1. **What is** Tosca**?**

* Automation testing tool
* Supports Web, desktop, SAP, API testing
* Model-based test automation
* Used for regression, functional, E2E testing

1. What is model based testing?

* Testing Approach: uses models instead of scripts
* Models = modules + TestSteps
* No coding required (drag&drop)
* Reusable components
* Faster automation & less duplication

1. What is module?

* Building block for Testcase
* Stores scanned objects of application (buttons, fields)
* Created using Xscan
* Reusable

1. What is Testcase?

* Set of Steps to validate a functionality
* Creates using Modules & Teststepvalues
* Can be manual or automated
* Supports data-driven testing using TCD
* Executed through Exe Lists

1. Explain Tosca Commander?

* Main interface
* Used to create, manage and execute tc
* Section:
* Modules 🡪 Stores scanned objects
* Testcases 🡪 Create automation flows
* Execution 🡪 run & track results
* Requirement 🡪 link & track coverage
* Testcasedesign 🡪 For data-driven testing

1. Explain Test steps?

* Action Step
* Created by modules
* Contains test step values (input, verify, buffer)
* Interaction with AUT

1. What are Test step values?

* Smallest unit of Teststeps
* Holds actual values or actions for field
* Types of actions:

1. Input – enter data
2. Verify – check expected result
3. Buffer – store data for reuse

* Define what to do

1. What are datatypes in Teststeps
2. String 🡪 text values
3. Number/integer 🡪 numeric values
4. Boolean 🡪 True/false
5. Date/time 🡪 date or timestamp values
6. Binary/file 🡪 files or encoded data for upload
7. What is Execution Section?

* Process of running testcase to validate functionality
* Uses Execution lists
* Features:

1. Run manual or automated tc
2. View pass/fail results in log
3. Supports parallel exe using DEX
4. Integrates with CI/CD tools like Jenkins
5. Allows subset optimization for faster regression
6. What are Execution lists?

* Stores testcases for execution
* Can be grouped by modules or functionality

1. What is Execution lists folders?

* Organize multiple execution folder
* Used for project or release-wise execution

1. What are Execution Entries (testcases)

* Individual testcases inside Execution Lists
* Shows status: Pass/fail/no run

1. Explain Execution Logs?

* Detailed step-by-step run results
* Includes screenshots and errors

1. What is Execution Entry point?

* Allows integration with CI/CD (Jenkins, ToscaCI)
* Define where execution should start

CI/CD Integration is responsibility is of DevOps/Build teams

1. Explain different fields in Module section?
2. Control name – name of element
3. Control type – (Textbox, button, dropdown)
4. TechnicalID/Property – unique property tosca uses to identify the element
5. Value/ Teststepvalue – input or verify values used during tc execution
6. Steering parameter – define how tosca interaction
7. Action mode
8. Explain all action modes?
9. Input – Enter a value into field
10. Verify – check if the value matches expected
11. Buffer – stores a value to use later
12. Select - Choose an option (dropdown, list)
13. Exists – Verify if element exists
14. DoNothing – Skip step but keep in Testcase
15. Constraint – Use condition for loops/Repetitions
16. Waiton – Wait until a condition is met
17. VerifyExists/ VerifyNotExists – Confirm element presence or absence
18. Contains/ VerifyContains – validate partial text or value
19. What are Requirements in Tosca?

* Represent business or testing req inside tosca
* Used to track coverage of Tc
* Helps in Traceability (req 🡪 tc 🡪 Exe 🡪 Result)

1. Created in Req. section
2. Linked to tc to measure coverage
3. Risk & Weight:🡪

* Risk 🡪 high/med/low (failure impact)
* Weight 🡪 0-5/ 0-10 (Priority/importance)

1. Coverage status:

* Passed 🡪 all linked tc passed
* Failed 🡪 any linked tc failed
* No run 🡪 no tc executed

1. Req. Coverage

* Shows how many tc are covering a req.
* Helps in subset optimization for tight regression
* Ensures high-risk req. are tested first

1. What are all Sections in Requirement?
2. Req. folders

* Organize req. module-wise or release-wise
* Helps in structuring by functionality or sprint

1. Req. Objects

* Actual business req. or story
* Contains key fields:

1. Name/Desc – Req. title
2. Weight – priority (1-10)
3. Risk – High/Med/Low
4. Status – Draft/In progress/ Complete
5. Shows all tc linked to that req.
6. Req. Coverage section

