buttons

Basic view, AdvancedView, 'Selection on Screen' view option and Condensed view of Xscan.

Now explain it in the project.

Now right click module Section → Scan → Application

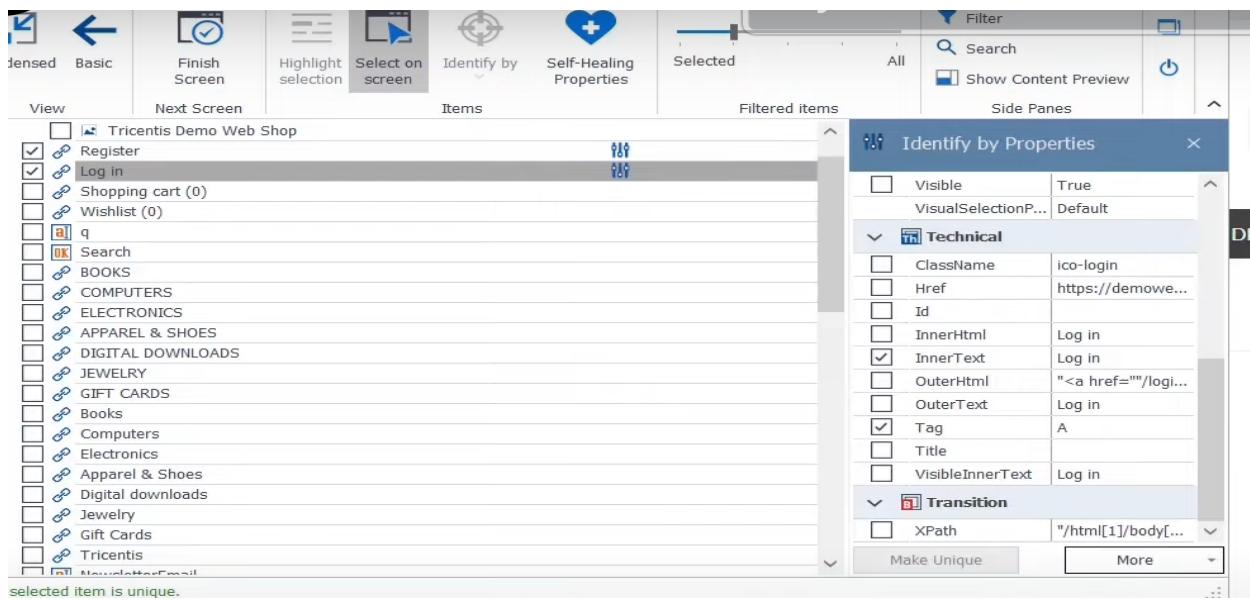


Generally, I will show all the application in your system, then select the one which you want say—Demo workshop and click on scan

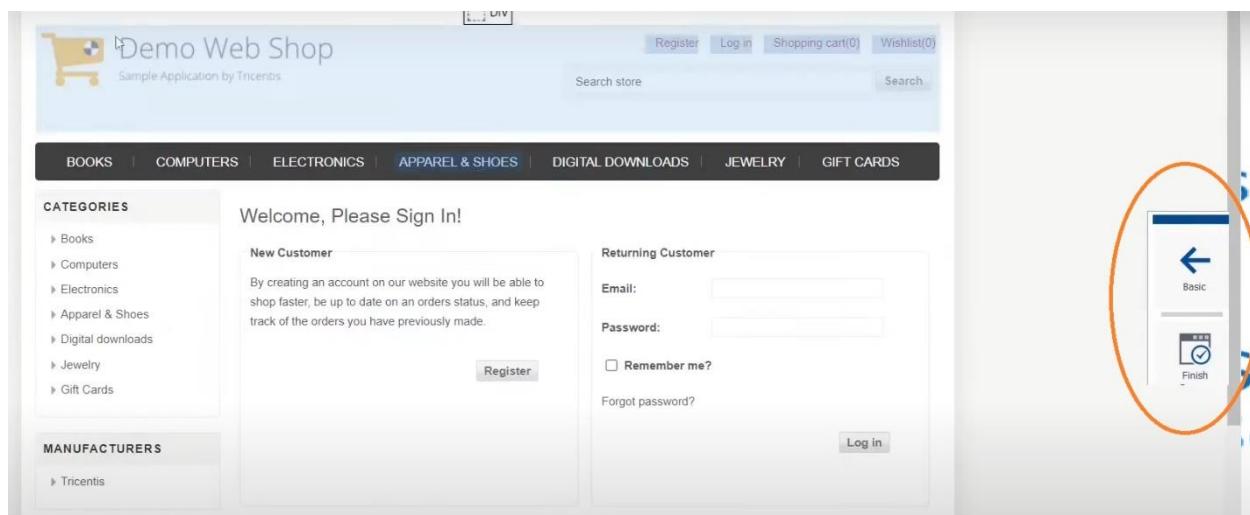
We get this” Basic view of xscan” and start selecting it and save so that we can see it in our project

Click on controls in your application to add them

Advance View—select Select on Screen and add some more button. Once done again we need to select on screen in order to deselect it.



Condensed View: Explain how task is done and Right side Basic and Finish –Tell them.



**Note: Explain—once done difference between
“Save”/”Finished Screen”/”Highlight”—select and
reselect/”Rescan”—Explained belowetc. in Detail.**

***Very important--structure Explain**

1)Once module section is done

2)Go to Test case:

Create Folder (Login Demo-Right click 1st one),Testcase(Login Testcase—Right click 4th one)then Lastly Test steps(Demo Web shop)

- ▲  Login Demo
 - ▲  Login Testcase
 - ▶  Demo Web Shop

**3)Drag the module which you have created in to Test caseand
see it is under Test Step(Log in)**

- ▲  Login Demo
 - ▲  Login Testcase
 - ▲  Demo Web Shop
 - ◀  Log in

**Then Enter Value—x/Click and Action mode-Input and Data
Type-String and save**

The screenshot shows a software interface with a left sidebar containing a tree view of test cases under a folder named 'AS1'. The tree includes nodes for 'Approved TestCases', 'Ready for Approval', 'In Work', 'Login Demo', 'Login Testcase' (which has 'Demo Web Shop' and 'Log in' as children), and 'LOGIN PageNAVi'. To the right of the sidebar is a 'DETAILS' section with a table. The table has columns for Name, Value, ActionMode, DataType, and WorkState. A single row is present with 'Log in' in the Name column, 'x' in the Value column, 'Input' in the ActionMode column, 'String' in the DataType column, and an empty cell in the WorkState column.

DETAILS				
Name	Value	ActionMode	DataType	WorkState
Log in	x	Input	String	

Now right click on In Testcase window select folder → Run in Scratch Book.

And view the results pass/fail and down Elapsed time N Remaining Time also –Explain.

Rescan on the same page or new window can be done by Right clicking on module Rescan.

The screenshot shows a software interface with a toolbar at the top and a main window divided into two panes. The left pane contains a navigation bar with 'Modules', 'TestCases', 'Requirements', 'Execution', 'TestCaseDesign', 'Issues', 'Configurations', 'TestPlanning', and 'Data Integrity Testing'. Below this is a tree view of modules: 'Standard modules', 'Webshop' (with 'Completed', 'Ready for Approval', 'In Work', 'Customer', and 'Products' as children). The right pane displays a table titled 'Details' with columns 'Name', 'Image', 'TechnicalID', 'TypeInfoDescription', 'ValueRange', and 'Description'. Under the 'Customer' category, there are three entries: 'Exercise 03 - Log In Page', 'WebShop - Log in Page', and 'WebShop - Log out'. A blue arrow points from a callout box to the 'WebShop - Log in Page' entry. The callout box contains the text: 'If you need to make changes to already existing controls update a Module add new controls to a Module'.

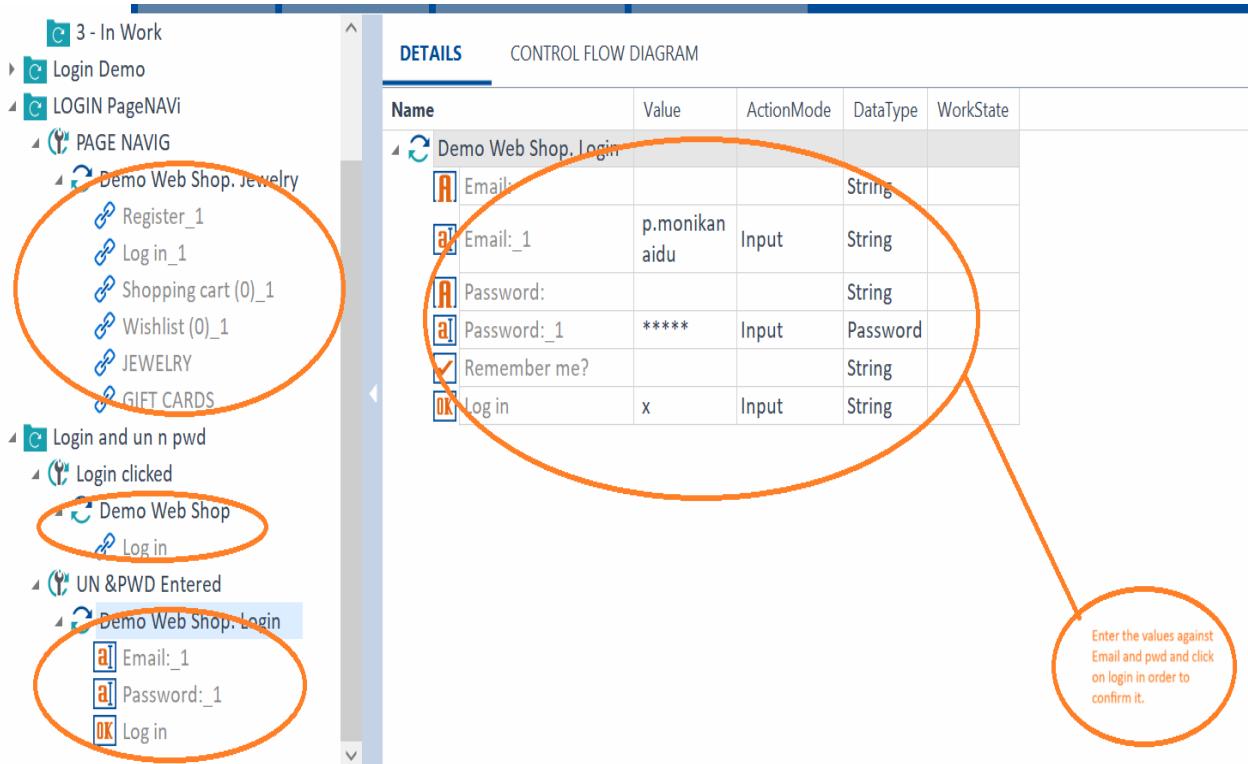
Same window adding few more fields:

The screenshot shows the TestComplete application interface. The top navigation bar includes tabs for TestCases, Modules (highlighted with a red circle), Execution, TestCaseDesign, and Requirements. The main content area displays a tree view of test modules. A context menu is open over an item named 'Demo Webshop Navig page'. The menu is titled 'SELF-HEALING PROPERTIES' and contains several options: Attach File, Capture screenshot, Rescan (highlighted with a red circle), Rescan WebDriver, Cut, Copy, Paste, Modify, Reset property, Jump to Object in TCFolder, Search all, and Purge missing references.

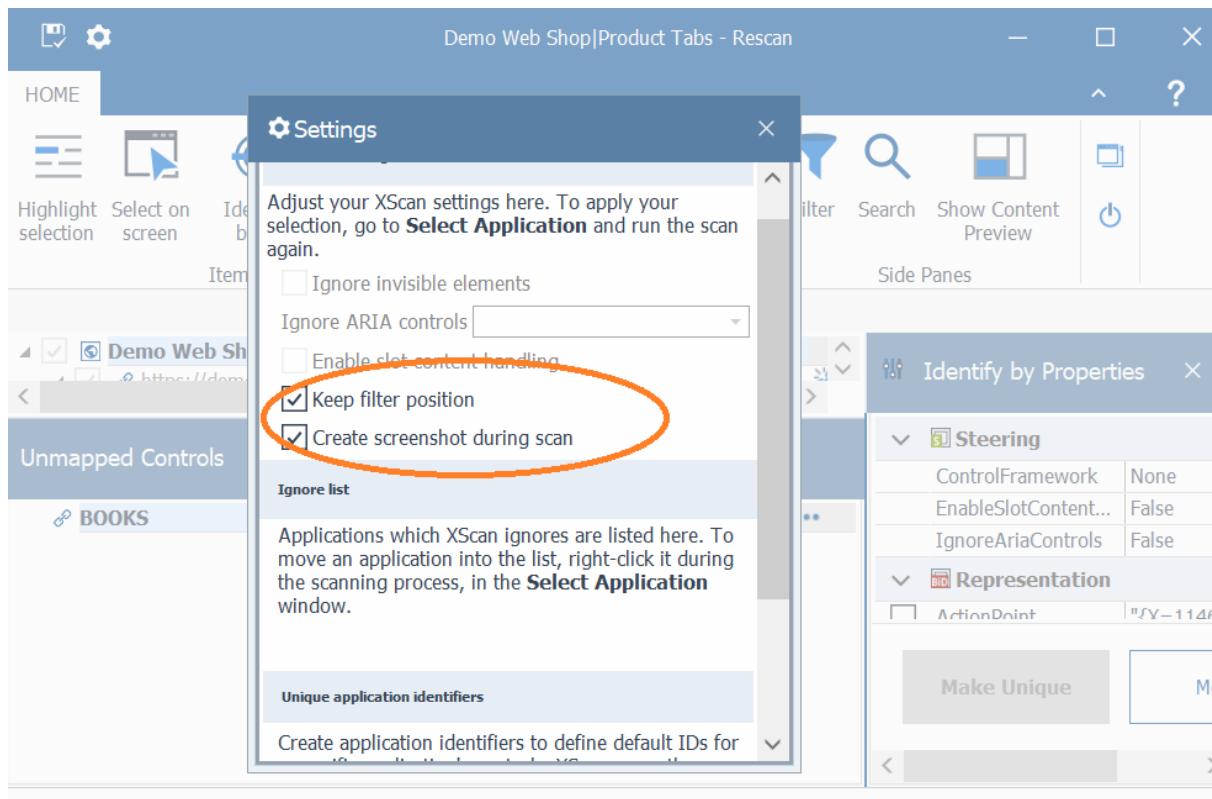
Different window Rescan after login:

This screenshot shows the TestComplete interface again, with the Modules tab selected. The left sidebar shows a tree of test modules, including 'Webshop' and 'Demo Web Shop'. A context menu is open over a 'Log in' element within the 'Demo Web Shop' module. The menu is titled 'TEST CONFIGURATION' and includes options like Attach File, Capture screenshot, Rescan (highlighted with a red circle), Rescan WebDriver, Cut, Copy, Paste, Modify, Reset property, Jump to Object in TCFolder, Search all, Purge missing references, Export Subset, and Drag this object.

Once done we need to Drag it to Testcase as below for both above and Run in ScratchBook:



Xscan—Settings we have these two options--Explain



Identify Controls by Anchor in XScan:

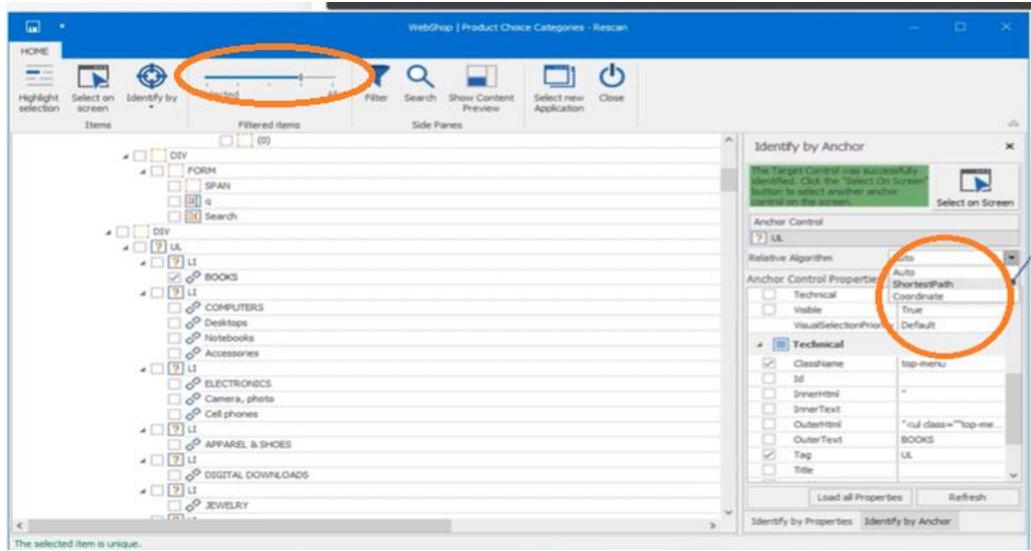
- # Identify Controls
- Identify controls by properties
 - Identify controls by specific properties default identification method in Tosca
 - Can use unlimited number of properties for identification
 - Use properties that are necessary and unique
 - Identify controls by Anchor
 - Tosca looks for an Anchor control and search for target control in relation to it
 - Create a Control Group

Explain about Tile property to make it unique say * and here I change in name to “Demo Web Shop” and add properties by tick mark also Explain.

The screenshot shows the Tricentis Studio interface with the 'Identify by Properties' dialog open on the right side. The 'Representation' tab is selected, showing properties like ActionPoint, Context, ControlArea, Enabled, Focused, Hwnd, InteractiveElement, IsSteerable, Technical, Title, and Visible. The 'Title' property is highlighted with a red circle. The 'Technical' tab is also visible. In the 'Items' pane on the left, a list of items is shown, with 'Demo Web Shop' selected and highlighted with a red circle. The URL 'https://demowebshop.tricentis.com/' is also selected in the list.

In Scan → Identify by → Anchor select and Drag and drop and select Relative Algorithm—Auto and explain about other options too and Selected ----ALL also drag and explain.

Note: It is not must in each and every project we need to use Anchor Explain.



Once Anchor is selected you can see like this:



Control Groups:

At the module level select all and right click on the selection to Convertit into Control Group(Product Categories-links)---in order to group all the like things categories wise.

Modules

- Standard modules
- Webshop
- Demo Webshop Navig page-2
- Demo Web Shop just login-1
- Demo Web shop login-unnpwd-3
- Demo Web Shop | Product Tabs

DETAILS

chnicalID	TypeInfoDescription
	{CI}

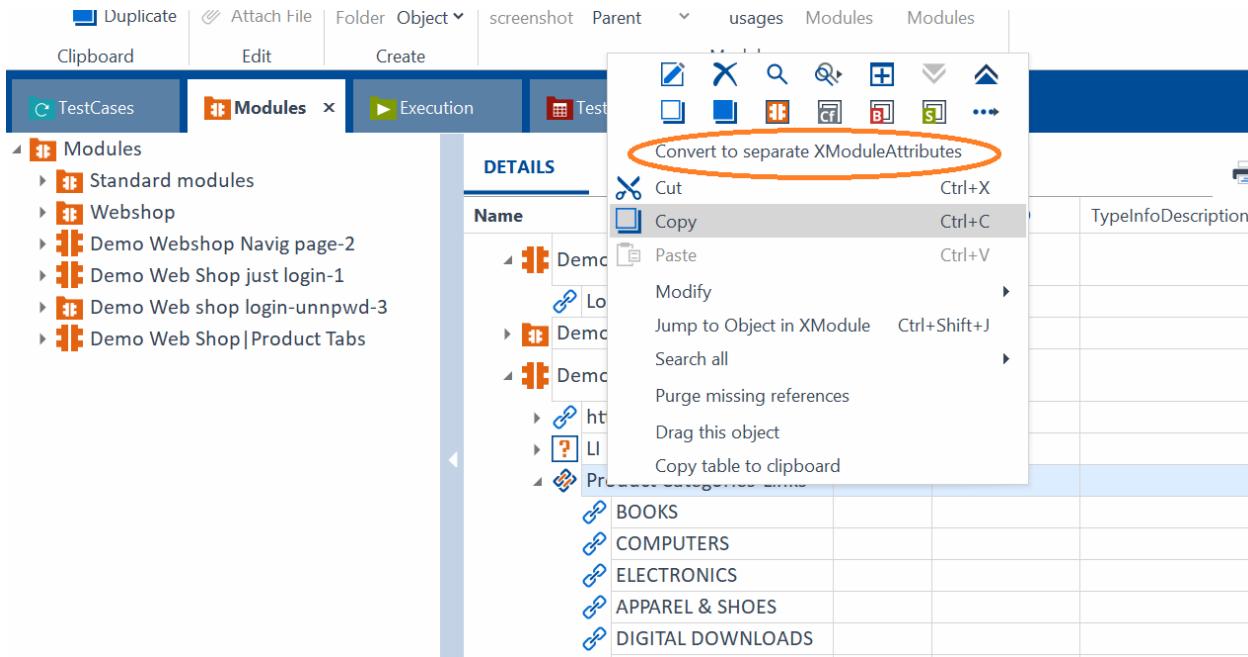
Modules

- Standard modules
- Webshop
- Demo Webshop Navig page-2
- Demo Web Shop just login-1
- Demo Web shop login-unnpwd-3
- Demo Web Shop | Product Tabs

DETAILS TEST CONFIGURATION

Name	Image	TechnicalID	TypeInfoD
Demo Web Shop just login-1			
Demo Web shop login-unnpwd-3			
Demo Web Shop Product Tabs			
https://demowebshop.tr...			
LI			
Product Categories-Links			
BOOKS			
COMPUTERS			
ELECTRONICS			
APPAREL & SHOES			
DIGITAL DOWNLOADS			
JEWELRY			

Now again to remove it from Control Group:



SECTION-3:

Lesson 11:

Introduction to Test Cases and How to create it:

Introduction to Test Cases

- Navigate around the TestCase section
- Define & Create a folder structure for TestCases creation that reflects Workflow
- Elements containing sequence information
- Modules: Technical Information; TestCases Business Information
- Have proper naming conventions for TestCases
- Use the three approval stages for artifact creation (Planned: In-Work: Complete)

Modules are related to technical information and Testcase are related to Business Information.

The screenshot shows a software interface with a navigation bar at the top and a main content area below. The navigation bar includes tabs for TestCases, Modules, Execution, TestCaseDesign, and Requirements. The Requirements tab is currently selected, indicated by a blue background.

