UPES

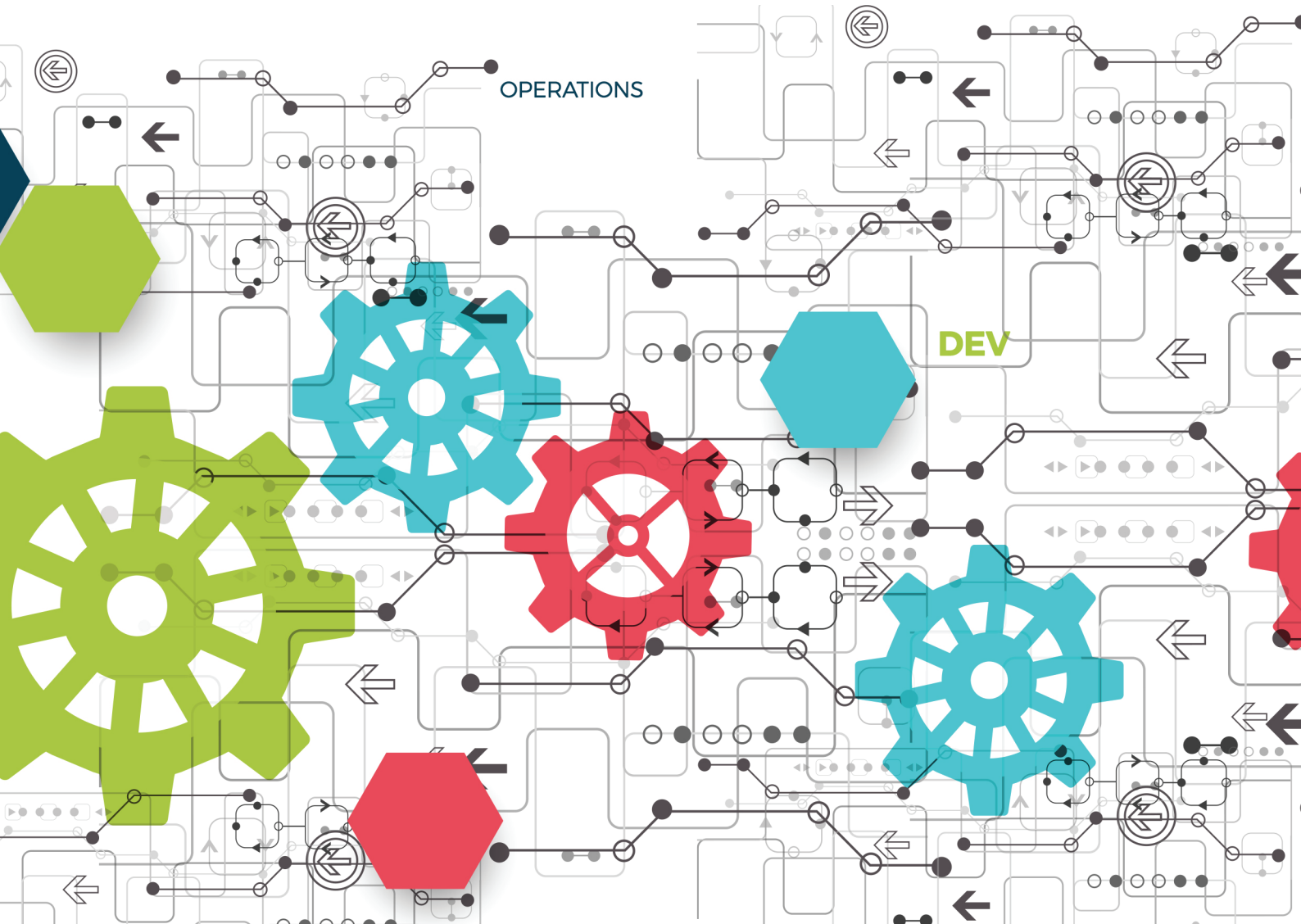
Xebia

YEAR 3

B.Tech Computer Science and Engineering in DevOps

Test Automation

MODULE 5
Introduction to Test Design



Contents

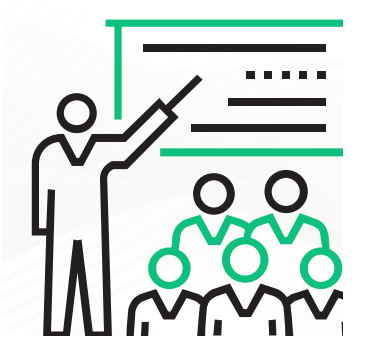
MODULE 5

Introduction to Test Design

You will learn about the ‘Software Testing’, best practices and Automation Testing

Module Objectives

At the end of this module, you will be able to learn:



- Understand test scenario.
- Explain test case design.
- Define test basis.
- Describe traceability matrix.

Test Case Design and Development is an integral part of Testing. This is the heart of the Software Testing process. A good Test design leads to better analogies of software testing concepts and this shall lead to better testing and finally a better tested product.

At the end of this module, you will be able to learn:

- Understand test scenario.
- Explain test case design.
- Define test basis.
- Describe traceability matrix.