* Shows passed/failed/no run status
* Auto-calculated after EL runs

1. Requirement reports

* Provides coverage % and risk-based testing report

1. What is Xscan?

* Object scanner
* Used to identify and capture UI elements from application
* creates modules automatically for automation

1. Where Xscan is used?

* Web app (XBrowser)
* Desktop application
* SAP GUI
* Mobile apps

1. Benefits of Xscan

* Fast
* Support dynamic identification (Anchor, Explicitname)
* Reduces manual efforts

1. Describe all Scanning modes in Tosca?
2. Xscan

* Primary
* Supports web, Desktop, sap, mobile
* Highlights and captures UI elements
* Creates modules with controls

1. Tosca Wizard

* Quick Scan for web app
* Works as browser extension
* Captures elements directly from web page without opening xscan

1. LiveCompare

* Integrated tool for SAP testing
* How it works

1. Connect Live compare with SAP system & tosca
2. Analyze transports/updates in SAP
3. Identify impacted programs, transcations, or tables
4. Generate list of affected tc

* Its separate SAP impact analysis tool
* Not inside tosca
* Web-base livecompare portal
* Integration with tosca
* SAP based/ Admin team sets up LiveCompare
* Setup handled by Admin/DevOps team, not tester

1. TBox / Engine Scanning

* Auto recognized elements using TBox engines
* Used when actions modes are applied

1. Tosca mobile scan

* For iOS/Android app

1. Give all Different Identification controls in Xscan?
2. Identify by property: Uses elements properties (ID, name, class, value)
3. Anchor: uses nearby stable element

* Best for dynamic tables or controls

1. Index: by Position/Order in list or table
2. Image (VisionAI/Image-based): Uses Screenshots

* Used for non-standard UI or Citrix/Remote Apps

1. OCR/Text Recognition

* Uses text recognition when properties are mission
* Ex. PDFs or scanned application

1. XPath/CSS (web): for web applications

* XPath/CSS Selector can locate elements in DOM

1. Relation: Parent-child relationship in UI tree

* Ex. Find button inside a specific panel

1. What is TCP?

* Project-level Parameter
* Stores configuration values like URLs, Browser type, env.

1. Purpose:

* Avoids hardcoding values
* Centralized management
* Support multi-env. testing (QA, UAT, Prod)
* Preference: Folder > EL > TC
* Use: {CP[Paramname]}

1. Used for

* Environment Details

1. Application url
2. API endpoints
3. Database names/server names

* Browser & Execution Setting

1. Browser
2. Headless = Yes/No
3. Parallel Execution = Yes/No

* Credentials & keys

1. Username
2. Access tokens/ API Keys

* Path/File locations (system)

1. Upload/Download Folder paths
2. Test data excel file location

* Feature Toggles/Flags

1. Dynamic Value

* Values that change during test execution instead of being fixed

1. Purpose

* Handle data that changes each run (ex. Timestamp, ID)
* Avoid hardcoded values
* Reusable tests

1. Ways to handle
2. Buffer
3. {Random} function – generate random text, number, dates
4. Date/time function – {DATE[]}
5. Steering Parameters

Use dynamic id and ExplicitName for changing UI elements

1. TBox functions

TBox set buffer – capture values Dynamically

TBox Get Text – Extract runtime values from UI

1. TCP

Pass env. -dependent dynamic values

1. Library

* Collections of Reusable components to maintain modular and maintainable automation

1. What are libraries?

* Central storage of reusable modules, testcases and templates
* Reusability
* Reduce maintenance

1. Types
2. Modules

* Stores all scanned modules (web, sap, API, Desktop)
* Reused to create tc

1. Reusable testcase library

* Stores common tc:
* Login/logout
* Data creation
* Pre-condition

1. Template/Teststep library

* Contains tc templates and reusable ts

1. Standard library

* TBox
* Classic engine
* Standard modules

1. Reusable Teststep block

* Group of Teststeps combined as a single reusable block
* Can be called in multiple tc
* Works like function

1. Value Range

* Set of allowed values for teststep values or tcd parameters
* Used to define multiple input options for testing
* Helps in data-driven testing

1. Uses

* TCD
* Templates and teststeps
* Risk based testing

1. How to create

* Tcd🡪 Attributes
* Add value range
* Add multiple values using (;)
* Map to tc template

1. What is Business Parameter

* High level representation of an end-to-end business workflow in tosca
* Made in multiple tc/ teststepblock arranged in seq.