The main content area is divided into two sections: a left sidebar and a right panel. The sidebar contains a tree view of test cases:

- TestCases
 - AS1
 - Login Demo
 - LOGIN PageNAVi
 - Login and un n pwd
 - Zoho End to End
 - 11-In work
 - Session 11-Testcase Structure
 - Shipping Costs
 - Precondition
 - Process
 - Order product
 - Start checkout
 - Checkout process
 - Verification of Prices
 - Confirmation Prices
 - Verification of Sucess
 - PostCondition
 - Shipping Costs
 - Precondition
 - Process
 - PostCondition

Explained in Detail:

The screenshot shows a software interface for managing test cases. The top navigation bar includes tabs for TestCases, Modules, Execution, TestCaseDesign, and Requirements. The main area has two sections: 'DETAILS' and 'TEST CONFIGURATION'. The 'DETAILS' section lists various test cases and their sub-components. The 'TEST CONFIGURATION' section is a table with columns: Name, Value, ActionMode, DataType, and WorkState.

Annotations highlight specific features:

- A red oval labeled "Main and sub Folders" points to the tree view on the left, which shows nested folder structures like AS1, Login Demo, and Session 08-Testcase Stru.
- A red oval labeled "IN WORK" points to the "WorkState" column in the table, where the value for the selected row is "IN WORK".
- A blue oval labeled "3--Test Step Structure" points to the "Value" column of the table, which contains a complex JSON-like structure representing the test steps.
- A blue oval labeled "6 different Folders structure under Process Folder" points to the "Value" column of the table, which shows a list of six items under the "Process" category.
- An orange oval labeled "Test Case" points to the bottom right of the table area.

Table Data (approximate values from screenshot):

Name	Value	ActionMode	DataType	WorkState
Session 08-Testcase Structure	{...}			IN WORK
Shipping Costs				
Precondition				
Process	[{"Order product": {"Start checkout": {"Checkout process": {"Verification of Prices": {"Confirmation Prices": {"Verification of Success": {"PostCondition": {...}}}}}}}}]			

Lesson 12:

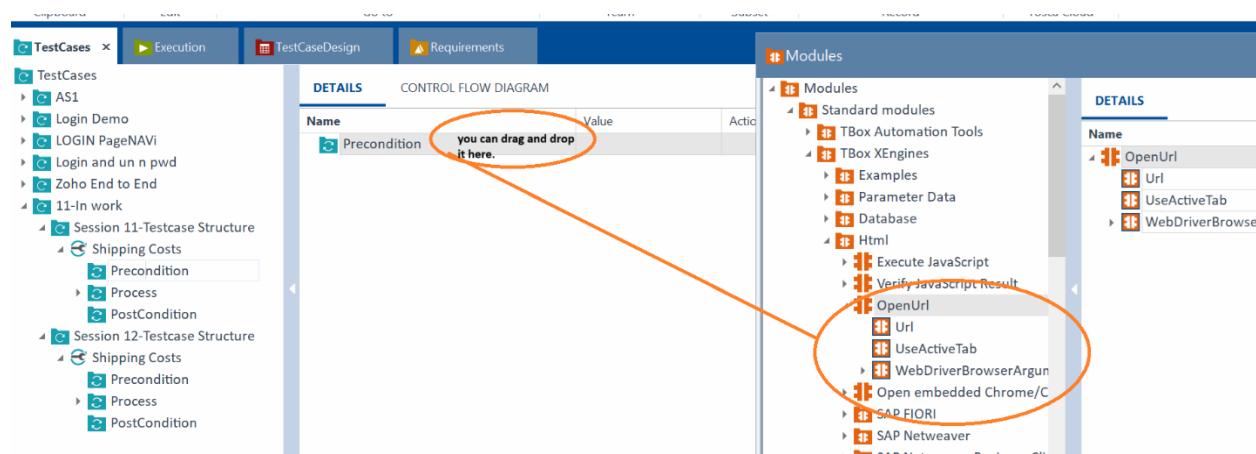
Create Test Steps

- Populate your Test Case with Test Steps
- Use Standard Subset (.tsu) Standard Modules to perform common steering
- Two ways of Test Step creation in Tosca

 1. Drag & Drop Module onto a Test Case folder
 2. Use function Add Test Step at Test Case folder levels

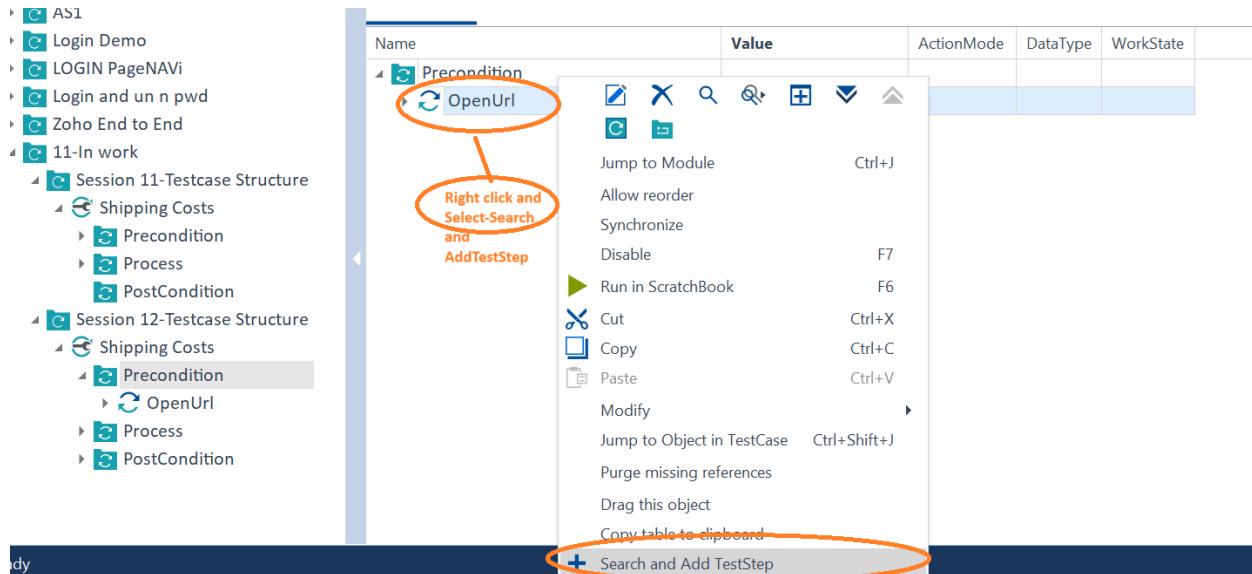
Explain in the Project:

You can place the module section on Right and drag and drop.

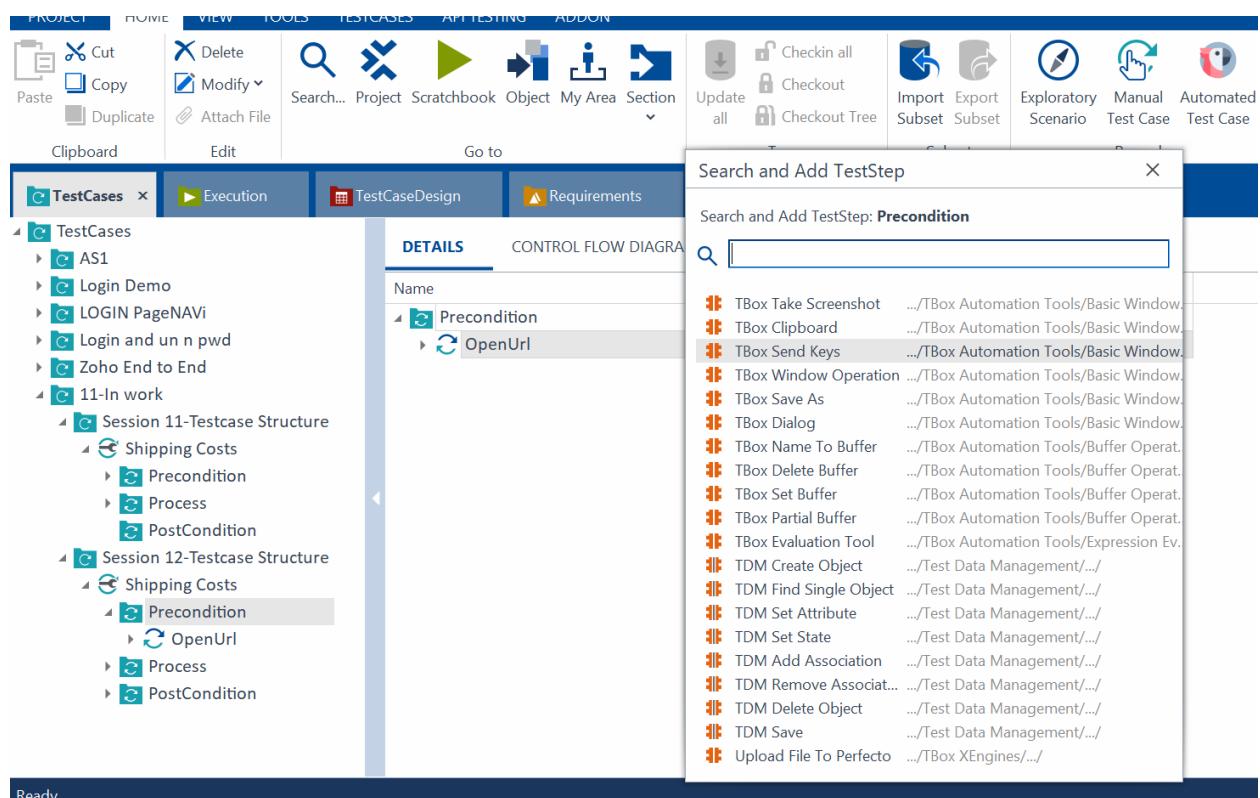


Or

- AS1
- Login Demo
- LOGIN PageNAVi
- Login and un n pwd
- Zoho End to End
- 11-In work
 - Session 11-Testcase Structure
 - Shipping Costs
 - Precondition
 - Process
 - PostCondition
 - Session 12-Testcase Structure
 - Shipping Costs
 - Precondition
 - OpenUrl
 - Process
 - PostCondition



We get this:



Or

Use CTRL +T and now use the module which you defined earlier to add it as TS:

The screenshot shows a software interface with a navigation bar at the top: TestCases, Execution, TestCaseDesign, Requirements, and Modules. The TestCases tab is active. On the left, there's a tree view of test cases, including 'TestCases', 'AS1', 'Login Demo', 'LOGIN PageNAVi', 'Login and un n pwd', 'Zoho End to End', '11-In work', 'Session 11-Testcase Structure' (which contains 'Shipping Costs' with 'Precondition', 'Process', and 'PostCondition'), and 'Session 12-Testcase Structure' (which contains 'Shipping Costs' with 'Precondition' (selected), 'OpenUrl', and 'Demo Webshop Navig pa'). On the right, under 'DETAILS', is a table titled 'CONTROL FLOW DIAGRAM'. The table has columns: Name, Value, ActionMode, DataType, and WorkState. It lists items like 'Precondition', 'OpenUrl', and 'Demo Webshop Navig page-2' (which is circled in red). Below this table are several rows with icons and names: 'Register_1', 'Log in_1', 'Shopping cart (0)_1', 'Wishlist (0)_1', 'JEWELRY', and 'GIFT CARDS'. The 'Precondition' row in the table is also circled in red.

Name	Value	ActionMode	DataType	WorkState
Precondition				
OpenUrl				
Demo Webshop Navig page-2				
Register_1			String	
Log in_1			String	
Shopping cart (0)_1			String	
Wishlist (0)_1			String	
JEWELRY			String	
GIFT CARDS			String	

Create one big one—and show them(End-to-End)[Just in order to have an idea]

Name	Value	ActionMode	DataType	WorkState
Session 12-Testcase Structure				
Shipping Costs				
Precondition				
OpenUrl				
WebShop Top Menu				
Process				
Order product				
Start checkout				
Navigate tp ShoppingCart				
Shopping Cart Procedures				
Checkout process				
Verification of Prices				
Confirmation				
Verification of Success				
PostCondition				
WebShop Order Successful				
WebShop Top Menu				
CloseBrowser				

Lesson 13:

Populate Test Step values for Test Cases:

Populate Test Step Value

- Populate your TestSteps with Values to instruct Tosca to steer the SUT according to the specified values.
- Use basic Values and basic DataTypes.
- TestStep values can be seen/entered from Test Case section of Tricentis Tosca.
- To view them, you just need to expand the TestSteps.
- The "Expand All" option can be used to expand all the TestSteps, TestStep Folders & TestCases.

Show Example: Same one Values added and Action Mode Explain about x,Demo*,Wait on, Input and Select.

Name	Value	ActionMode	DataType	WorkState
OpenUrl	https://demowebshop.tricentis.com/	Input	String	
UseActiveTab		Input	String	
Log in	x	Input	String	
Email:	p.monikanaidu@gmail.com	Verify	String	
Password:	Ugandhar@1975	Input	String	
Log in	x	Input	String	
Order product				
Navigate to Apparels & S...				
Product Categories	APPAREL & SHOES	Input	String	
Navigation to Blue Jeans				
Blue Jeans	x	Input	String	
Ordered Blue jeans				

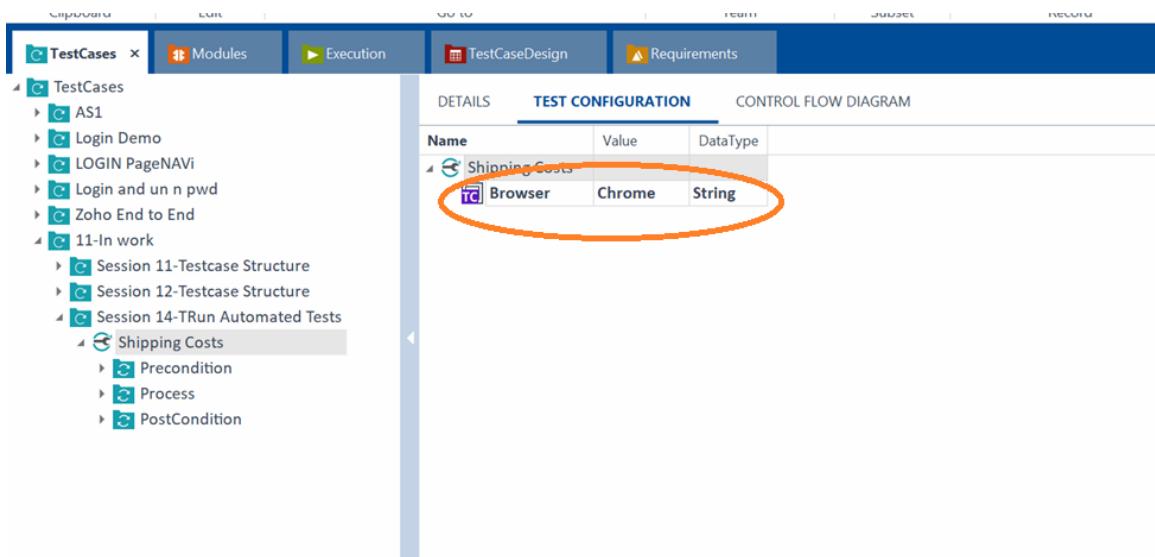
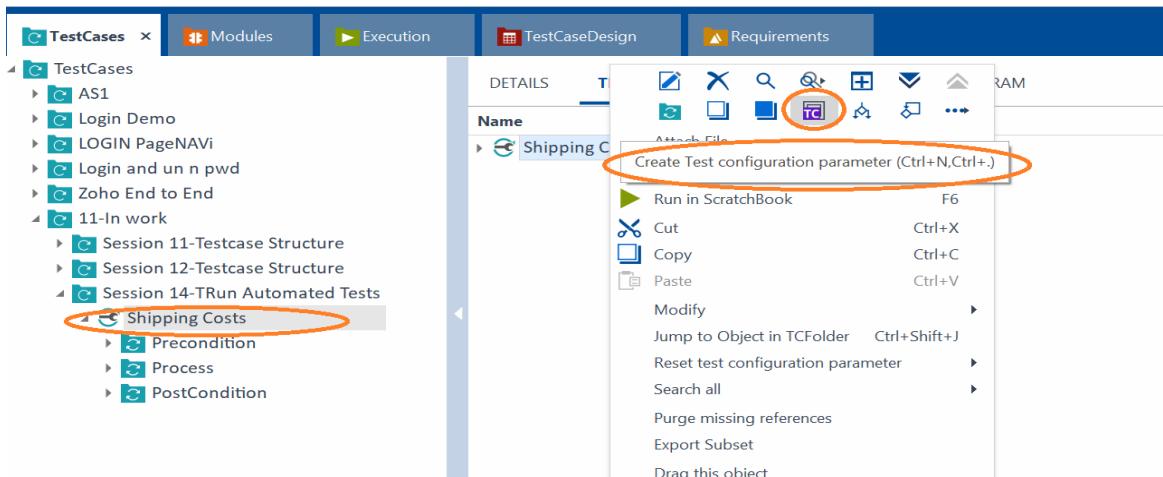
Lesson 14: Test Configuration Parameter (TCP) and how to Run the first Automated Test.

RUN YOUR FIRST AUTOMATED TEST

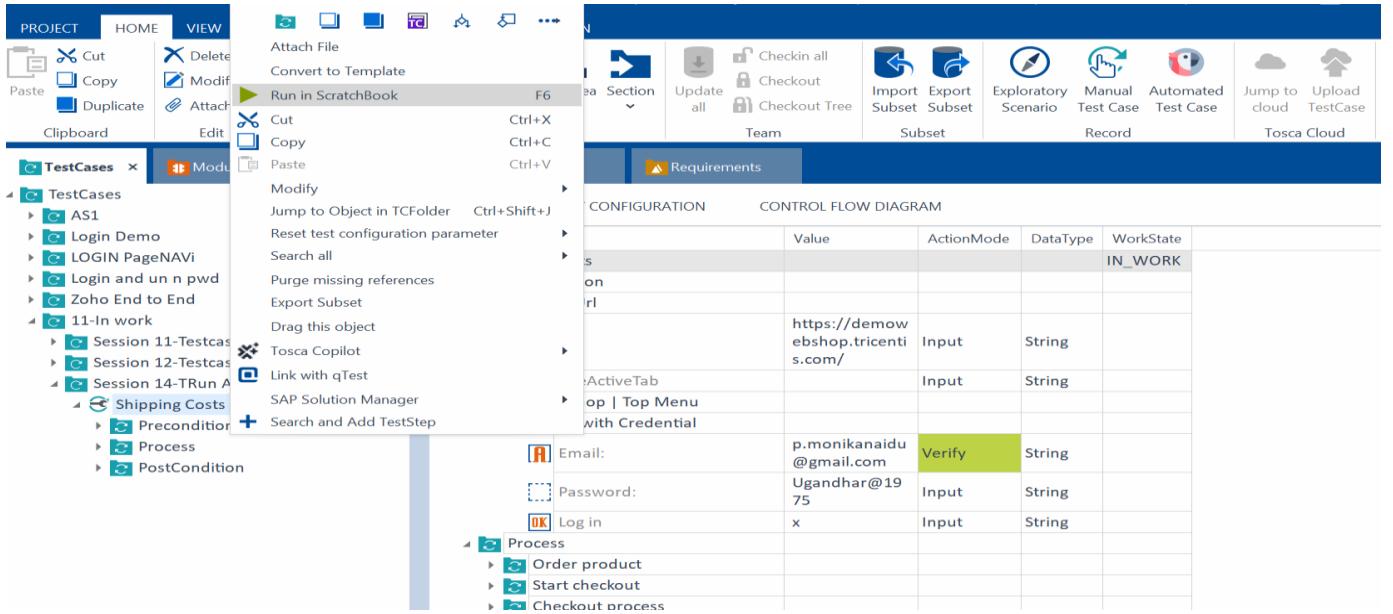
- **Test Configuration Parameter (TCP):** This is a parameter you can set for Tosca object. Example should run on a specific Browser, Various Request Numbers, Various Test Environments, Connected Identifiers (URLs) etc
- Simplify maintenance of your tests and avoid repetitions
- Populate a specific Test Case with TCP: Population at different levels like TCs, TC Folders, Execution. TCP of Execution list overrides Test Case level TCP
- Run Automated Tests in ScratchBook

We can configure at Folder TC, TS levels.

Now here we are going to see at folder level—Select Right click and select TC.



Now Run you Test Case and view Results-



Lesson 15&16:

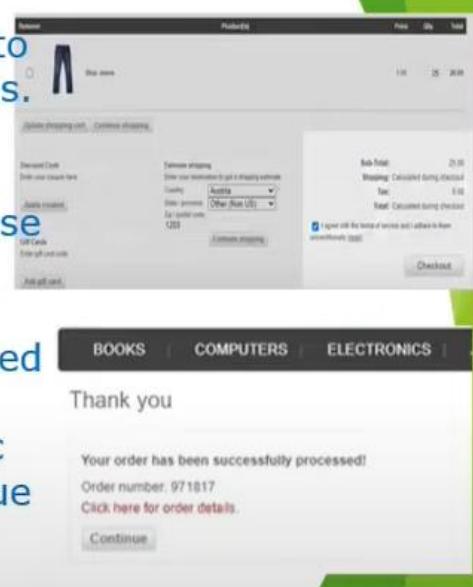
Action mode above covered Explain (Here we r not going to discuss)

Action Modes

- Action Modes tell Tosca what to do with Value to steer a particular control
- You can select them within TestCase set the column Action Mode
- Input (Default - Transfer values to test)
- Insert (we don't cover as this is non-UI)
- Buffer & Constraint (we will cover later)
- We will focus on Select, Verify and WaitOn

Action Modes

- **Select:** Passive action mode used to navigate tables & drop-down menus.
Select table you want to focus on.
But if you want to input or retrieve data from table then you need to use 'Input'
- **Verify:** Used to check values or controls. For every test case we need to have verification.
- **WaitOn:** Used to perform Dynamic waits. Used to wait for entered value or message or control to visible



Explain –Verification of Prices:

The screenshot shows the Tosca Copilot interface with the 'TestCaseDesign' tab selected. On the left, a tree view of test cases is shown, including 'AS1', 'Login Demo', 'LOGIN PageNAVi', 'Login and un n pwd', 'Zoho End to End', '11-In work', 'Session 11-Testcase Structure', 'Session 12-Testcase Structure', 'Session 14-Run Automated Tests', 'Shipping Costs', 'Precondition', 'Process' (which contains 'Order Product', 'Start checkout', 'Checkout process', 'Verification of Prices', 'Confirmation', 'Verification of Success', and 'PostCondition').

The main area displays the 'DETAILS' section of the 'CONTROL FLOW DIAGRAM'. A table lists steps with their names, values, action modes, data types, and work states. One step, 'Shipping:*', has its value set to 'Verify' and its action mode set to 'String'. The 'Value' column for this step shows a dropdown menu with 'Select' and 'Verify' options.

