

Tosca cheat sheet

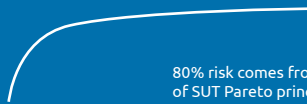


sogeti
Part of Capgemini

Requirements

State of TestCase	Coverage
PLANNED	20%
IN_WORK	50%
COMPLETED	100%

Frequency x Damage = Weight



80% risk comes from 20% of SUT Pareto principle

Linking

TestCase <> Requirement
ExecutionList <> RequirementSet

Modules

Maximum 15-20 attributes

- Split modules for SUT areas
- Use control groups

Re-use of attribute

- Use property Cardinality

Configuration parameter

- AvoidSnapper
- ExplicitName

Steering parameter

- LoadStrategy
- ScrollingBehaviour (Standard = Center)
- User Simulation

TestCases

If statement > replace by conditions
Do/While statement > replace by repetition

Ensure a library consists of:

- Business parameters {PL} for all configurable values in the test steps
- One clearly-defined action
- Verification of the action included

Scratchbook

No recovery
No clean-up
Not run on agent
Scratchbook

vs. ExecutionList

Recovery & retries
Clean-up
Agents can be used
Buffer TC name gives:
Actual TC name

TestCaseDesign

Business relevance

- Yes
- No
- Result

- Valid/Inner value
- StraightThrough
- Invalid
- Boundary

Character/Position

- Valid/Inner value
- StraightThrough
- Invalid
- Boundary

Dedicated test goal per test sheet

Generate logical testcases based on chosen test coverage

	All combos	Orthogonal	Pairwise	Linear expansion
Location				
City	12	2	8	6
Country	12	2	4	1
Age				
<18	6	1	3	1
18-23	6	1	3	1
24-65	6	1	3	4
>65	6	1	3	1
Gender				
F	8	1	4	1
M	8	2	4	5
X	8	1	4	1
Total instances	24	4	12	7

ExecutionList

Set-up your agents configurations to match different testing requirements during Execution (e.g. Office installed, browser versions)

Testcase can only be added ONCE to an ExecutionList

State the purpose of the ExecutionList in its name (e.g. SmokeTest for specific application/team)

For performance of the Tosca project archive ExecutionLogs daily -> Trend chart can still be used with archive

Reporting

Make use of the standard reporting available through the Standard subset

Ensure that different levels of reporting are available for different needs of users and try to make use of existing tooling available in your workplace

Beware that the Reporting folder can only be used once within your project structure, therefore it's advised to put it at the Project Root level

Create root cause analysis for failure in test automation scripts

Settings are related to the project and can be overruled anywhere in the project using Test Configuration Parameters {CP}

- SynchronizationTimeout
- TargetDecimalCharacter
- TestCaseRetries
- WebserviceEngineLogRequestResponse (not a setting but really useful to check request and responses)

Options are related to the local installation, such as:

- Autosave
- See graph of execution over time
- Log TestStepFolder in ExecutionList

General -> minutes between autosaves (disable with -1)

View -> Enable trend charts

General -> Advanced -> Create Logs for TestStepFolders