1. Single user vs multi user workspace
2. Single user workspace

* 1 person only
* Local machine
* Small project or practice

1. Multi user workspace

* Multiple testers
* Central common repo
* Real time team collaboration in project

1. Recovery scenario

* Set of steps to handle unexpected errors or interruptions during test execution
* Recover and continue testing instead of failing complete execution

Handle popups, system error, crashes

Smooth execution in long regression runs

1. What is cleanup scenario

* Set of steps executed after test execution
* Used to reset environment or clean test data
* Ensure system is ready for next execution
* Purpose
* Remove temporary test data
* Close application/browser
* Reset environment to initial state

Create RS and CS in execution section

**6️⃣ Key Difference vs Recovery Scenario**

| **Feature** | **Recovery Scenario** | **Cleanup Scenario** |
| --- | --- | --- |
| When it runs | **On error/failure** | **After execution ends** |
| Purpose | Continue test after error | Reset/clean environment |

1. What is dynamic comparison

* Verification method
* Compares expected vs actual values during runtime
* Handles values that change dynamically

1. Purpose

* Avoid false failures when values are not constant
* Useful for dynamic fields like:

1. Order id
2. Dates & timestamps
3. Transaction numbers
4. What is Parent control

* Stable reference element used to identify a child element
* Helps locate elements in dynamic tables or grids
* Works like parent-child in UI

1. Dynamic id

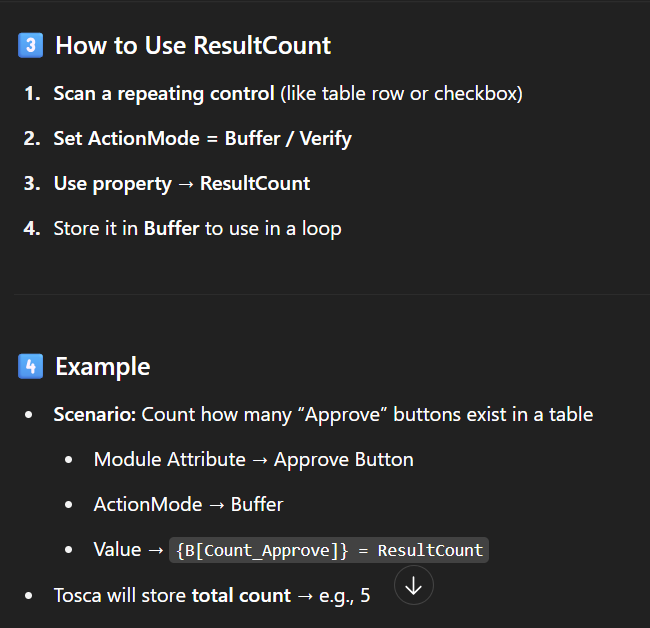
* When an element’s ID changes every time the page loads
* Tosca cannot find elements with normal identification
* Use dynamic id
* Use wildcard (\*)
* Use anchor/parent control
* XPath or CSS

**3️⃣ Key Difference**

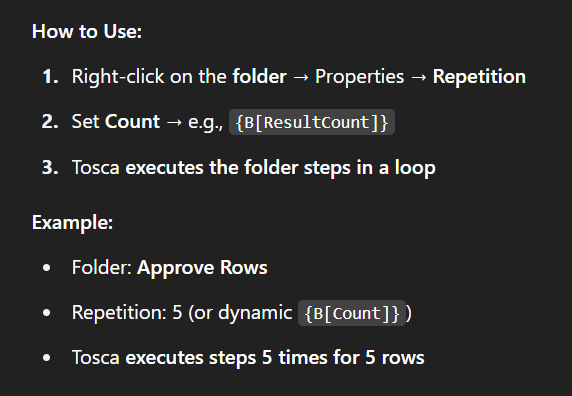
| **Feature** | **Dynamic ID Handling** | **Explicit Name** |
| --- | --- | --- |
| Purpose | Identify **unstable elements** | **Readable test steps** & logs |
| Usage | Module scanning / Object ID | TestSteps / TestStepValues |
| Applied | Xscan- properties \* | In tc properties add  Explicitname = true |
| Impact on Execution | Stability of test | Reporting & readability only |

1. What is Result count?

* Tosca Property that returns the number of matching elements for a given control during execution
* Used when multiple dynamic elements are present (like rows, buttons, checkboxes)
* Purpose
* Handle dynamic lists or tables
* Support loops/repetitions for multiple matching elements
* Validation if expected number of elements exists



1. Repetition on folder



1. Explicit name

* Custom name for teststep/Teststepvalue