Name	ActionMode	ValueType	WorkState
Verification of Prices			
WebShop Confirm Order			
Products	Select	String	
Cart Total	Select	String	
<Row>	Select	String	
#2	Select	String	
Shipping:*	Verify	String	
Back	Input	String	
Confirm	Input	String	

Select is to steal the tables but not to input the values And Verify to input the values.

This screenshot is similar to the previous one, showing the Tosca Copilot interface with the 'TestCaseDesign' tab selected. The tree view of test cases is identical.

In the 'DETAILS' section of the 'CONTROL FLOW DIAGRAM', the step 'Shipping:*' now has its value set to '10.00' and its action mode set to 'Verify'. The 'Value' column for this step shows the value '10.00'.

Name	ActionMode	ValueType	WorkState
Verification of Prices			
WebShop Confirm Order			
Products	Select	String	
Cart Total	Select	String	
<Row>	Select	String	
#2	Select	String	
Shipping:*	Verify	String	
Back	Input	String	
Confirm	Input	String	

Now “Visible ==True” and Action mode “Verify” (On confirmation message verification) ---in case “Your order has been successfully processed!” message is not displayed it fails otherwise it passes.

The screenshot shows the QTP Test Case Designer interface. On the left, there's a tree view of test cases and sessions. In the center, the 'Test Case Design' tab is active, showing a 'Control Flow Diagram'. A specific step, 'Verificator: Order Success', is selected and highlighted with a red circle around its 'ActionMode' dropdown. The 'ActionMode' dropdown menu is open, showing 'Select' as the current choice. To the right, an arrow points to a screenshot of a web browser window. The browser displays a success message: 'Your order has been successfully processed!' followed by 'Order number: 1346230' and a link 'Click here for order details.'

Explain “WaitOn”

This screenshot shows the QTP Test Case Designer interface. The left pane shows a tree view of test cases and sessions. The central area shows a 'Control Flow Diagram' with a step named 'PostCondition' highlighted with a yellow circle around its 'ActionMode' dropdown. The 'ActionMode' dropdown is set to 'WaitOn'. An arrow points from this diagram to another screenshot of a web browser. The browser shows a success message: 'Your order has been successfully processed!' along with an order number and a link for details.

Lesson 17& 18:==

Action mode Buffer, Math Function, Dynamic Expressions, Dynamic Text and Numbers, Dates .

Action Mode Buffer

- Use Action Mode 'Buffer' while automating
- Important part of Testing is to Save and values that the SUT has provided you. makes verification flexible
- Instead of using hard coded data, save generated values by using buffer
- Syntax: {B[BufferName]}
- Should only be created and used in the Test Case

Dynamic Expressions

- Use Math Functions
- Generate Dynamic Text
- Generate Dynamic Random Values
- Generate Dynamic Date

Now In our project—Jeans price is 1 rupees so if I order 25 it would be $1*25=25$ so Initial price I am storing it in Buffer i.e 1 rupee and validating it.

Name	Value	ActionMode	DataType	WorkStat
Process				
Order Product				
Navigation to Apparels & shoes				
Navigation to Blue Jeans				
WebShop Blue Jeans				
Price	InnerText → PriceBlueJeans	Buffer	String	
Quantity	25	Input	String	
Add to cart	x	Input	String	
Add to wishlist		Input	String	
Start checkout				
Checkout process				
Verification of Prices				
Confirmation				
Verification of Success				

Now you can add Math functions and Buffer—Explain

Name	Value	ActionMode	DataType	WorkStat
Products		Select	String	
<Row>		Select	String	
<Cell>		Select	String	
#2		Verify	String	
Shipping:	10.00	Select	String	
Sub-total:	(MATH[[B PriceBlueJeans]] * 25)	Verify	Numeric	
Sub-Total:		Buffer	String	
Total:	MATH[B SubTotal] + 10	Verify	String	

The screenshot shows a software interface for test case design. On the left, a tree view lists various test cases and processes. The main area displays a 'DETAILS' section with a 'CONTROL FLOW DIAGRAM' table. The table has columns for Name, Value, ActionMode, DataType, and WorkState. A specific row for 'Shipping: *' is highlighted with an orange circle, showing its value as '10.00'. The table also includes rows for Sub-Total, Sub-Total, and Total, which are calculated using mathematical expressions involving 'B[PriceBlueJeans]'.

Name	ActionMode	DataType	WorkState
WebShop Confirm Order	Select	String	
Products	Select	String	
Cart Total	Select	String	
<Row>	Select	String	
<Cell>	Verify	String	
#2	Select	String	
Shipping: *	Verify	String	
Sub-Total:	Verify	Numeric	
Sub-Total:	Buffer	String	
Total:	MATH B SubTotal +10	Verify	Num...
Back	Input	String	
Confirm	Input	String	

Dynamic Text and Random Values & Dynamic Date:

As I used –COD option

In case you use Credit card option Dynamic Text, Random value, Date work—Explain

The screenshot shows a 'Test configuration' interface with a 'Details' tab. The main area is a table of test steps with columns for Name, Value, ActionMode, DataType, WorkState, and ActionPr. A specific row for 'Cardholder name' is highlighted with an orange circle, showing its value as 'RANDOMTEXT 10'. Another row for 'Expiration date | Month' is also highlighted with an orange circle, showing its value as 'DATE +4M | MM'. A callout bubble points to this cell with the text 'Right click --> Translate Value in order to check'. The table also includes rows for 'Card number', 'Expiration date | Year', 'Card code', and 'Continue'.

Name	ActionMode	DataType	WorkState	ActionPr.
Start Checkout				
Checkout Process				
Billing Address Continue				
Shipping Address Continue				
Shipping Method Ground				
Payment Method Credit Card				
Payment Information Credit Card				
Select Credit Card				
Cardholder name	RANDOMTEXT 10	Input	String	
Card number	4485564059489345	Input	String	
Expiration date Month	DATE +4M MM	Input	String	
Expiration date Year	DATE +3y YYYY	Input	String	
Card code	RND 100 999	Input	String	
Back				
Continue	X	Input	String	
Verification of Prices				
Confirmation				
Verification of Success				

Lesson 19& 20: Libraries & Reusable Test Steps Blocks

Libraries & Reusable TestStep Blocks

[Watch later](#) [Share](#)

- TestStep Library: Create Library where you can store reusable steps. The Benefit is that whenever you want to use/reuse, simply drag and drop them
- Reusable TestStep Blocks: Steps inside the Library are called Reusable TestStep Blocks. These steps can be reused throughout multiple test cases
- Libraries can contain any number of "Reusable TestStep Blocks"

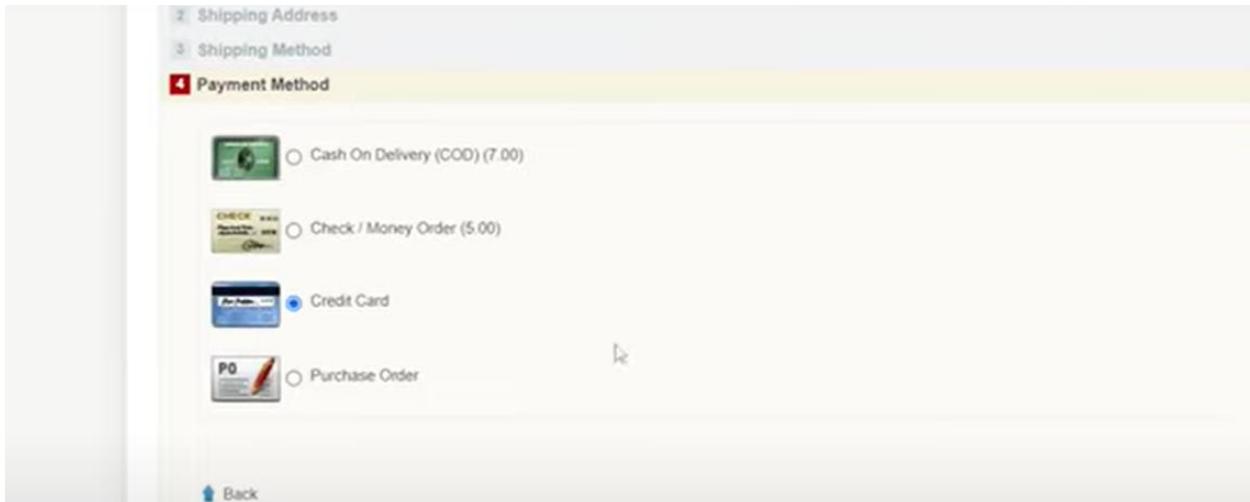
Libraries & Reusable TestStep Blocks

[Watch later](#) [Share](#)

- Create a Test Case using Reusable TestStep Blocks and automate the new test case with Reusable TestStep Blocks
- Modify a Reusable TestStep Block Reference using Resolve Reference

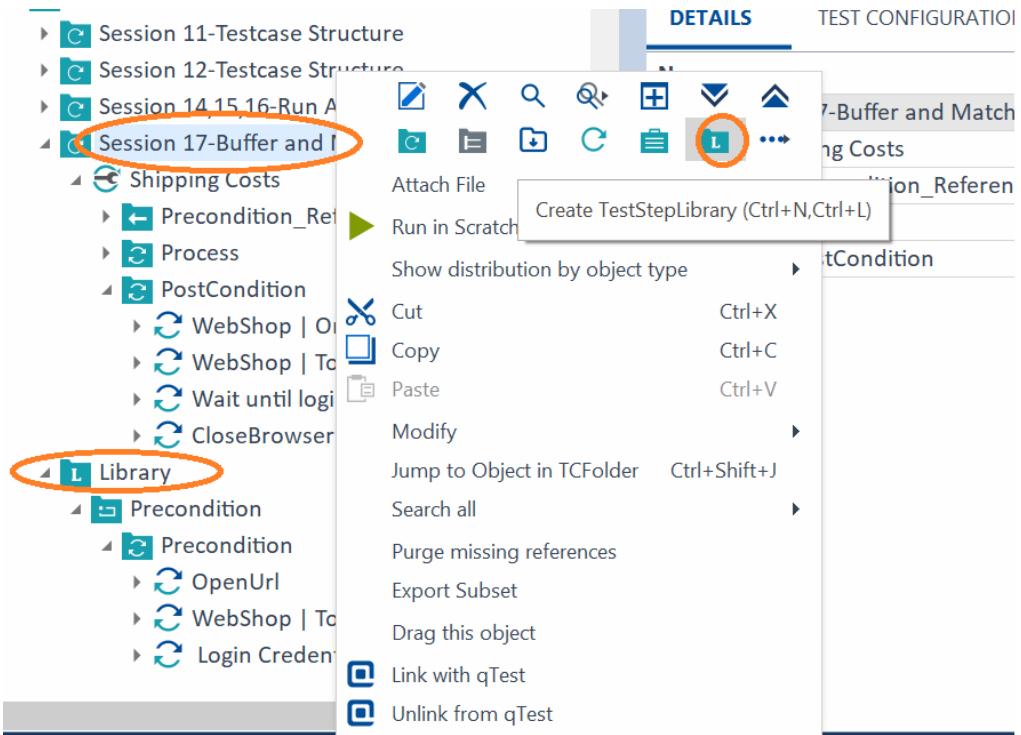
Show Example:

When Payment option is changed then we Create Library and identify reusable TS and Map those TS under library as a Reusable TS Blocks.

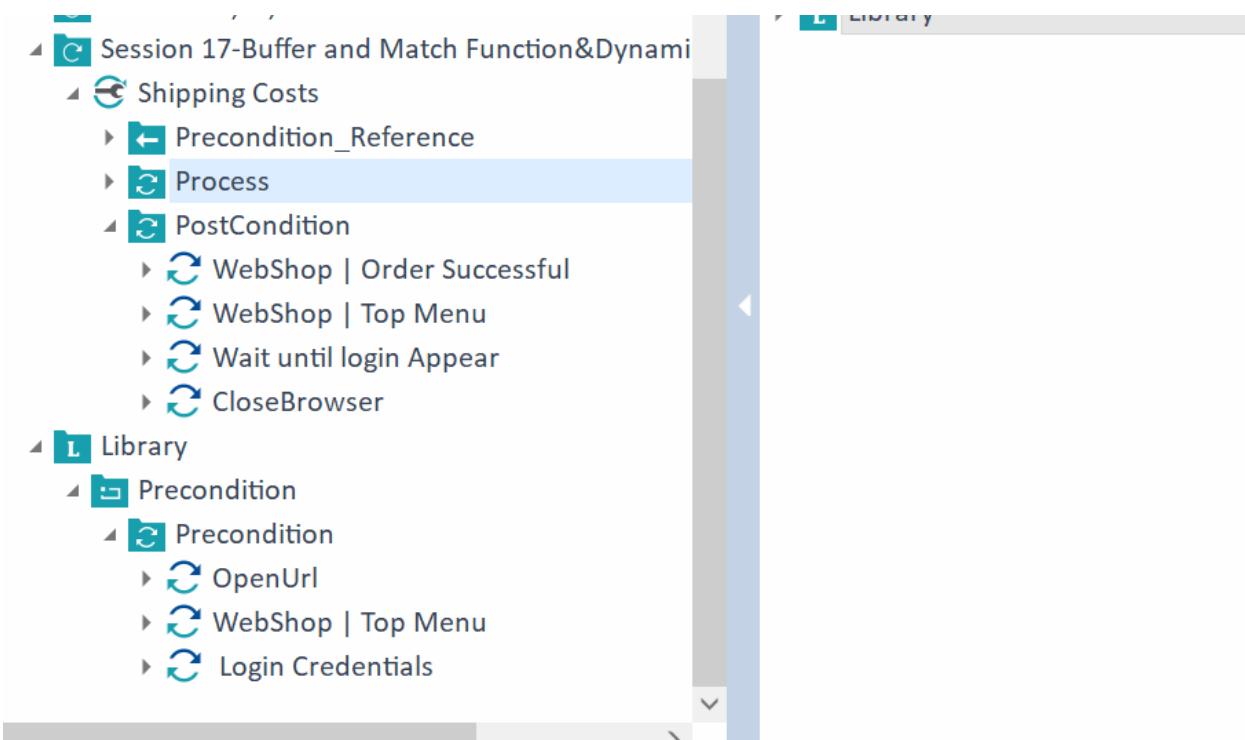


Credit card details: • Select Credit Card: Visa • Cardholder name: Barbara Gordon • Card number: 4485 5640 5948 9345 • Expiration date Month: 04 • Expiration date Year: 2024 • Card code: 123

We Create Library first:



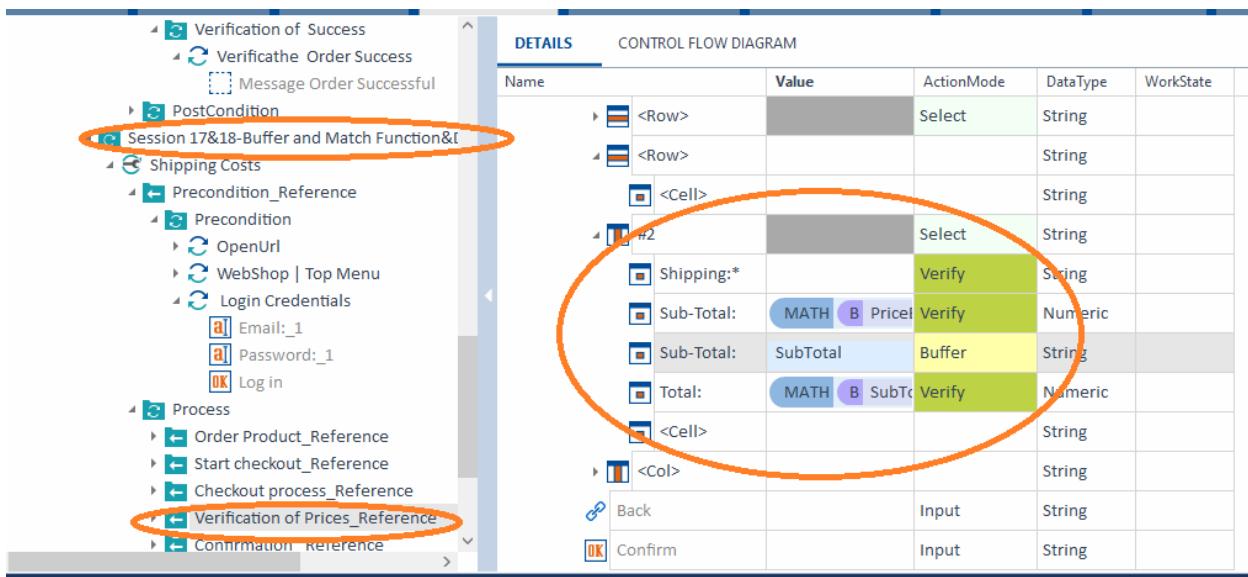
Now



So now identify and make all folders Reusable wherever they are necessary.

The screenshot shows the Tricentis Tosca interface. On the left, a navigation pane lists sessions: Session 08 - TestCase Structure, Session 09 - Test Steps Creation, Session 11 - Run Automated Tests, Session 12 - Action Modes, Session 13 - Buffer and Match Function, Session 14 - Dynamic Text Numbers Dates, Session 14 - Library and Reusable Test Steps, and ShippingCosts Money Order. The 'ShippingCosts Money Order' session is selected and expanded, revealing its structure. An orange arrow points from the 'Reusable' text in the slide to the 'Precondition_Reference' section of this expanded tree. The main workspace displays the 'Test configuration' tab of the 'ShippingCosts Money Order' details. A tooltip with the text 'New TC and you can drag and drop the necessary Ts in to it.' is overlaid on the 'Precondition_Reference' section. The table in the workspace has columns for Name, Value, ActionMode, DataType, WorkState, and ActionPr... The first row is labeled 'PLANNED'.

https://documentation.tricentis.com/tosca/1510/en/content/tutorial/modules_create.htm====Good Example see



Property=={MATH[{B[PriceBlueJeans]}]*}

Property=={{MATH[{B[SubTotal]}+10]}}

SECTION-4: Advanced Module

Lesson 21:

Rescan, Value range and Module merge

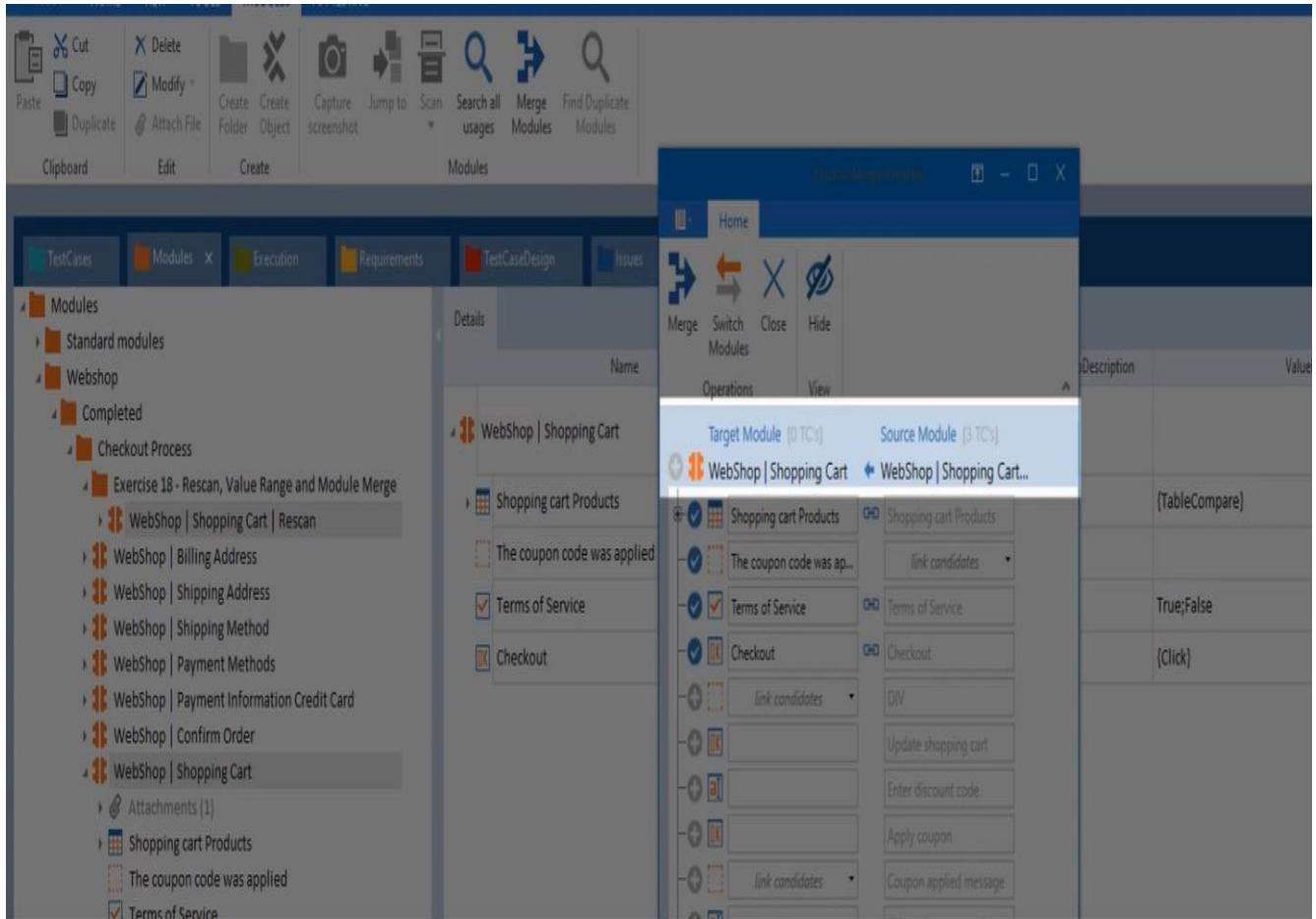
Rescan---When you want to add coupon code and Enter then you must rescan and add them in your Existing module like below:

The screenshot shows a shopping cart page with a product named "Blue Jeans". A discount code input field contains "AutomationDiscount2". An "Apply coupon" button is highlighted with a red oval. Below it, a message "The coupon code was applied" is also highlighted with a red oval. To the right, a sidebar lists items with a checkbox for "discountcouponcode". The "The coupon code was applied" item is checked and highlighted with a red oval.

Now Value range option you can enter more than one:

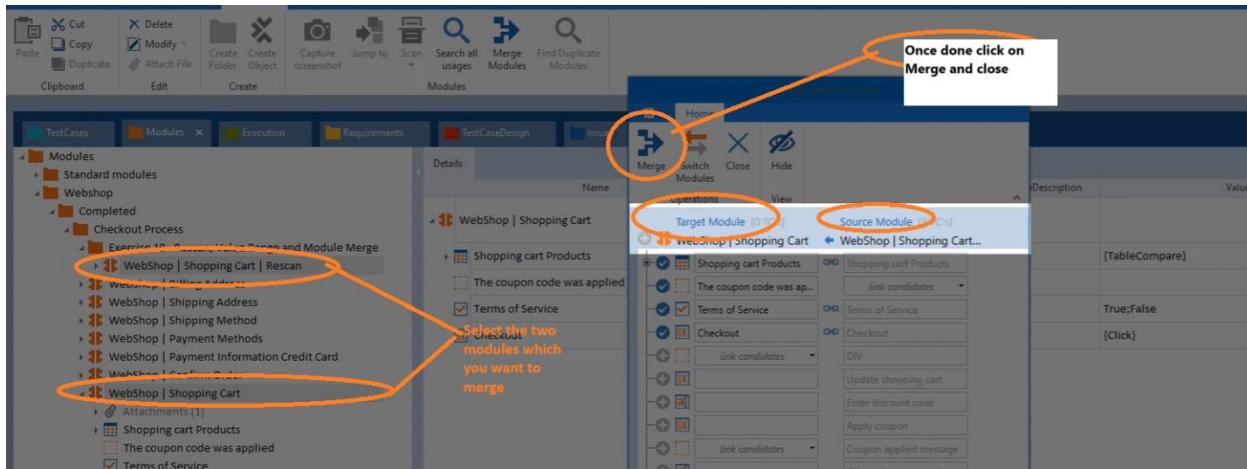
The screenshot shows a test planning interface with a "ValueRange" column. One row for "Enter Discount code" has a value of "[k];[DoubleClick];[Rightclick];AutomationDiscount2;Per...". A callout bubble points to this value with the text "Enter all the possible things-- say Right click,double click etc".

Note when you rescan –it create a Duplicate of previous module in order to avoid it we use module merge.



Explain in Detail:--In general when we need to merge a very huge module we import as we are not aware of all of them.

Usually Tosca Architect might do it.But you need to be aware of it.

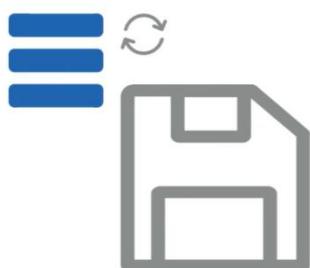


Section-5:

Lesson 22:[For all this only Theory---as a part of EX –practice it says]

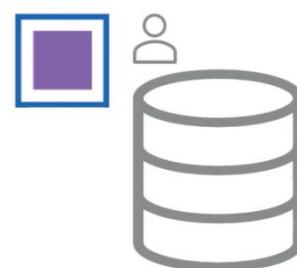
TCP---Earlier we used it for Browser say Chrome---Under Test Configuration

Note The way it stores in Buffer and TCP:



Buffer

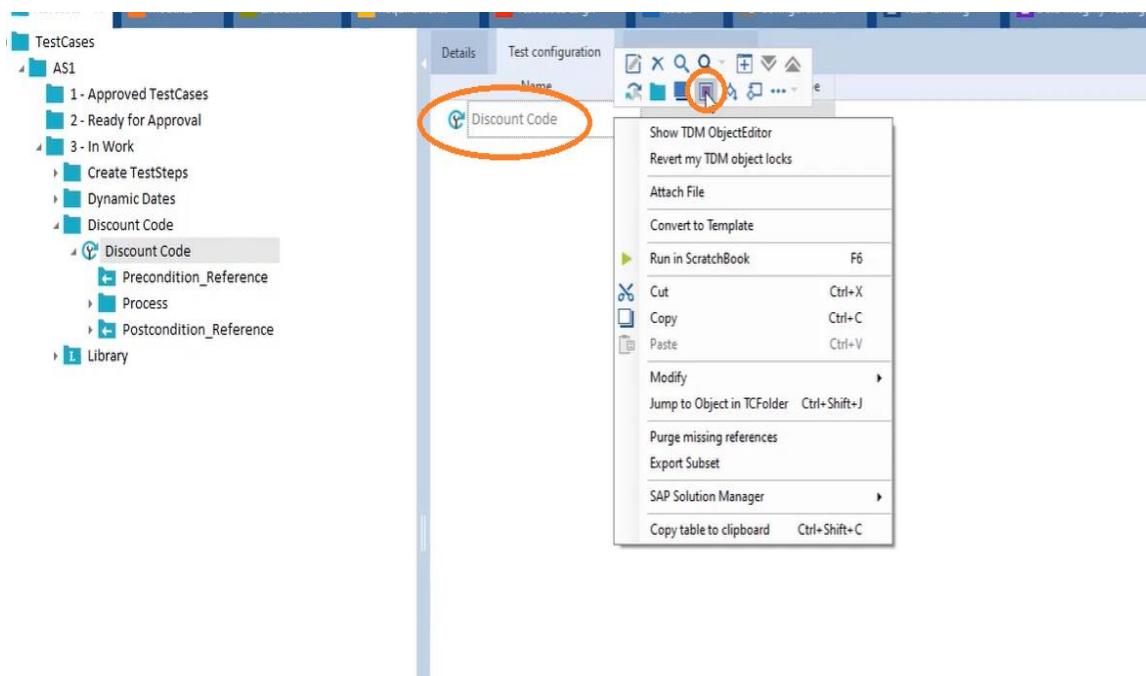
Generated during run time



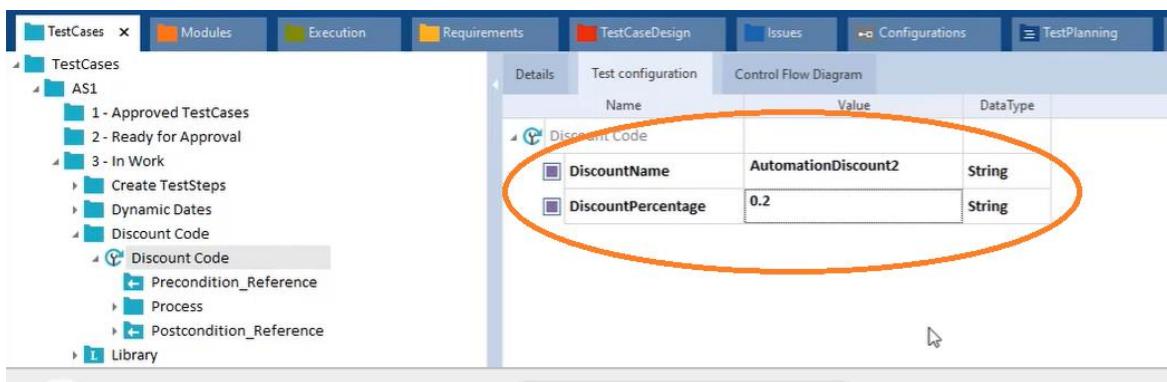
TCP

Generated by a tester

Now we will learn how to use Self Defined TCP (Test configuration Parameter)---you can define it:



Discount may be 20% then we can add it as parameter----we
create it at TC level



Now you can use of this TCP as reference whenever they are needed:

Previously we used this---

The screenshot shows a software interface with two main sections. On the left is a tree view of test cases under 'AS1':

- AS1
 - 1 - Approved TestCases
 - 2 - Ready for Approval
 - 3 - In Work
 - Create TestSteps
 - Dynamic Dates
 - Discount Code
 - Discount Code
 - Precondition_Reference
 - Process
 - Postcondition_Reference
 - Library

Control Flow Diagram		Name	Value	ActionMode	DataType	WorkState
Process						
	Order Product_Reference					
	Start Checkout_Reference					
	Start Checkout					
		Navigate to Shopping Cart				
		Shopping Cart Procedures				
		Enter discount code	AutomationDiscount12	Input	String	
		Apply coupon	x	Input	String	
		Coupon applied message	Visible == True	Select	String	
		Terms of Service	True	Input	String	
		Checkout	x	Input	String	
		Checkout Process_Reference				
		Verification of Success_Reference				
		Confirmation_Reference				
		Verification of Prices_Reference				

Now you can change it to:

Advantage of using TCP---lowest maintains and in the future when discount % changes just we need to change it in TCP

that it.

- AS1
 - 1 - Approved TestCases
 - 2 - Ready for Approval
 - 3 - In Work
 - Create TestSteps
 - Dynamic Dates
 - Discount Code
 - Discount Code
 - Precondition_Reference
 - Process
 - Postcondition_Reference
 - Library

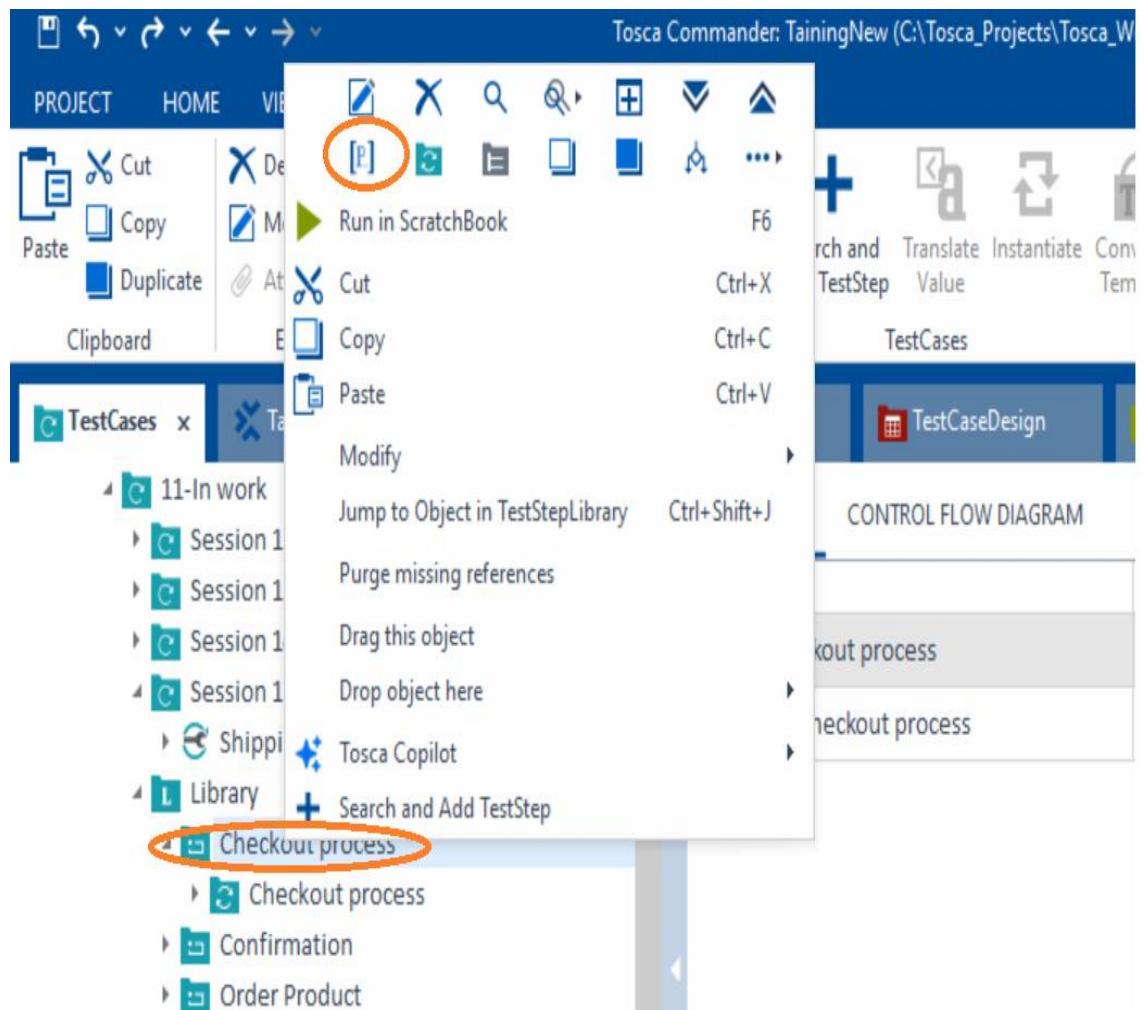
Name	Value	ActionMode	DataType	WorkState
Process				
Order Product_Reference				
Start Checkout_Reference				
Start Checkout				
Navigate to Shopping Cart				
Shopping Cart Procedures				
Enter discount code	(CP[DiscountName])	Input	String	
Apply coupon	x	Input	String	
Coupon applied message	Visible == True	Select	String	
Terms of Service	True	Input	String	
Checkout	x	Input	String	
Checkout Process_Reference				
Verification of Success_Reference				
Confirmation_Reference				
Verification of Prices_Reference				

Lesson 23:

Business Parameters:

Business Parameters are used at Test case level

For our Library---Reusable Test Step Block right click and select
Business Parameter container



We get like this:

The screenshot shows a software interface with a left sidebar containing a tree view of test cases and a right panel titled "DETAILS".

- Tree View:**
 - Session 11-Testcase Structure
 - Session 12-Testcase Structure
 - Session 14,15,16-Run Automated Tests
 - Session 17&18-Buffer and Match Function&C
 - Shipping Costs
 - Library
 - Checkout process
 - Business Parameters
 - Checkout process
 - Billing Address Continue
 - Shipping Address Continue
 - Shipping Method Ground
 - Payment Methods Credit card
 - Confirmation
 - Order Product
 - PostCondition
 - Precondition
 - Precondition
 - OpenUrl

DETAILS Table:

Name	Value	ActionMode	DataType	WorkState
[B] Business Parameters				

Now right click on Business Parameters in order to create Parameters:

The screenshot shows the same software interface as the first one, but with a context menu open over the "Business Parameters" row in the "DETAILS" table.

Context Menu Options:

- Cut (highlighted with a red circle)
- Copy
- Paste
- Jump to Object in TestStepLibrary Ctrl+Shift+J
- Purge missing references
- Drag this object
- Drop object here
- Tosca Copilot
- Copy table to clipboard

Or

Screenshot of a software interface showing a tree view of test steps on the left and a 'DETAILS' tab with a 'CONTROL FLOW DIAGRAM' on the right.

Tree View (Left):

- Shipping Address Continue
- Shipping Method Ground
- Payment Methods Credit card
- Confirmation
- Order Product
- PostCondition
- Precondition
- Business Parameters
- Precondition
- OpenUrl
- WebShop | Top Menu
- Login Credentials
- Start checkout
- Verification of Success
- Verification of Prices
- Session- 19&20-Library and Reusable TestStep
- Shipping Costs with TCP
- Precondition
- Process

DETAILS Tab (Right):

Name	Value	ActionMode	DataType	WorkState
Precondition				
Business Parameters				
BURL		Input		
Precondition				
OpenUrl				
Url	{PL[BURL]}	Input	String	
UseActiveTab		Input	String	
WebShop Top Menu				
Login Credentials				

Then we get below---how many we want that many we can create parameter.

Now rename the parameter to URL, Username, Password etc as per your requirement as shown below:

Screenshot of a software interface showing a tree view of test steps on the left and a 'DETAILS' tab with a 'CONTROL FLOW DIAGRAM' on the right. The 'Business Parameters' node is circled in orange. A red line connects the 'BURL' parameter to a 'UR Choice' input. Another red line connects the '{PL[BURL]} placeholder in the 'Url' field to a 'PL-Parameter Link'. A note 'Drag and Drop this' points to the placeholder.

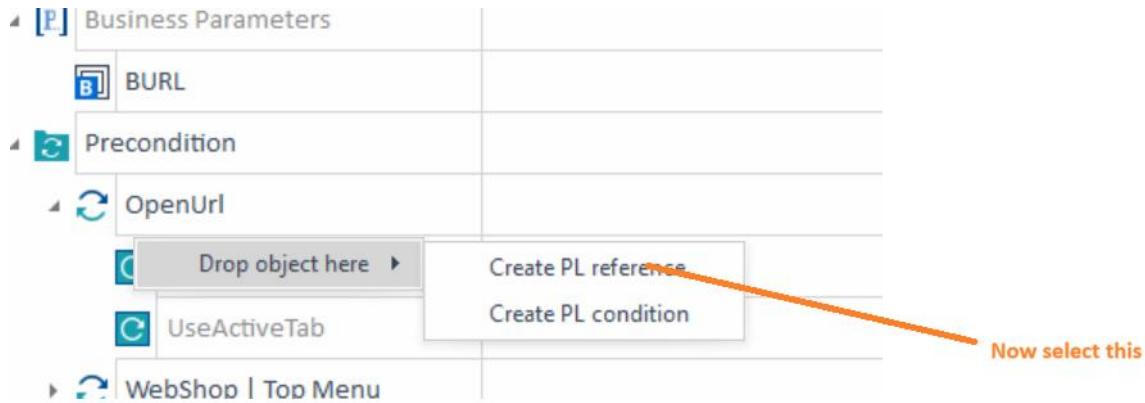
Tree View (Left):

- Shipping Address Continue
- Shipping Method Ground
- Payment Methods Credit card
- Confirmation
- Order Product
- PostCondition
- Precondition
- Business Parameters
- Precondition
- OpenUrl
- WebShop | Top Menu
- Login Credentials
- Start checkout
- Verification of Success
- Verification of Prices
- Session- 19&20-Library and Reusable TestStep
- Shipping Costs with TCP

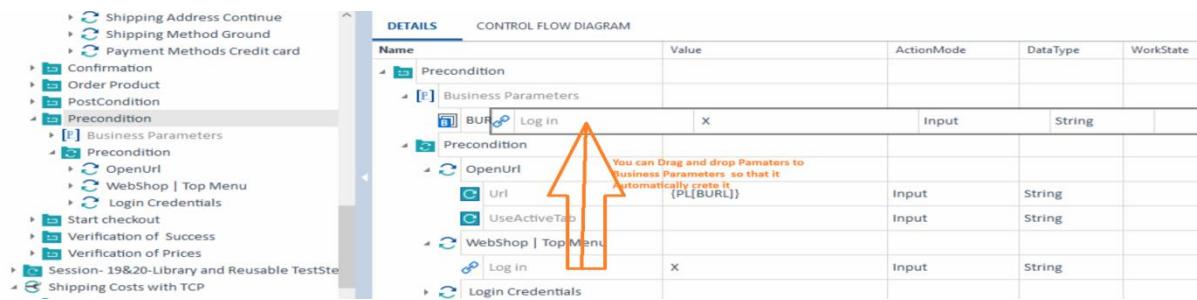
DETAILS Tab (Right):

Name	Value	ActionMode	DataType	WorkState
Precondition				
Business Parameters				
BURL		Input		
Precondition				
OpenUrl				
Url	{PL[BURL]}	Input	String	
UseActiveTab		Input	String	
WebShop Top Menu				
Login Credentials				

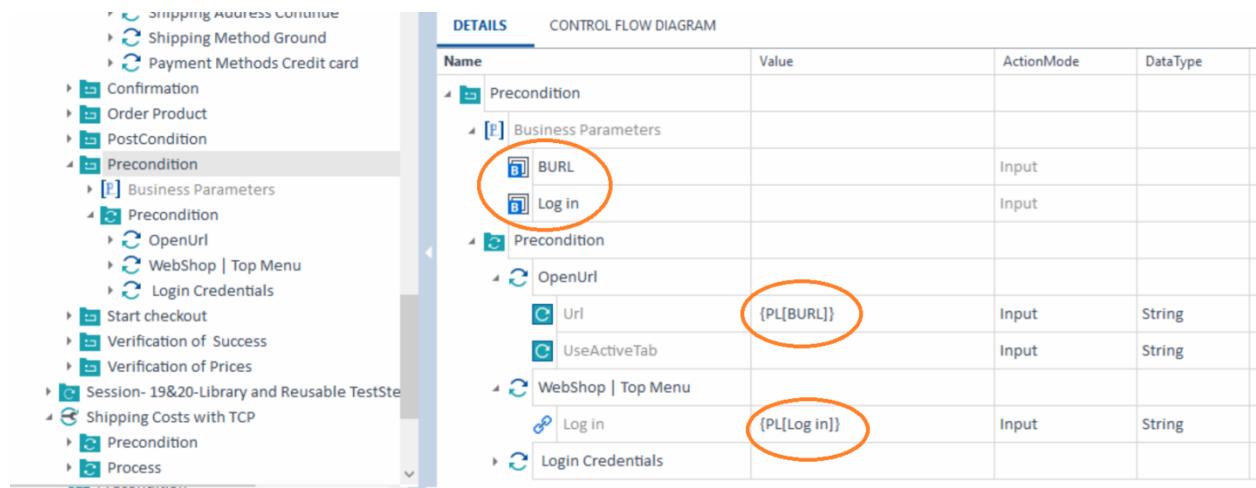
When you Drag and drop it ask



Now In case we already have Parameters then we can drag and drop in Business parameters

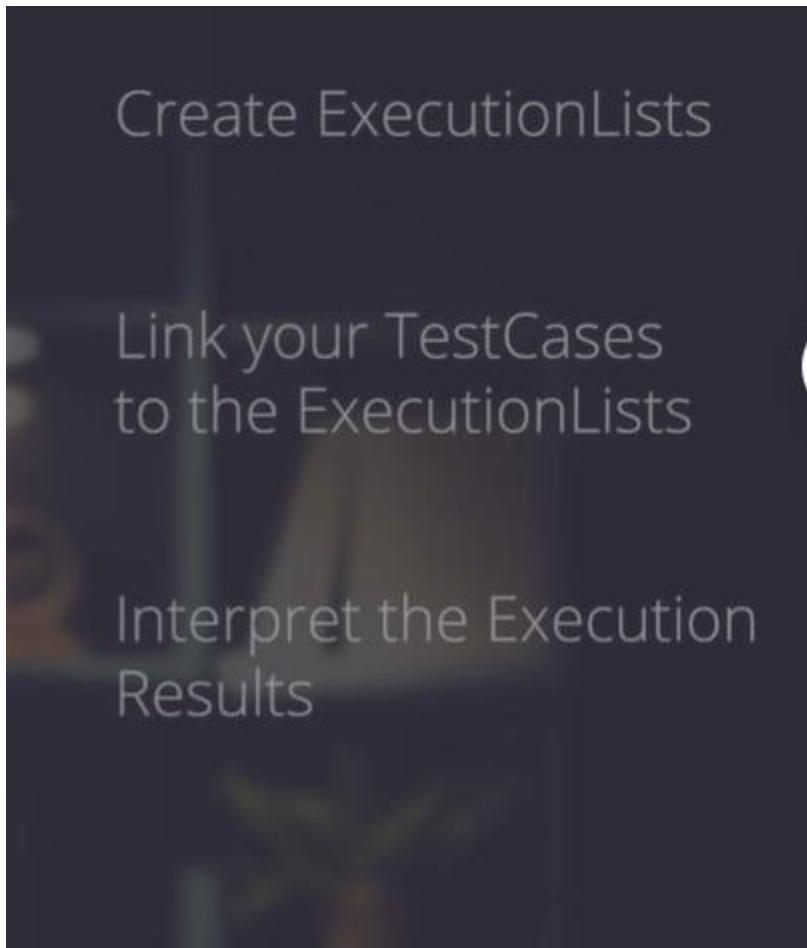


Like this: as Name it only gives automatically:

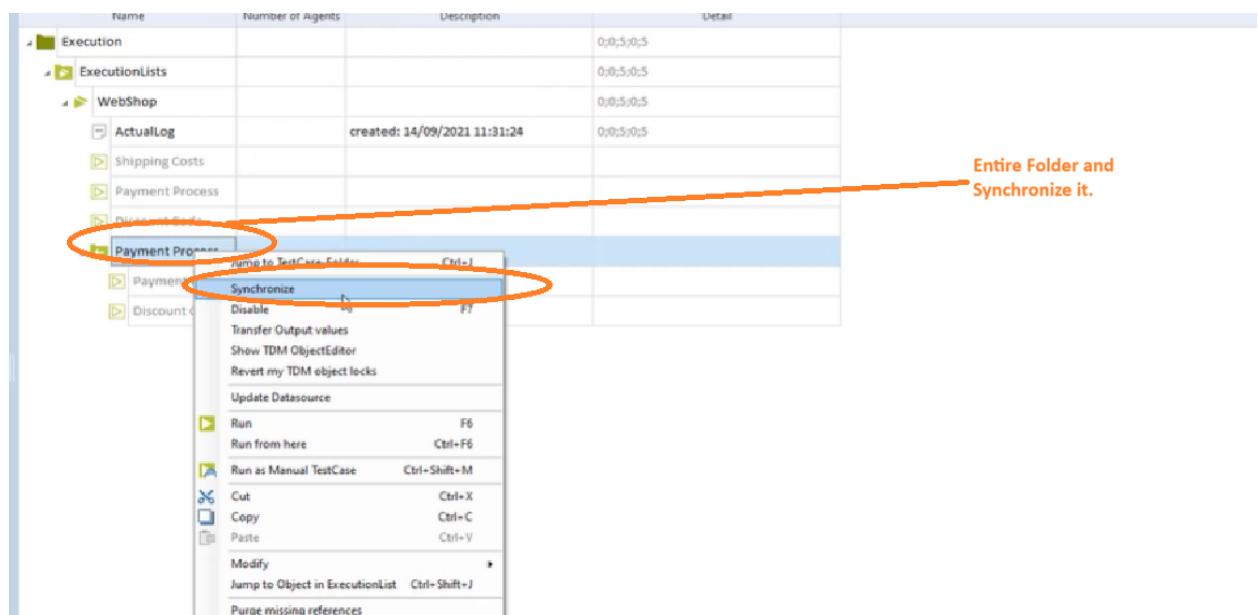
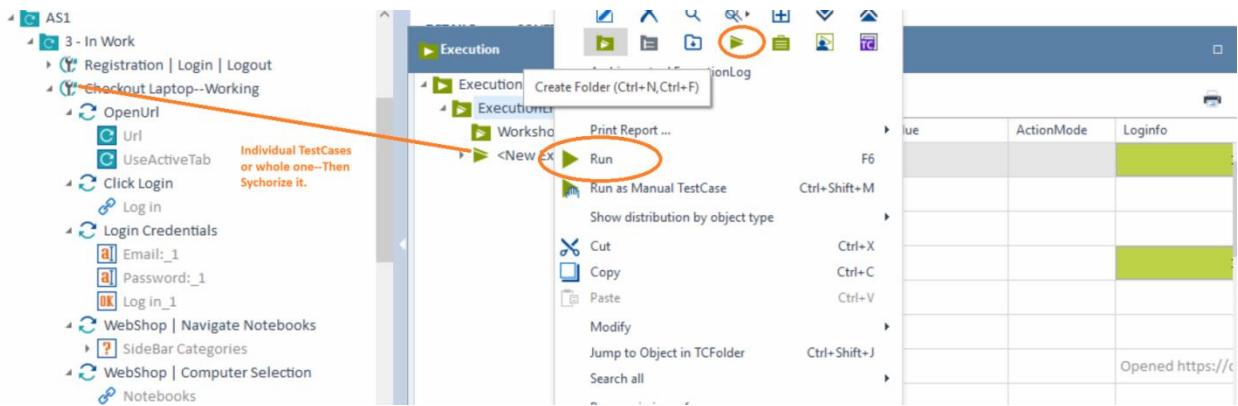


Section 6:--

Lesson 23: Execution list: Now no more we run Test cases in the Scratch Book but in Execution



Go to Execution Tab: Create a Execution list and a folder then dragn and drop all the TC which you want to run or One Entire TC/Folder and right click and Synchronize it and then Run.



In order to run more than once use Repetitions for more than once and see—under properties

Name	Value	ActionMode	Loginfo
Checkout Laptop--Working			
30.11.24 00:36:31			

Name	Value
NodePath	/Execution/Exec...
CreatedBy	Admin
CreatedAt	11/30/2024 12:3...
ModifiedBy	Unknown
ModifiedAt	Unknown
HasMissingReferences	False
UniqueId	3a168b7d-ed1c...
DisabledDescription	
Repetitions	1

EXECUTION SHORTCUTS

Run



Run from here

CTRL + F6

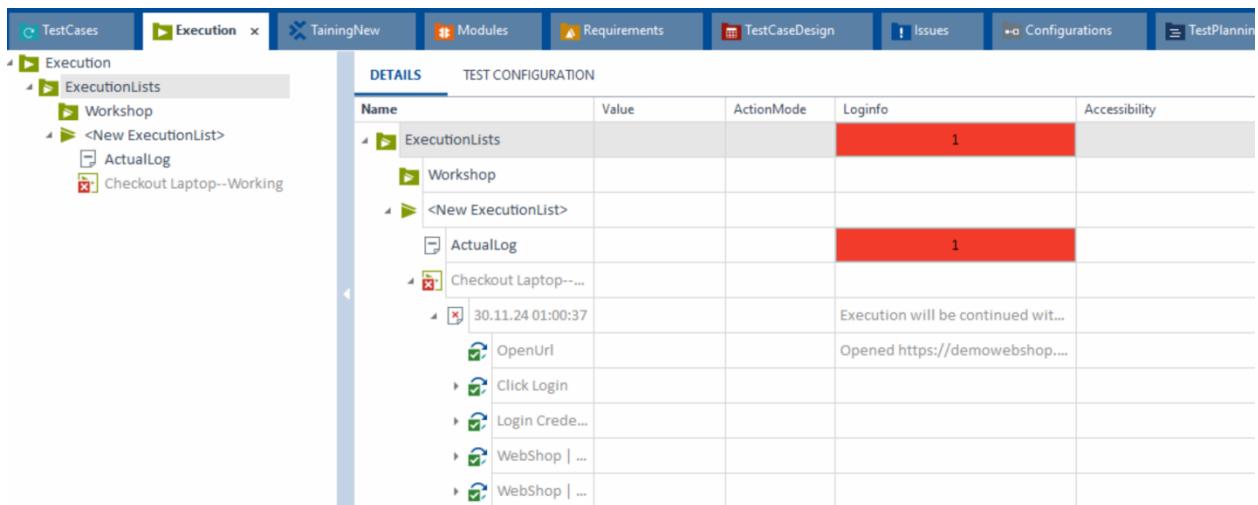
Run as Manual TestCase

CTRL + SHIFT + M

You can write Test configuration at Execution level too but it will overwrite the previous one—SAY on TC.

Name	Value	DataType
Checkout Laptop--Working		
Browser	Chrome	String

This way you can see the results:



The screenshot shows a software interface with a navigation bar at the top containing links like TestCases, Execution, TainingNew, Modules, Requirements, TestCaseDesign, Issues, Configurations, and TestPlanning. The 'Execution' tab is selected. On the left, there's a tree view under 'Execution' showing 'ExecutionLists', 'Workshop', and a new entry '' which includes 'ActualLog' and 'Checkout Laptop--Working'. The main area is titled 'DETAILS TEST CONFIGURATION' and displays a table with columns: Name, Value, ActionMode, LoginInfo, and Accessibility. The table shows the configuration for the 'Checkout Laptop--Working' execution list, including its sub-items and their actions.

Name	Value	ActionMode	LoginInfo	Accessibility
ExecutionLists			1	
Workshop				
<New ExecutionList>			1	
ActualLog				
Checkout Laptop--...				
30.11.24 01:00:37			Execution will be continued wit...	
OpenUrl			Opened https://demowebshop....	
Click Login				
Login Crede...				
WebShop ...				
WebShop ...				

Section7:--

Lesson 24: Requirement, weight Requirement, link requirement and Link Test cases and Execution list to the Requirement:

Requirement:



Create Requirements, Requirements Sets, and Requirement Folders

Right click on Requirement section Create Requirement Folders and SETS.

The screenshot shows a software application window with a dark header bar containing various tabs: TestCases, Execution, TrainingView, Modules, Requirements (highlighted), TestCaseDesign, Issues, Configurations, and TestPlanning. On the left, there is a tree view under the Requirements tab. The tree starts with 'Requirements' which has several collapsed branches like 'Requirements_import241128_2114', 'Virtual folders', and 'Requirements_import241129_1311'. One branch, 'Webshop Requirements', is expanded, showing 'Webshop Frontend' with further sub-items: 'Customer tasks', 'Handle products', 'Shopping cart', and 'Order process'. The 'Order process' item is selected and highlighted with a blue background. On the right, there is a 'DETAILS' pane with a table titled 'Name' and 'Description'. The table includes columns for 'Freq...', 'Dam...', 'Weight', and 'Contribution (%)'. It lists several requirements, each with a small orange icon and a plus sign. The first requirement is 'Order process' with a detailed list of sub-tasks: 'Execute checkout', 'Billing and shipping a...', 'Calculate shipping costs', 'Payment methods', and 'Reorder'. The 'Execute checkout' item is also highlighted with a blue background.

Name	Description	Freq...	Dam...	Weight	Contribution (%)
Order process		5	5	1024	
Execute checkout		3	5	256	
Billing and shipping a...		2	5	128	
Calculate shipping costs		5	5	1024	
Payment methods		5	5	1024	
Reorder		2	4	64	

Link TestCases → Requirement and Execution→Requirement and show:

The screenshot shows a software interface with multiple tabs at the top: TestCases, Modules, TestCaseDesign, Execution, Requirements, and Baseline. The Requirements tab is currently active.

TestCases:

- AS1
 - 1 - Approved TestCases
 - 2 - Ready for Approval
 - 3 - In Work
 - Exercise 05 - Create TestCase Structure
 - Exercise 06 - Open and Close the Application
 - Exercise 07 - Create TestSteps
 - Exercise 08 - TestStep Values
 - Exercise 09 - Predefined Test Configuration
 - Exercise 10 - ActionMode WaitOn
 - Exercise 11 - ActionMode Verify & WaitOn
 - Exercise 12 - Buffering
 - Exercise 13 - MATH Functions
 - Exercise 14 - Dynamic Text
 - Exercise 15 - Dynamic Dates
 - Library
 - Exercise 17 - Create a TestCase from a Library
 - Exercise 19 - TCP
 - Exercise 20 - Business Parameters
 - Discount Code

Requirements:

Name	Description	Frequ...	Dama...	Weight	Contribution (%)
Customer tasks		4	4	256	10.81
Register		3	4	128	1.97
Login		5	4	512	7.86
Modify customer data		3	2	32	0.49
Check orders		2	3	32	0.49
Handle products		3	3	64	2.7
Product configuration		3	5	256	1.78
Modify products view		4	2	64	0.45
Compare products		1	1	4	0.03
Search for products		4	2	64	0.45
Shopping cart		5	5	1024	43.24
Add products		3	5	256	17.3
Gift cards		3	4	128	8.65
Discounts		3	4	128	8.65
Discount Code				1	0.05
Manage shopping cart		3	4	128	8.65
Order process		5	5	1024	43.24
Execute checkout		3	5	256	4.44
Billing and shipping address				2	0.10
Calculate shipping costs				5	0.50
Shipping Costs				5	0.50
Payment methods				5	0.50

Now see the link by going back and explain them:

The Requirements module is now active, showing a detailed list of requirements under the Webshop Frontend category. Each requirement has its name, description, frequency, damage, weight, and contribution percentage.

Name	Description	Frequ...	Dama...	Weight	Contribution (%)	Coverage Specified (%)
Web Shop				1		44 56
Customer tasks				4 4	256	10.81 100
Register				3 4	128	1.97 100
Login				5 4	512	7.86 100
Modify customer data				3 2	32	0.49 100
Check orders				2 3	32	0.49 100
Handle products				3 3	64	2.7 100
Product configuration				3 5	256	1.78 100
Modify products view				4 2	64	0.45 100
Compare products				1 1	4	0.03 100
Search for products				4 2	64	0.45 100
Shopping cart				5 5	1024	43.24 20 80
Add products				3 5	256	17.3 100
Gift cards				3 4	128	8.65 100
Discounts				3 4	128	8.65 100
Discount Code					1	0.05
Manage shopping cart				3 4	128	8.65 100
Order process				5 5	1024	43.24 82 100
Execute checkout				3 5	256	4.44 100
Billing and shipping address				2 5	1024	43.24 82 100
Calculate shipping costs				5 5	1024	43.24 82 100
Shipping Costs				5 5	1024	43.24 82 100
Payment methods				5 5	1024	43.24 82 100

Section8:

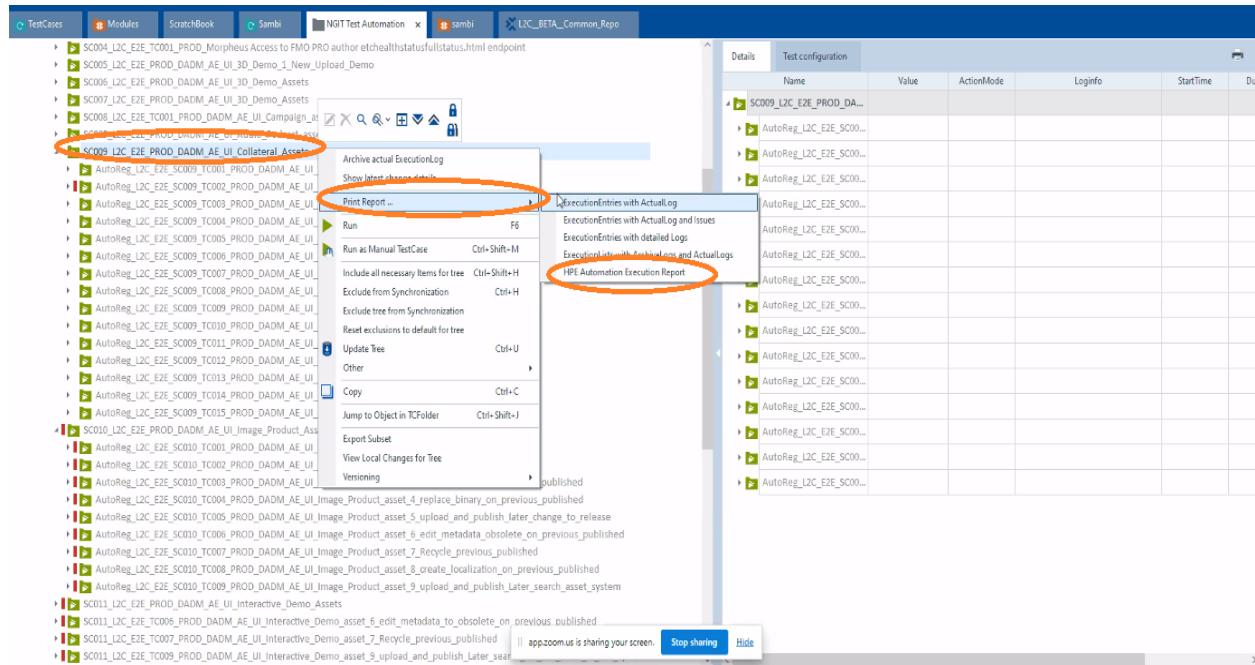
Lesson 25: Additional Advance Topics:

Multiuser Environment---Already discussed----Explain once again in detail...It must be connected to the DB.

The screenshot shows the DEX Workspace interface. On the left, there's a navigation tree with items like DEX Workspace, Configurations, Execution, Issues, Modules, Reporting, Requirements, TestCaseDesign, TestCases, Standard module examples, Virtual Folder, sample test cases, Set Banner, Evaluate, Set Now, and NewTestCase. The 'Modules' item is circled in orange. The main area has tabs for Details, Test configuration, and Control Flow Diagram. The 'Test configuration' tab is selected, showing a table with columns Name, Value, ActionMode, DataType, and WorkState. A single row is present with 'Name' as 'NewTestCase', 'Value' as an empty field, 'ActionMode' as 'PLANNED', and 'WorkState' as 'PLANNED'. Three orange arrows point from the circled items in the tree to text annotations on the right: one arrow points from 'Modules' to the text 'Check out--Locked--so others can not use it.', another from 'sample test cases' to 'Check in All--now others can use and make necessary changes', and a third from 'NewTestCase' to 'TC is not yet assigned to Tosca PRJ'.

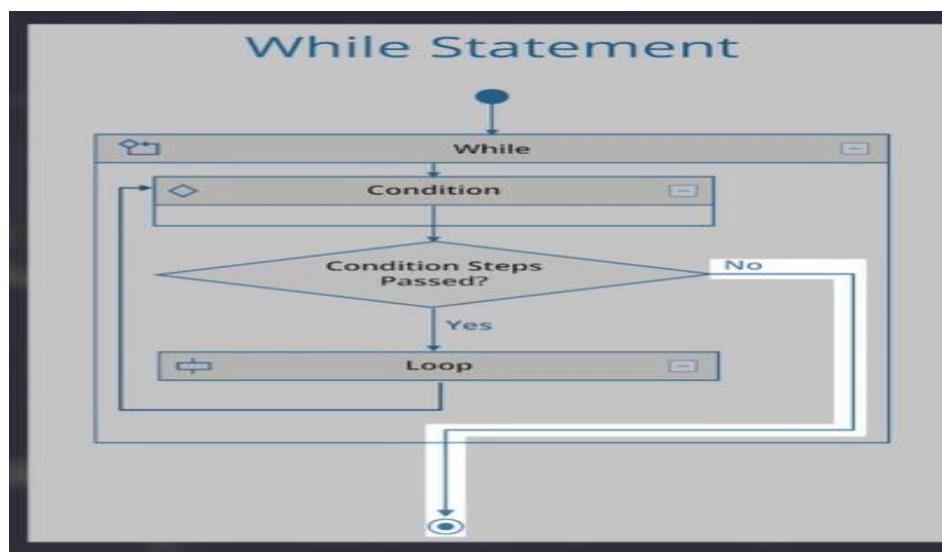
Name	Value	ActionMode	DataType	WorkState
NewTestCase		PLANNED		PLANNED

In real time we have this and save it in Desktop/Specific folder and view it—Hp project.

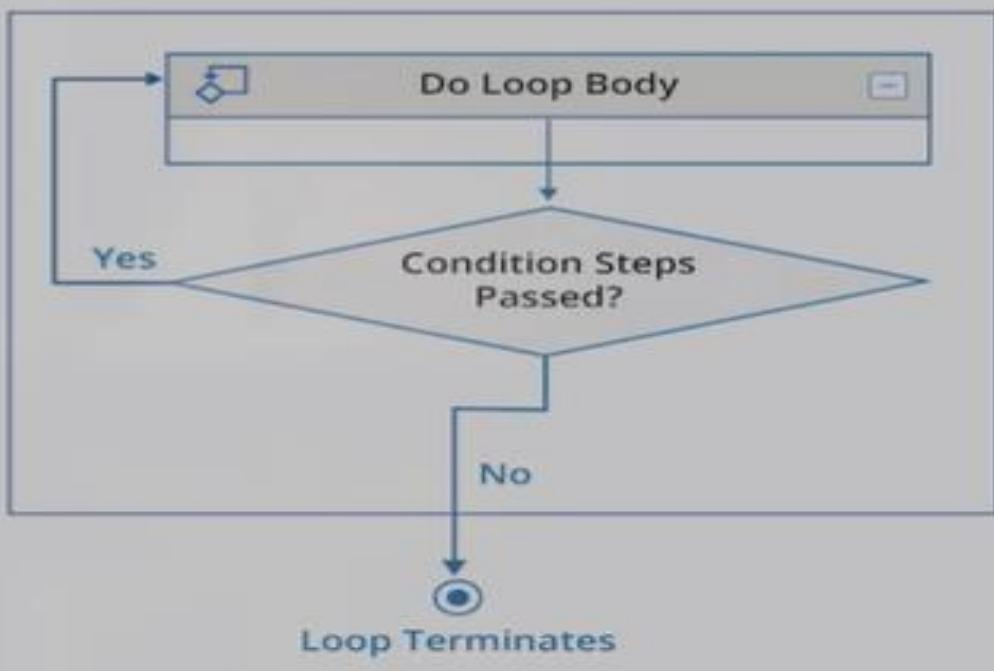


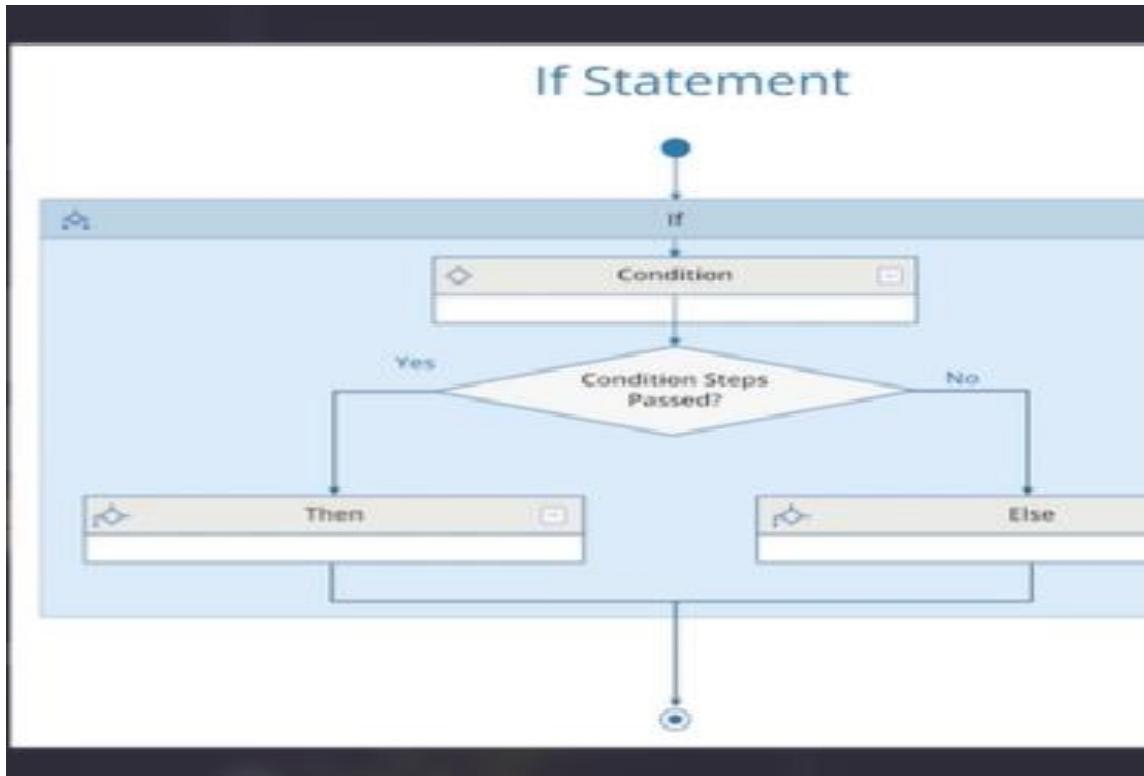
LOOPS:--While, Do and IF only

While condition:



Do Statement

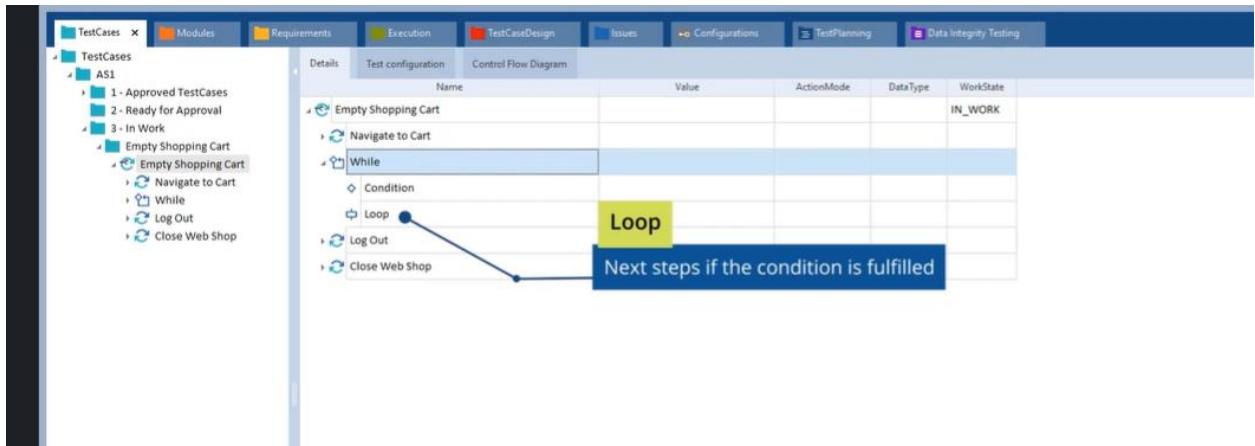




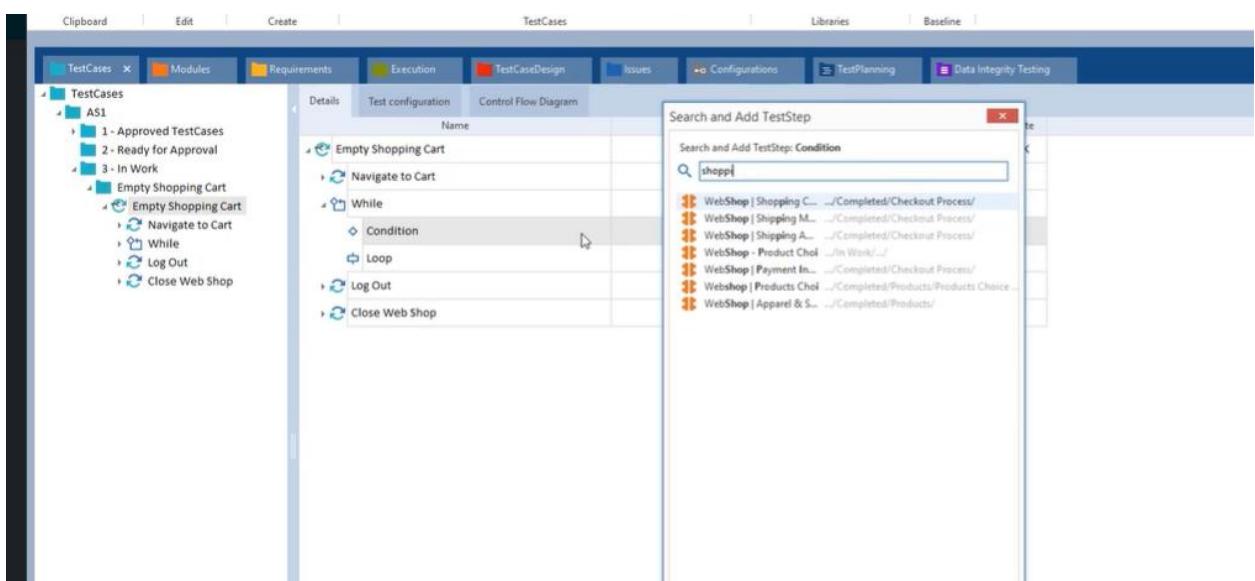
For Example:

The screenshot shows a software interface for managing test cases. The left sidebar lists test cases under 'TestCases' and 'AS1'. The main area has tabs for 'Details', 'Test configuration', and 'Control Flow Diagram'. The 'Control Flow Diagram' tab is active, displaying a table with columns for Name, Value, ActionMode, DataType, and WorkState. A row for a 'While' loop is selected, showing its internal steps: 'Condition', 'Loop', 'Log Out', and 'Close Web Shop'. A yellow callout box labeled 'Condition' points to the 'Condition' step in the loop. A blue callout box labeled 'Condition to be met for the step to be completed' points to the 'Condition' step in the loop.

Name	Value	ActionMode	DataType	WorkState
Empty Shopping Cart				IN_WORK
Navigate to Cart				
While				
Condition				
Loop				
Log Out				
Close Web Shop				



Now you can add modules –under while



Under loop—we are checking cart is Empty or not—using If loop

stCases

A51

- 1 - Approved TestCases
- 2 - Ready for Approval
- 3 - In Work
- Empty Shopping Cart**
- Empty Shopping Cart
- Navigate to Cart
- While
- Condition
- Verify Cart Exists
- Loop
- WebShop | Shopping Cart
- Shopping cart Products
- The coupon code was applied
- Terms of Service
- Checkout
- Update shopping cart
- Enter discount code
- Apply coupon
- Coupon applied message
- Gift card coupon code
- Add gift card
- Log Out
- Close Web Shop

IN_WORK

Change name to Empty cart

Then you can make cart empty

pproval

ping Cart

opping Cart

ate to Cart

ndition

op

ut

Web Shop

- Empty Shopping Cart
- Navigate to Cart
- While
- Condition
- Loop
- Empty Cart
- Shopping cart Products
- \$1
- Remove
- Remove
- <Cell>
- Remove
- <Row>
- <Cell>
- <Col>
- The coupon code was applied
- Terms of Service
- Checkout
- Update shopping cart

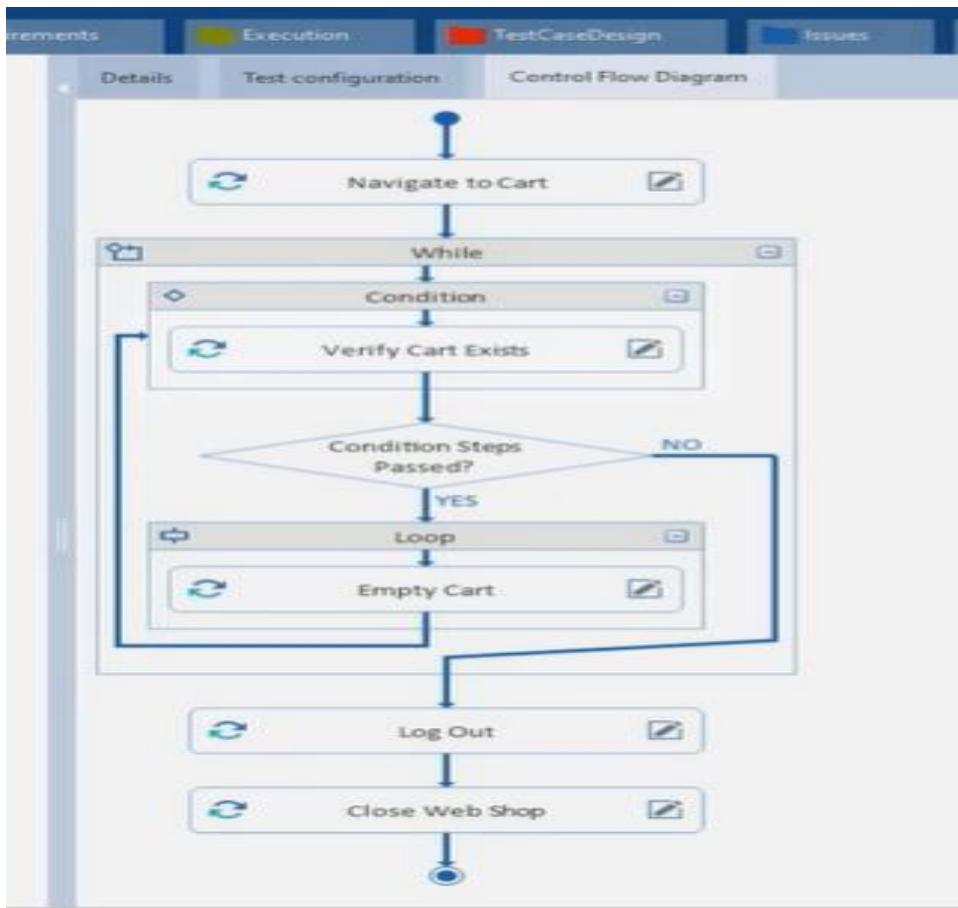
IN_WORK

Properties

Name	Value
NodePath	/TestCases/AS1/...
CreatedBy	Admin
CreatedAt	25/11/2023 12:4...
ModifiedBy	Unknown
ModifiedAt	Unknown
HasMissingReferences	False
Unsaved	True
DisabledDescription	3a008915-9f78-4...
MaximumRepetitions	30

How many time you want to repeat this--say 30 times

Show them how control Flow Diagram ---how Tosca follows



Lesson 30: Recovery Scenarios



Good to know

Recovery Scenarios kick in when the TestCase is executed from the ExecutionList

The screenshot shows a software interface for managing test cases. On the left, a tree view lists "TestCases" under "AS1", with nodes for "Approved TestCases", "Ready for Approval", "In Work", "Empty ShoppingCart", and "Shipping Costs". The "In Work" node is expanded, showing three sub-nodes: "*** Recovery Scenarios ***", "Recovery Scenario", and "Condition_Reference". The "Recovery Scenario" node is highlighted with a red circle. A callout arrow points from this node to a "Good to know" box containing the text: "Recovery Scenarios kick in when the TestCase is executed from the ExecutionList". On the right, a "Properties" panel displays details for the "Recovery Scenario" node. A second red circle highlights the "RetryLevel" property, which is set to "TestCase".

Name	Value	ActionMode	DataType	WorkState
*** Recovery Scenarios ***				
Recovery Scenario				COMPLETED
Condition_Reference				
Order Product_Reference				
Start Checkout_Reference				
Checkout Process_Reference				
Verification of Prices_Reference				
Confirmation_Reference				
Verification of Success_Reference				
Postcondition_Reference				

Name	Value
Recovery Scenario	[TestCases/AS1/...]
NodePath	
CreatedBy	Admin
CreatedAt	26/11/2023 11:5...
ModifiedBy	Unknown
ModifiedAt	Unknown
HasMissingReferences	False
UniqueId	4e09611-13ce-...
EnabledDescription	
RetryLevel	TestCase

Now in project we need to Enable—Recovery first

Go to Project→Settings→Tbox→Recovery (see as per your project they r set in real time)

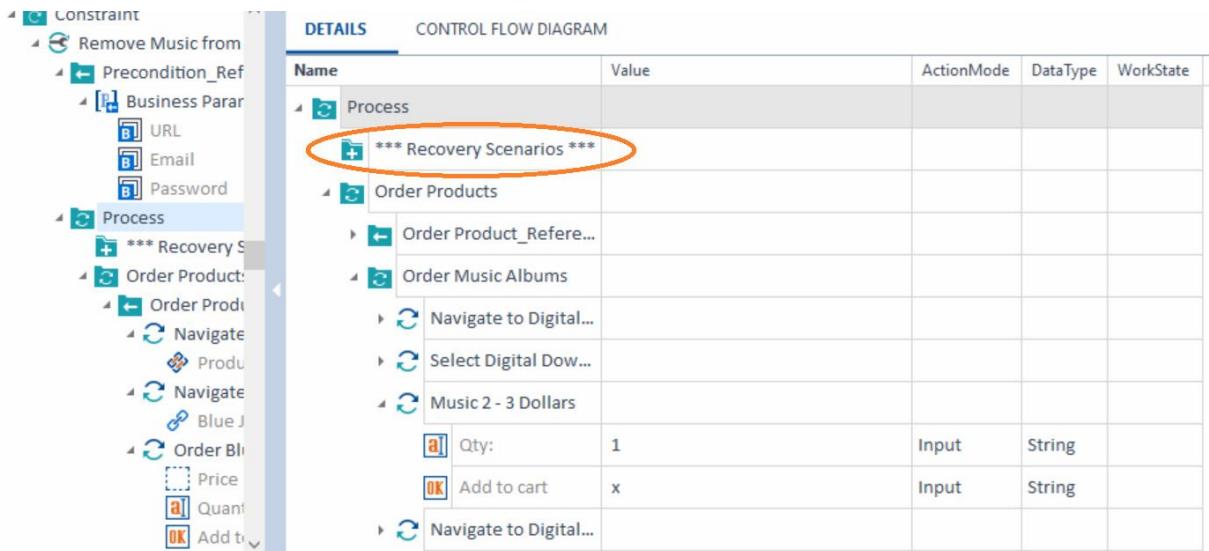
The screenshot shows the 'Tosca Settings [Role ADMIN] - C:\Program Files (x86)\TRICENTIS\Tosca Testsuite\Settings\XML\MetaSettings.xml' window. The left sidebar navigation tree includes 'Settings', 'Commander', 'Diagnostics', 'License', 'TBox' (which is expanded to show 'DokuSnapper', 'Dynamic temporal...', 'Engines', 'Execution', 'Execution recorder', 'Extension loading', 'Logging', 'Number formats', 'OCR', 'Recorder', 'Recovery' (selected), and 'Simulation'), and 'Simulations'. The main table lists recovery settings:

Name	Value	Comment
On dialog failure	Execute next TestCase	Choose what Tricentis Tosca should do in case of a dialog failure.
On exception failure	Execute next TestCase	Choose what Tricentis Tosca should do in case of an exception.
On verification failure	Continue	Choose what Tricentis Tosca should do in case of a verification failure.
TestCase retries	1	Maximum number of recovery attempts for a TestCase.
TestStep retries	2	Maximum number of recovery attempts for a TestStep.
TestStepValue retries	3	Maximum number of recovery attempts for a TestStepValue.

Now Set Recovery scenario in your project:

The screenshot shows the 'Control Flow Diagram' view. On the left, there is a tree structure with nodes like 'Remove Music from', 'Precondition_Ref', 'Business Parar', 'Process' (circled in orange), 'Postcondition_Re', 'Dynamic Comparison', 'Verify Order Numbe', 'Precondition_Ref', 'Business Parar', 'Process', 'Order Product', 'Navigate to', 'Product!', 'Navigate to', and 'Blue Jean'. The right side shows a 'DETAILS' table with columns 'Name', 'Value', 'ActionMode', 'DataType', and 'WorkState'. A context menu is open over the 'Process' node, with the 'More' option circled in orange. The menu options include: Create DO Statement (Ctrl+N, Ctrl+D), Create WHILE Statement (Ctrl+N, Ctrl+W), Create Recovery Scenario Collection (Ctrl+N, Ctrl+R) (circled in orange), and Create Manual XTestStep (Ctrl+N, Ctrl+M).

Then we get: Recovery Scenario collection



Now under Recovery Scenario Collection → Create Recovery Scenario.

*** Recovery Scenarios ***

+ Recovery Scenario

Order Blue Jeans

Right click and Create Scenario

Add any TS for which you want Recovery

Price	InnerText → PriceBlueJeans	Buffer	String
Quantity	25	Input	String
Add to cart	x	Input	String

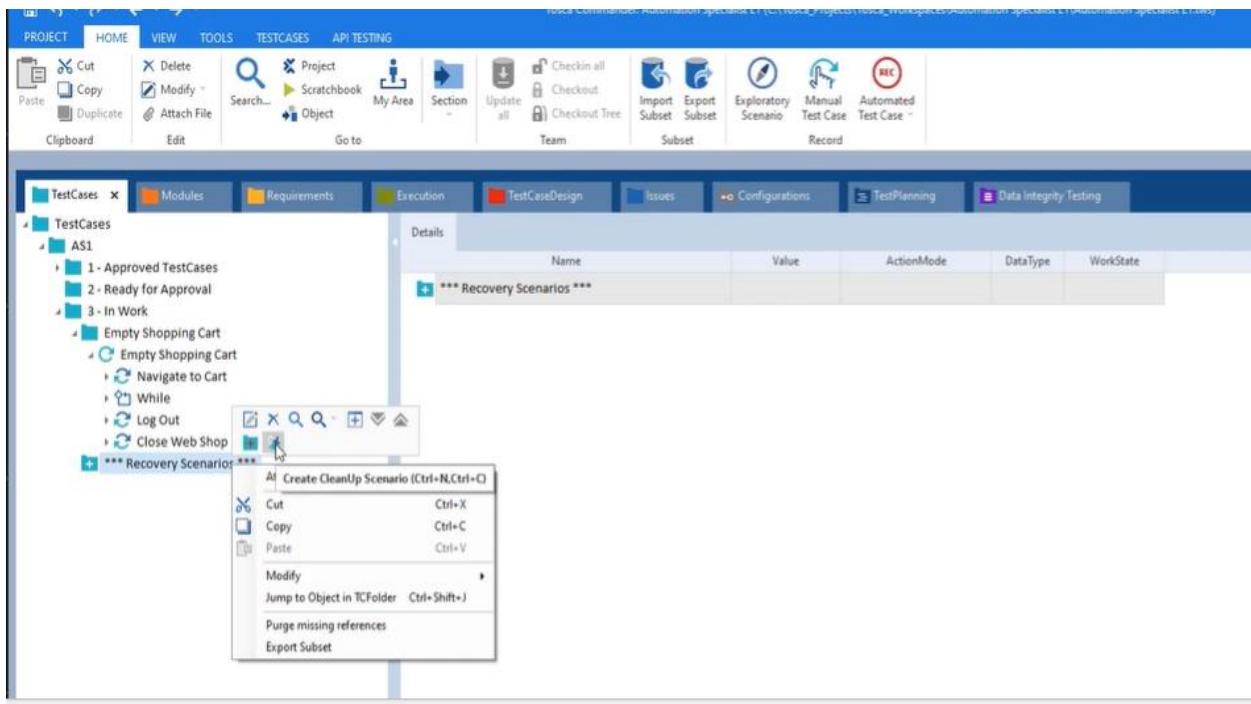
Remember not in Scratch Book--

Recovery Scenarios kick in
only during execution in
ExecutionLists

Lesson 31: Clear Scenario



You can add this clean up scenario in Recovery Scenario and must be run only in Execution list.



How to create it:

Now under Recovery Scenario Collection → Create Clean up Scenario

The screenshot shows a software application window with a toolbar at the top containing icons for TestCases, Modules, Requirements, Execution, TestCaseDesign, Issues, Configurations, TestPlanning, and Data Integrity Testing. The TestCases tab is selected. On the left, a tree view shows a folder structure under 'TestCases' labeled 'AS1'. The 'Execution' tab is active, displaying a table titled 'Details' with columns: Name, Value, ActionMode, DataType, and WorkState. A single row is present with the value '*** Recovery Scenarios ***'. A context menu is open over the 'Recovery Scenarios' entry in the table. The menu items are: Close Web Site, Create Cleanup Scenario (Ctrl+N, Ctrl+C), Cut (Ctrl+X), Copy (Ctrl+C), Paste (Ctrl+V), Modify, Jump to Object in TCFolder (Ctrl+Shift+J), Purge missing references, and Export Subset. The 'Create Cleanup Scenario' option is circled in orange.

Name	Value	ActionMode	DataType	WorkState
*** Recovery Scenarios ***				

Close Web Site

Create Cleanup Scenario (Ctrl+N, Ctrl+C)

Cut (Ctrl+X)

Copy (Ctrl+C)

Paste (Ctrl+V)

Modify

Jump to Object in TCFolder (Ctrl+Shift+J)

Purge missing references

Export Subset

Lesson 32: Dynamic Comparison

Dynamic Comparison

Value	ActionMode
Static XB Dynamic	Verify

XBuffer Syntax:
{XB}

Example in our Application once order is done we get at the end this order number 1079089(Which is Dynamic and keep on changing every time)



So we can use in our Application:

A screenshot of a software interface for managing test cases. The left sidebar shows a tree view of test cases, including sections like "TestCases", "Additional Material", "Exercises", and "Verify Order Number". The main area displays a "Control Flow Diagram" with a table titled "Details". The table has columns: Name, Value, ActionMode, DataType, and WorkState. One row in the table is highlighted, showing "Visible == True" under "Value", "Verify" under "ActionMode", and "String" under "DataType". A yellow callout box labeled "ATTENTION!" is overlaid on the interface, containing the text: "The static text portion of the string is case sensitive. Be sure to check capitalization for the verification to work properly."

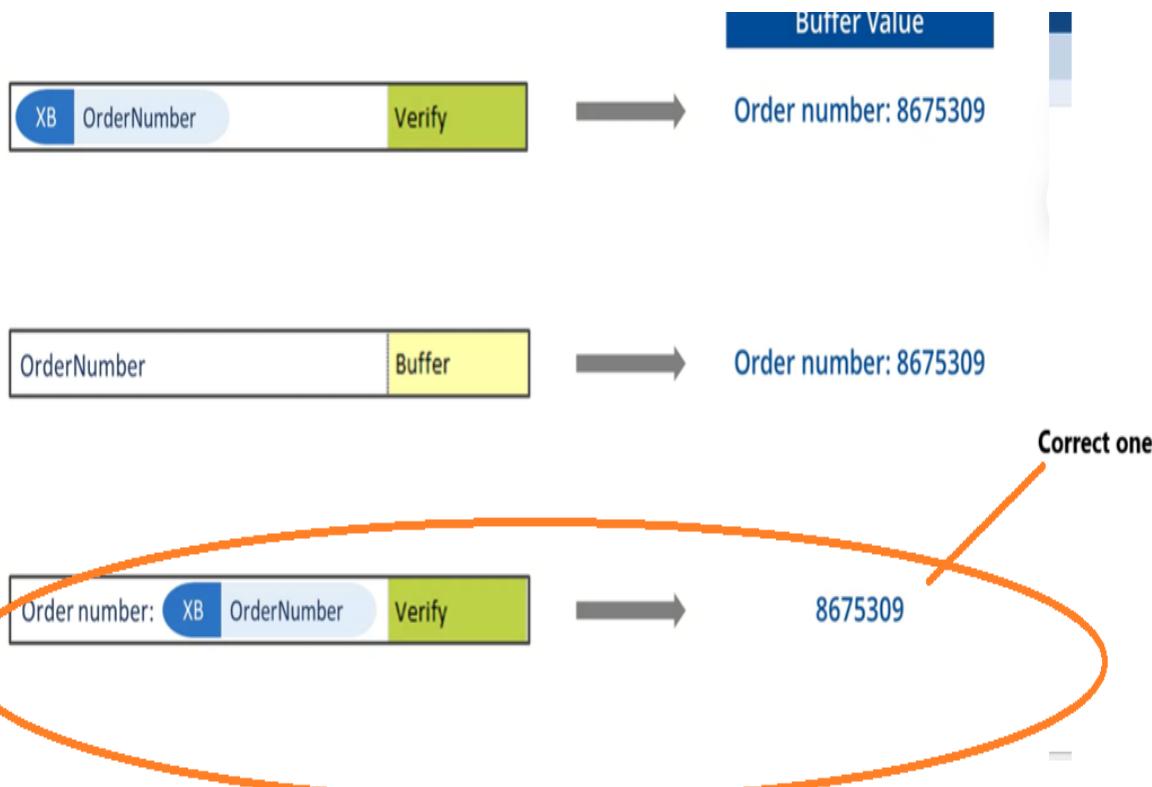
Name	ActionMode	DataType	WorkState
Visible == True	Verify	String	
Order Number		String	
Order Details		String	
Continue		String	

The screenshot shows a software interface for managing test cases and configurations. The top navigation bar includes tabs for TestCases, Modules, Requirements, Execution, TestCaseDesign, Configurations, TestPlanning, and Data Integrity Testing. The left sidebar is titled 'TestCases' and contains a tree view of test cases, including '*** Recovery Scenarios ***', 'Additional Material', 'Library', 'Lessons', 'Exercises', 'Recovery Scenarios', 'Constraint', 'Dynamic Comparison', 'Verify Order Number' (which is expanded to show 'Precondition_Reference', 'Process' (with 'Order Product_Reference', 'Start Checkout', 'Checkout Process_Reference', 'Verification of Prices', 'Confirmation_Reference', 'Verification of Success'), 'Navigate to My Orders', 'Check Order', 'Postcondition', 'ResultCount', 'Repetition', and 'Recently Viewed Items'), and 'Recently Viewed Items'. The main area displays a 'Control Flow Diagram' for a 'Verification of Success' step under a 'WebShop | Order Successful' scenario. A table lists configuration details:

Name	Value	ActionMode	DataType	WorkState
Message Order Successful	Visible = True	Verify	String	
Order Number	Order number: {XB[OrderNumber]}	Verify	String	
Order Details				
Continue				

.InnerText==Order number: {XB[OrderNumber]}=====Syntax used.

Explain below Clearly:



Lesson 33: Parental Control

Example:

The screenshot shows a demo web shop interface. At the top, there's a navigation bar with links for Register, Log in, Shopping cart(0), and Wishlist(0). Below the navigation is a search bar labeled "Search store" and a "Search" button. The main content area features a large banner for "Tricentis Academy" with a "Subscribe" button. To the left, a sidebar contains sections for CATEGORIES, MANUFACTURERS, and POPULAR TAGS. The CATEGORIES section lists Books, Computers, Electronics, Apparel & Shoes, Digital downloads, Jewelry, and Gift Cards. The MANUFACTURERS section lists Tricentis. The POPULAR TAGS section lists apparel, AWESOME, book, camera, cell, computer, cool, digital, gift, jewelry, nice, shirt, shoes, TCP, and View all.

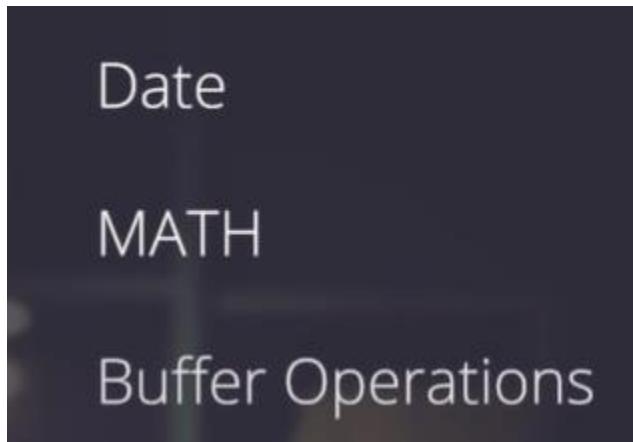
Not new already we covered it earlier:

The screenshot shows the XScan application interface. It displays a tree view of website elements under "ADVANCED VIEW". Two specific items in the tree view are circled in red: one item under the "Books" category and another item under the "Visible" property in the "Identify by Properties" panel. The "Identify by Properties" panel on the right lists various properties for the selected item, including Enabled, Focused, Id, InteractiveElement, IsSteerable, Label, Technical, and Transition. The "Visible" property is checked. The "Books" item in the tree view has a checked checkbox next to it.

Div mean Division (Just the short form)

Lesson 34: Dynamic ID

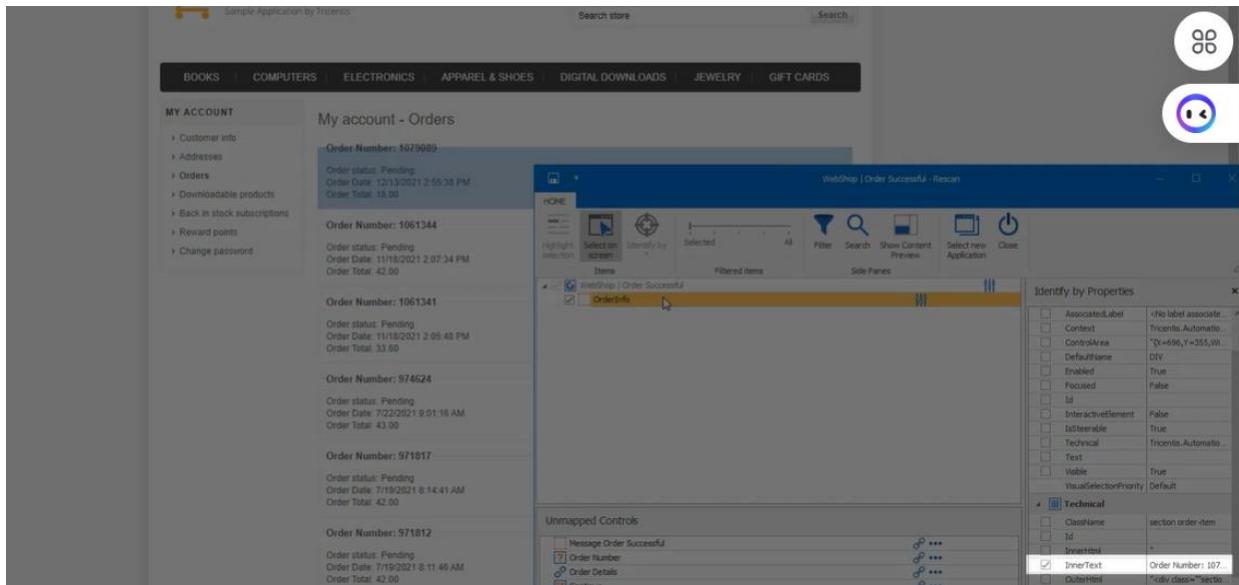
Like



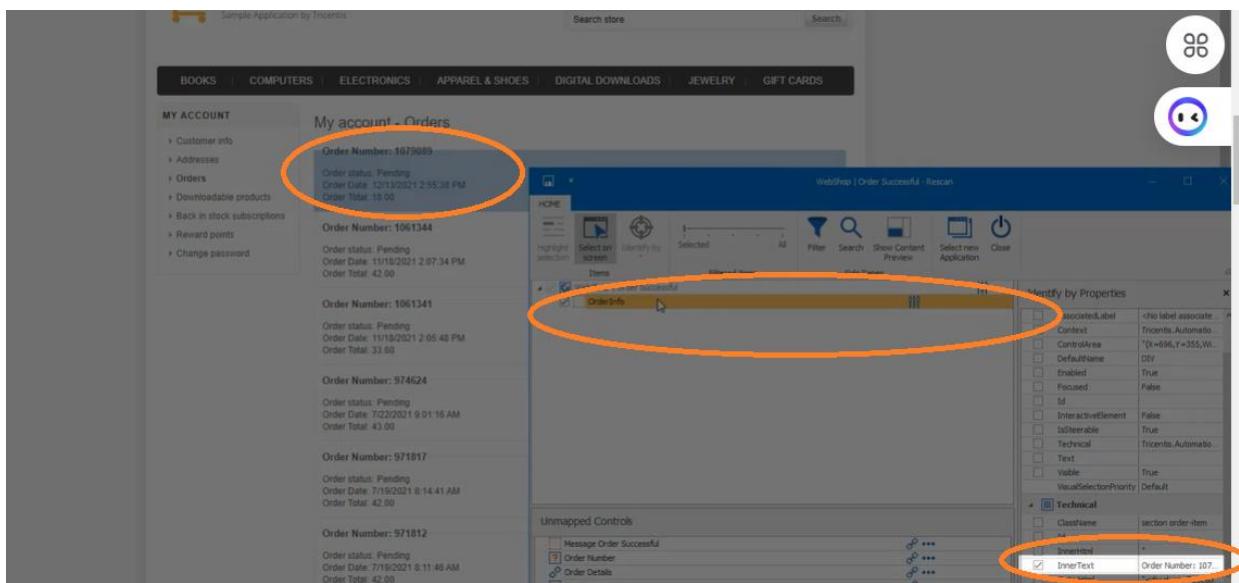
Explain about order and recent order Container

The screenshot shows a web store's account page titled "My account - Orders". On the left, there is a sidebar with "MY ACCOUNT" and links for Customer Info, Addresses, Orders, Downloadable products, Back in stock subscriptions, Reward points, and Change password. The main content area displays a list of recent orders:

Order Number	Date	Status	Total	Action
1079089	12/13/2021 2:55:38 PM	Pending	18.00	Details
1061344	11/18/2021 2:07:34 PM	Pending	42.00	Details
1061341	11/18/2021 2:05:43 PM	Pending	33.60	Details
974624	7/22/2021 9:01:16 AM	Pending	43.00	Details
971817	7/19/2021 8:14:41 AM	Pending	42.00	Details
971812	7/19/2021 8:11:45 AM	Pending	42.00	Details
940239				



One order container properties lets see:



Name	Value
Orderinfo	
NodePath	/Modules/Webshop/Customer/My account/11b Orders Page/Order Overview/OrderInfo
HasMissingReferences	False
Uniquelid	39eb63af-c4ba-61de-8bb9-d66d55911e74
Description	
BusinessType	Container
SpecialIcon	
Cardinality	0-1
DefaultDataType	String
DefaultActionMode	Select
InterfaceType	Implicit
DefaultValue	
BusinessAssociation	Descendants
Engine	Html
ExplicitName	True
ClassName	section_order_item
InnerText	*{!![OrderNumber]}*
Tag	

Note: This controls are uniquely Identified at Runtime only.

Lesson 35: Explicit Name (it is nothing but CP-Conf Par)



The screenshot shows the Tosca Studio interface with the following details:

- Left Sidebar:** Navigation elements including WebShop | Digital Downloads, Attachments (1), Digital Downloads, WebShop | Music 2 10 Dollars, WebShop | Music 2 3 Dollars, WebShop | Computers, WebShop | Products Choice Tab, and WebShop | Desktops.
- Details Pane:** Shows a list of objects under 'Digital Downloads':
 - Digital Downloads
 - 3rd Album
 - Music 2 - 10 Dollars
 - Music 2 - 3 Dollars
- Properties Pane:** Shows properties for the selected 'Digital Downloads' object:

Name	Value
Digital Downloads	

 A context menu is open over the 'Digital Downloads' row, with the 'Create Configuration Param (Ctrl+N, Ctrl+U)' option highlighted and circled in red.

Click enter and you can edit it as you require it

The screenshot shows the Tosca Studio interface with the following details:

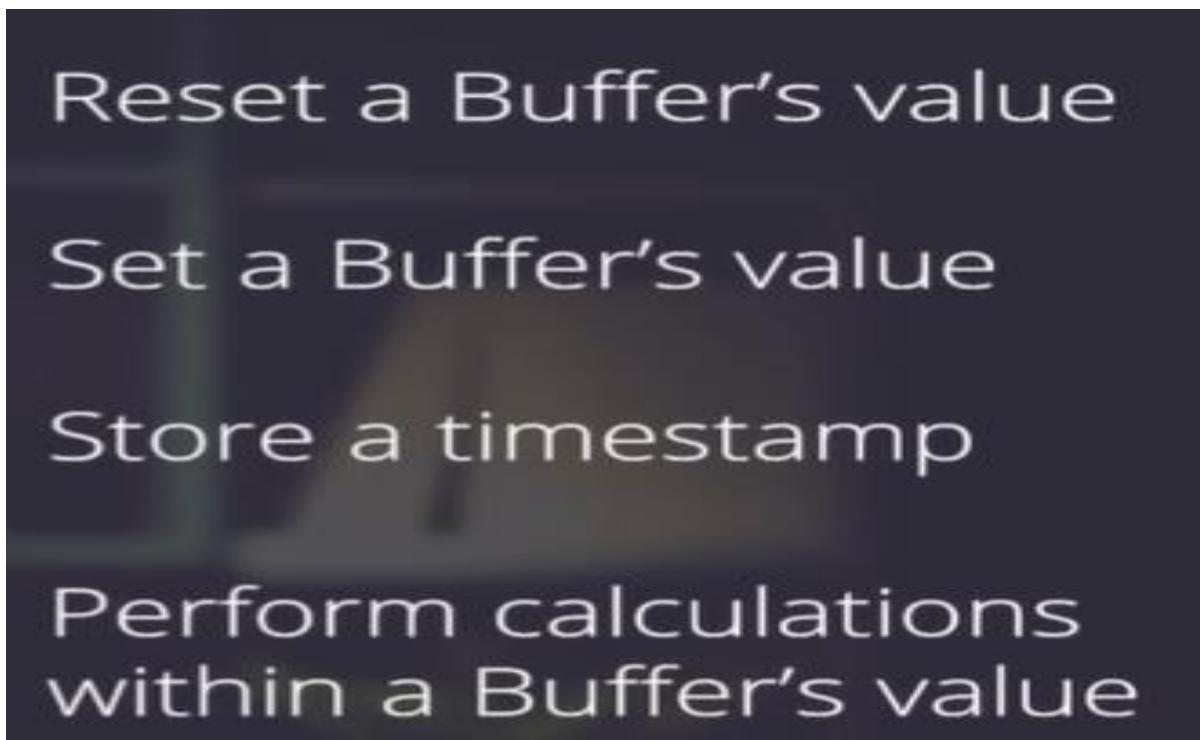
- Left Sidebar:** Navigation elements including WebShop | Top Menu, WebShop | My Account, WebShop | Recently Viewed Products, Order Confirmation and Details, Products, Additional Material, Ready for Approval, Work, and Standard modules.
- Middle Area:** A navigation element titled 'Recently viewed products' with the URL '[Click];[Rightclick]'.
- Properties Pane:** Shows properties for the 'Recently viewed products' navigation element:

NodePath	Value
CreatedBy	Admin
CreatedAt	27/12/2021 12:42:59
ModifiedBy	Admin
ModifiedAt	27/12/2021 13:03:47
HasMissingReferences	False
UniqueId	3a103de0-428a-4e6c-801b-175e29a8201b17
Description	Link
BusinessType	
SpecialIcon	
Cardinality	0-1
DefaultDataType	String
DefaultActionMode	Input
InterfaceType	GUI
DefaultValue	
BusinessAssociation	Descendants
Engine	HTML
ExplicitName	True
InnerText	recently viewed products
Tag	A

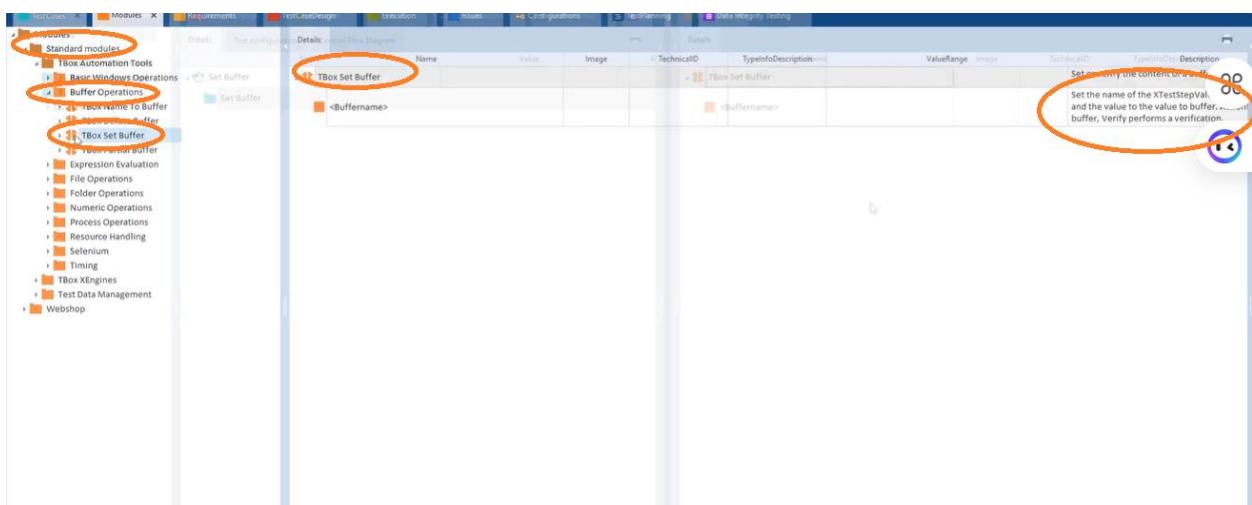
 The 'ExplicitName' property is highlighted and circled in red.

Note: we can give index of the product list say #1 etc too in order to select first product earlier –Explain them.

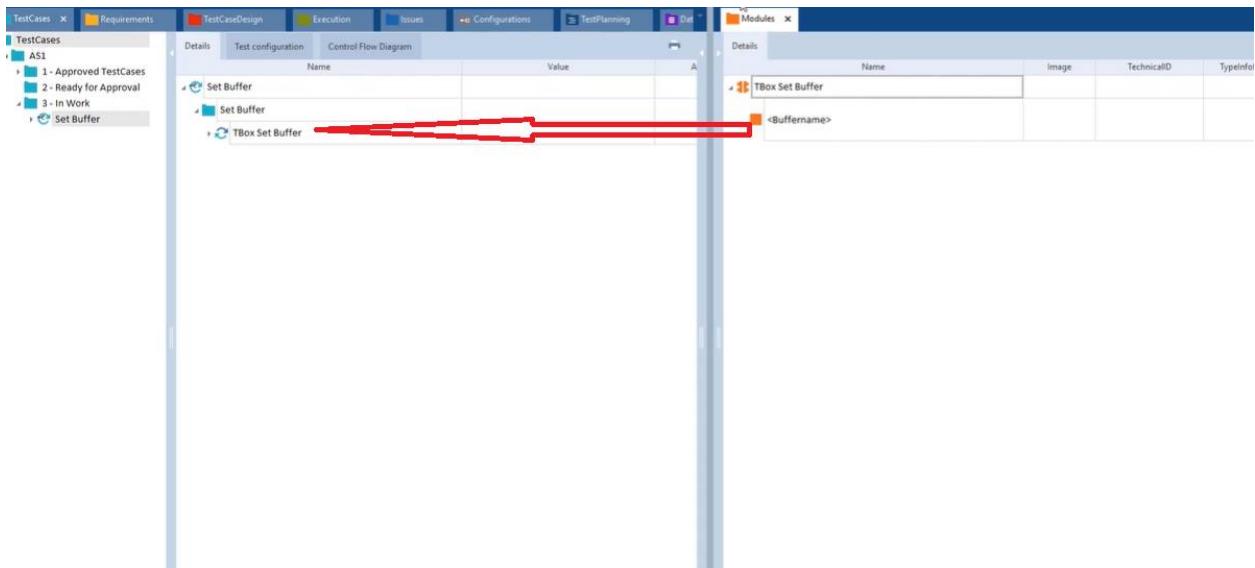
Lesson 36: Tbox Set Buffer



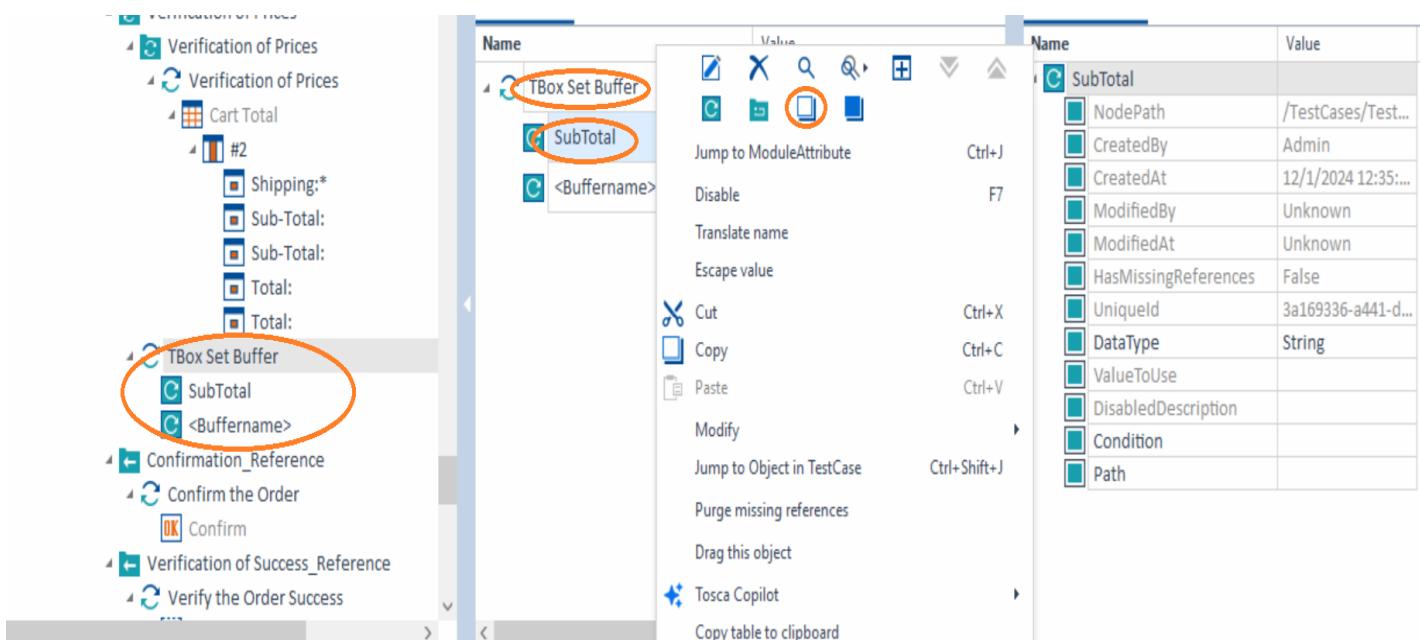
We can select under Standard modules → Buffer operations → TBox Set Buffer



Where every you need you can just add it to your TC and use it.



AS per the video it is not there but some enhancement are therein 24.0 latest versions---Explain them



You can add as many buffers you want under one TC and you must not leave Value blank otherwise—there is no point using buffer.

The screenshot shows a software interface for managing test cases. On the left, there's a tree view of test steps: 'Verification of Prices' (with two children), 'Cart Total' (with four children: 'Shipping:', 'Sub-Total:', 'Sub-Total:', 'Total:' each having a child 'Total'), 'TBox Set Buffer' (with two children: 'SubTotal' and 'Total'), 'Confirmation_Reference' (with two children: 'Confirm the Order' and 'Confirm'), and 'TBox Set Buffer' (with three children: 'SubTotal', 'Total', and 'Confirmation_Reference'). On the right, there's a 'DETAILS' section and a 'CONTROL FLOW DIAGRAM' table.

Name	Value	ActionMode	DataType	WorkState
TBox Set Buffer				
SubTotal	200	Input	String	
Total	1000	Input	String	

Annotations with orange circles highlight specific rows in the table:

- A circle around the first row 'TBox Set Buffer' with the text 'IT must not be Empty' below it.
- A circle around the second row 'SubTotal' with the text 'Always Input' below it.

Lesson 37: Result Count

You can use result count to count orders

Note—you can even use Row count/column count also in case you need.

The screenshot shows a web application for a 'Demo Web Shop'. The top navigation bar includes a logo, a search bar, and a 'Search' button. Below the navigation, there's a main menu with categories: BOOKS, COMPUTERS, ELECTRONICS, APPAREL & SHOES, DIGITAL DOWNLOADS, JEWELRY, and GIFT CARDS. On the left, there's a sidebar titled 'MY ACCOUNT' with links: Customer info, Addresses, Orders, Downloadable products, Back in stock subscriptions, Reward points, and Change password. The main content area is titled 'My account - Orders' and lists several orders with their details:

- Order Number: 1079089, Order status: Pending, Order Date: 12/13/2021 2:55:38 PM, Order Total: 18.00, with a 'Details' button.
- Order Number: 1061344, Order status: Pending, Order Date: 11/18/2021 2:07:34 PM, Order Total: 42.00, with a 'Details' button.
- Order Number: 1061341, Order status: Pending, Order Date: 11/18/2021 2:05:48 PM, Order Total: 33.60, with a 'Details' button.
- Order Number: 974624, Order status: Pending, Order Date: 7/22/2021 9:01:16 AM, Order Total: 43.00, with a 'Details' button.
- Order Number: 971817, Order status: Pending, Order Date: 7/19/2021 8:14:41 AM, Order Total: 42.00, with a 'Details' button.
- Order Number: 971812, Order status: Pending, Order Date: 7/19/2021 8:11:46 AM, Order Total: 42.00, with a 'Details' button.
- Order Number: 940239, Order status: Pending, Order Date: 6/23/2021 10:07:41 AM, with a 'Details' button.

You can use Result count ---to check beginning or end of page etc

Repeating a step a dynamic number of times

Going to the last item in a list

Verifying the expected number of items

We can use like this in order information

ResultCount == NumberOfOrders[Example]

The screenshot shows a software interface with a navigation bar at the top. The main area is a table with columns: Name, Value, ActionMode, DataType, and WorkState. A row is selected, and its details are shown in a modal-like window. The condition `ResultCount == NumberOfOrders` is highlighted with an orange oval.

Name	Value	ActionMode	DataType	WorkState
Set Buffer				IN_WORK
Set Buffer				
Set SUM to Zero				
SUM	10	Input	String	
OrderNumber		Input	String	
<Buffername>			String	
Count Number of Orders				
OrderInfo	.ResultCount==NumberOfOrders		String	
OrderTotal	ResultCount	=	==	NumberOfOrders
Details				
Message Order Successful				
Order Number				
Order Details				
Continue				

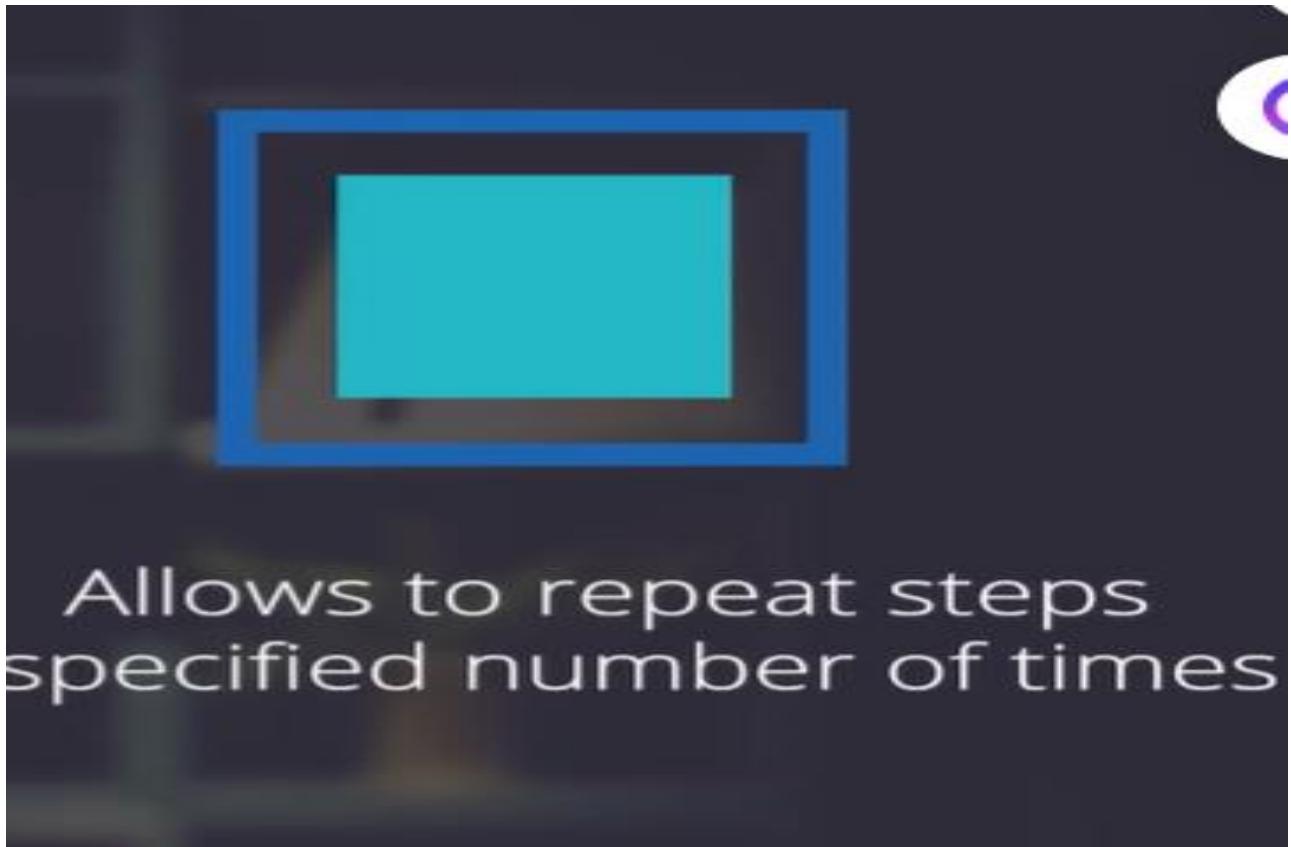
And ActionMode—Buffer

The screenshot shows a software interface with a navigation bar at the top. The main area is a table with columns: Name, Value, ActionMode, DataType, and WorkState. A row is selected, and its details are shown in a modal-like window. The `ActionMode` field is highlighted with an orange oval.

Name	Value	ActionMode	DataType	WorkState
Set Buffer				IN_WORK
Set Buffer				
Set SUM to Zero				
SUM	10	Input	String	
OrderNumber		Input	String	
<Buffername>			String	
Count Number of Orders				
OrderInfo	ResultCount == NumberOfOrders	Buffer	String	
OrderTotal			String	
Details			String	
Message Order Successful				
Order Number				
Order Details				
Continue				

NodePath	/Modules/Webshop/InWork/Customer/Addition/OrderExercise 27 - ExplicitName/Webshop Order Details/Order Overview
CreatedBy	Admin
CreatedAt	22/11/2021 15:14:30
ModifiedBy	Admin
ModifiedAt	22/11/2021 15:14:31
HasMissingReferences	False
UniqueId	39fffe1d5-dd74-6a08-8387-09ca96662490
Description	
BusinessType	Container
SpecialIcon	
Cardinality	0-1
DefaultDataType	String
DefaultActionMode	Select
InterfaceType	GUI
DefaultValue	
BusinessAssociation	Descendants
Engine	Html
ExplicitName	
InnerText	Order Number: Order number: DIV
OuterText	
Tag	DIV

Lesson 38: Repetition on Folder Level & ExplicitName on TestStepValue



Repeat a step:

- A certain number of times
- A random number of times
- A dynamic number of times

Repetition property can be found at every Test Step level—earlier you were deleting 2 times from cart one by one but with this linked with

Row Count and delete 4 or even Empty the cart

The screenshot shows a test management interface with the following details:

- Test Cases:** AS1
- Test Case Structure:** 1 - Approved TestCases, 2 - Ready for Approval, 3 - In Work, 3.1 - Empty Shopping Cart, 3.2 - Count Number of Items, 3.3 - Remove Shopping Cart.
- Control Flow Diagram:** A step labeled "Remove Shopping Cart" is highlighted with an orange circle.
- Properties Table:** Shows properties for the "Remove Shopping Cart" step.

Name	Value
NodePath	/TestCases/AS1/3 Work/Empty ShC Cart/Remove ShC
CreatedBy	Admin
CreatedAt	21/01/2022 15:20:34
ModifiedBy	Unknown
ModifiedAt	Unknown
HasMissingReferences	False
UniqueId	3a018f2f-8409-04da-efaa-01
Repetition	[{NumberofOrders}]
DisposableCopy	True
Reusable	Inherited
Pausable	Inherited
- Note:** An orange callout bubble points to the Repetition property in the Properties table with the text "You can use wherever you want".

From Tables:

— NUMERICAL VALUES

Static whole numbers

Math Expressions

Buffers

ResultCount

RowCount or ColCounts

— REPETITION ITERATION

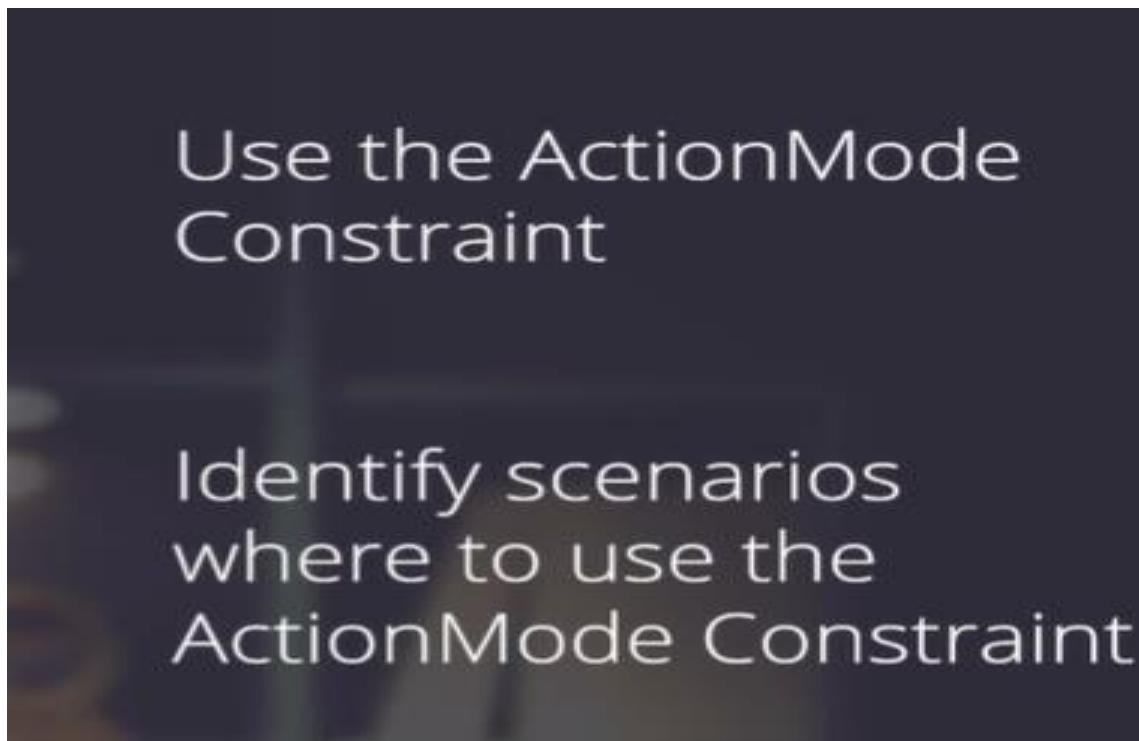
First Folder Repetition → {Repetition} = 1

Second Folder Repetition → {Repetition} = 2

Third Folder Repetition → {Repetition} = 3

The screenshot shows a software interface for managing test cases and requirements. On the left, there's a sidebar with 'TestCases' and 'AS1' sections. The main area has tabs like 'Details', 'Control Flow Diagram', and 'Properties'. In the 'Control Flow Diagram' tab, there's a table with columns 'Name', 'Value', 'ActionMode', 'DataType', and 'WorkState'. One row is selected, showing 'Remove' as the name, 'Select' as the value, 'Select' as the ActionMode, 'String' as the DataType, and an empty WorkState column. A yellow box highlights 'ActionMode 'Select''. Another yellow box contains the text 'Remember, checkboxes require True or False values for input'. In the 'Properties' tab on the right, there's a table with various properties for a node named 'Remove'. A callout arrow points from the 'Select' dropdown in the diagram to the 'Select' constraint in the properties table.

Lesson 39: Action mode construction



More than one property we need to search---Explain in detail

How to select “Music 2” with 10/-

- Automating web application testing (...)

Sample Application by: [redacted]

Search store

Search

BOOKS | COMPUTERS | ELECTRONICS | APPAREL & SHOES | DIGITAL DOWNLOADS | JEWELRY | GIFT CARDS

Shopping cart

Remove	Product(s)	Price	Qty.	Total
<input type="checkbox"/>	Music 2	3.00	<input type="text" value="1"/>	3.00
<input type="checkbox"/>	Music 2	10.00	<input type="text" value="1"/>	10.00
<input type="checkbox"/>	Computing and Internet	10.00	<input type="text" value="1"/>	10.00

Update shopping cart | Continue shopping

Discount Code

Estimate shipping

Sub-Total: 23.00

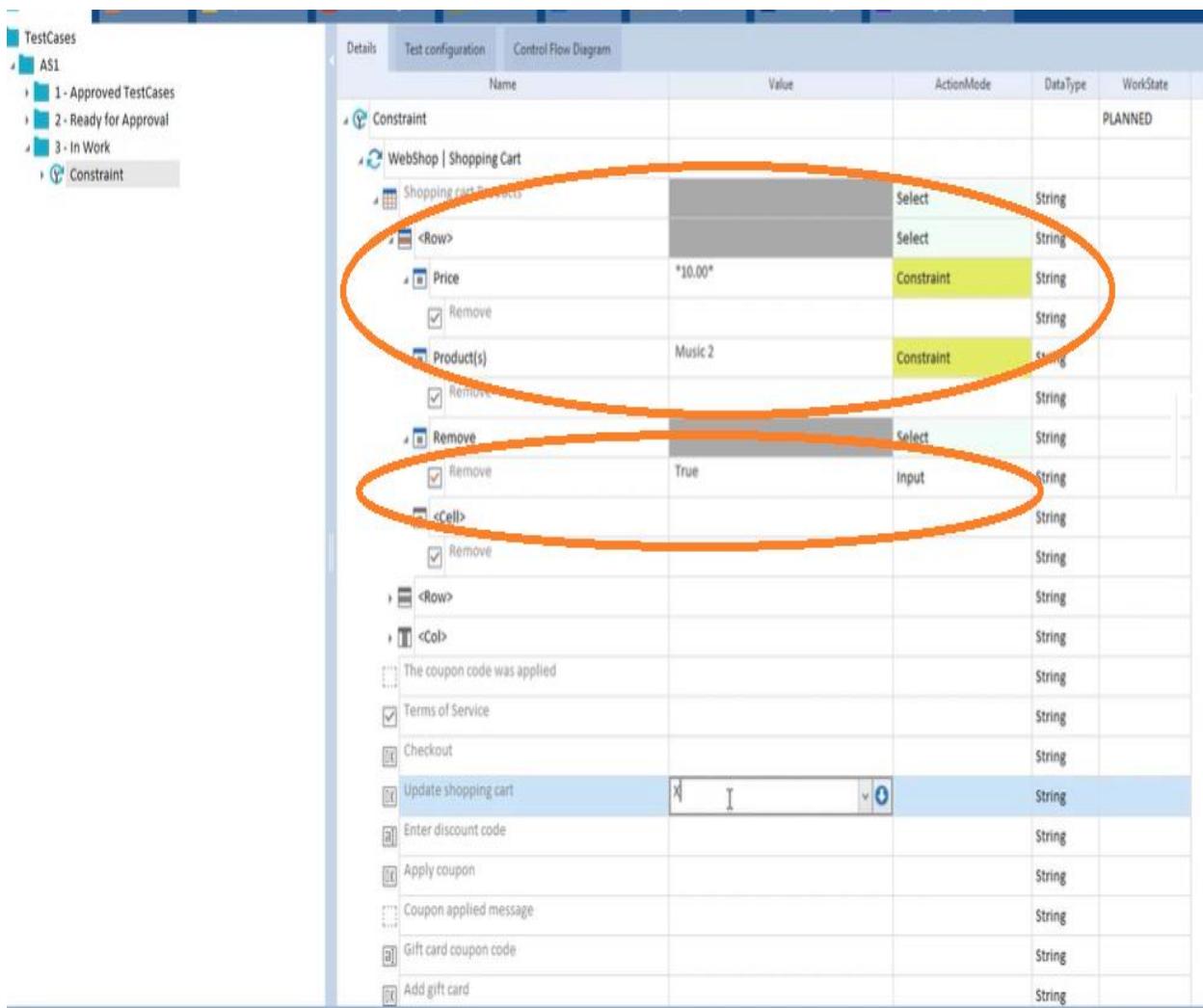
TestCases X Modules Requirements TestCaseDesign Execution Issues Configurations TestPlanning Data Integrity Testing

TestCases AS1 1 - Approved TestCases 2 - Ready for Approval 3 - In Work Constraint

Name	Value	ActionMode	DataType	WorkState
Constraint				PLANNED
WebShop Shopping Cart				
Shopping cart Products	Select	String		
<Row>	Select	String		
Price	"10.00"	Constraint	String	
<Row>				
Product(s)	Music 2	Constraint	String	
<Cell>				
<Row>				
<Col>				
The coupon code was applied				
Terms of Service				
Checkout				
Update shopping cart				
Enter discount code				
Apply coupon				
Coupon applied message				
Gift card coupon code				

Both of them must be checked then only pass otherwise fail

Irrespective of the row it will search and delete it:



The screenshot shows a software interface with a sidebar on the left containing a tree view of 'TestCases' under 'AS1'. The main area has tabs: 'Details', 'Test configuration', and 'Control Flow Diagram'. The 'Details' tab is active, displaying a table with columns: Name, Value, ActionMode, DataType, and WorkState. The table contains several rows, some of which are highlighted with orange circles. One row is highlighted with a large orange circle around its entire row. Another row, located below the first one, is also highlighted with a large orange circle around its entire row. The rows highlighted are: 1. A row with 'Price' and value '*10.00*' with 'ActionMode' set to 'Constraint'. 2. A row with 'Product(s)' and value 'Music 2' with 'ActionMode' set to 'Constraint'. 3. A row with 'Remove' and value 'True' with 'ActionMode' set to 'Input'. The 'WorkState' column for all rows is 'PLANNED'. The 'Value' column for the third row is 'True'. The 'ActionMode' column for the second row is 'Constraint'. The 'DataType' column for the first row is 'String'. The 'WorkState' column for the first row is 'PLANNED'.

Name	Value	ActionMode	DataType	WorkState
<Row>	Select	Select	String	PLANNED
Price	*10.00*	Constraint	String	
Product(s)	Music 2	Constraint	String	
Remove	Select	Select	String	
<Cell>	True	Input	String	
<Row>			String	
<Col>			String	
The coupon code was applied			String	
Terms of Service			String	
Checkout			String	
Update shopping cart	X		String	
Enter discount code			String	
Apply coupon			String	
Coupon applied message			String	
Gift card coupon code			String	
Add gift card			String	

=====End of AS1=====