Module Topics

Let us take a quick look at the topics that we will cover in this module:

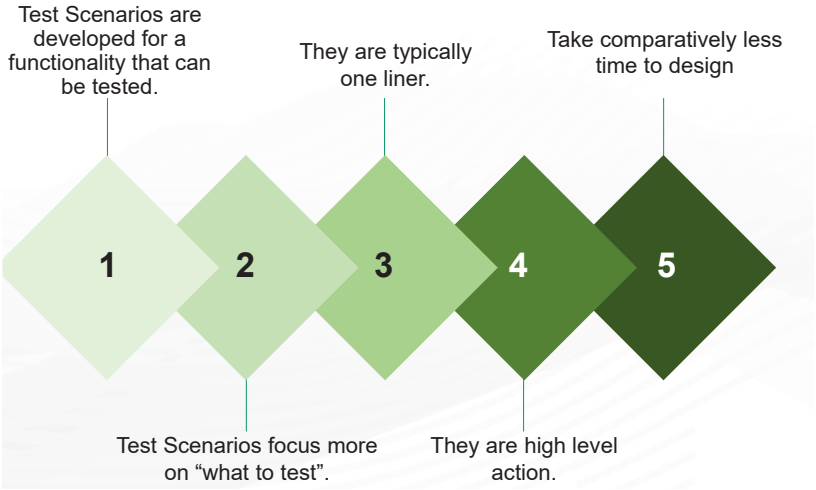
- Test Scenario
- Test Case Design
- Test Basis
- Traceability Matrix

You will learn about the following topics in this module:

- Test Scenario
- Test Case Design
- Test Basis
- Traceability Matrix

1.5.1 Test Scenario

Test Scenarios are also known as Test Condition / Test Possibility.



- Test Scenario is defined as any functionality that can be tested.
- Test Scenario gives a high-level idea of what we need to test.
- Test Scenario does not focus much on how to test
- Test Scenario is more popular in Agile Methodologies

1.5.1 Test Scenario

Benefits

- Test Scenario help in attaining test coverage.
- They assure that all use cases are working as intended.
- End-to-End functionality is understood in a faster way by Test Scenarios.
- Helps attain testing in fast paced environments.
- Easy to understand even for a new comer.

Let's understand by an example a few test scenarios of an E-Commerce Web Application.

Test Scenario 1 : Check the Application URL is opening with Chrome, Firefox, Microsoft Edge.

Test Scenario 2: Check the landing page loads within 6 seconds in all the compatible browsers

Test Scenario 3 : Check the Login Functionality

Of course, there will be a lot of arguments which will be like – "There is so much which we need to test and mention and test scenarios are not detailing any of it"

Well that is what it is. Test Scenarios are high-level actions which will cover the high priority use cases. Test scenarios are highly preferred when there is a time crunch and documenting detailed test cases to test the application is practically impossible. Test Scenarios help in achieving test coverage and ensuring work gets done.

1.5.1 Test Scenario

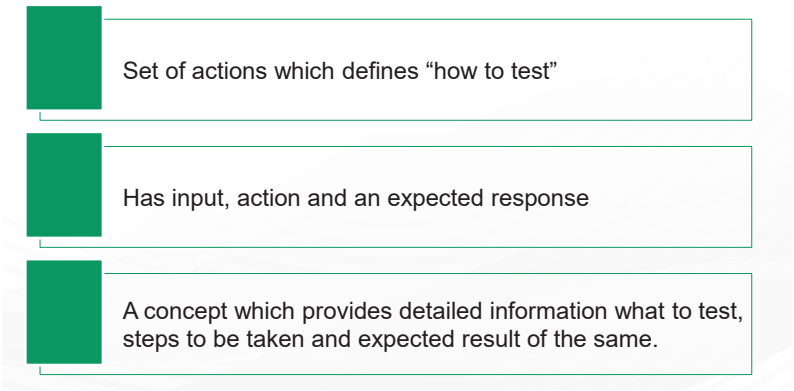
When not to use test scenario.

In case, the application has a highly complex business logic with multiple interfaces behaving differently as a cohesive unit and behaving differently as an individual component, in such cases, test scenarios are not recommended.It will not be able to handle thorough testing of such kind of an application.

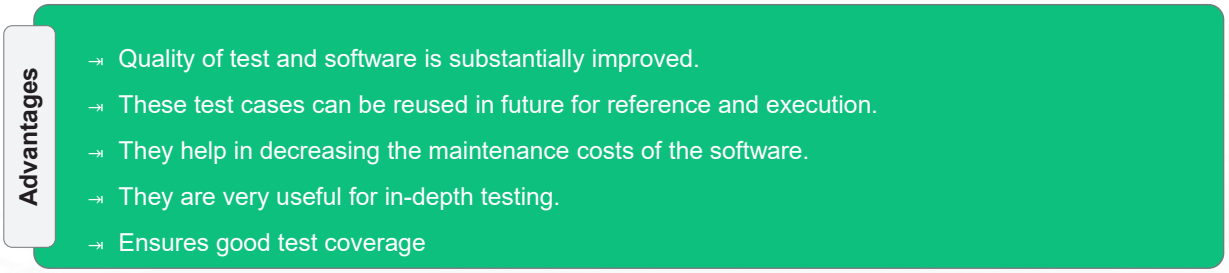
If an article in unstable and it behaves differently under different set of inputs, different loads and different environment. In that case, using test scenarios will not be able to do justice to proper testing of such an application.

1.5.2 Test Case Design

In simple words, test case design is



Test Case describes a specific idea that is to be tested, without detailing the exact steps to be taken or data to be used. They are detailed actions. Needs lot of documentation efforts. They describe a specific idea which has to be tested and they have certain ground rules how to go about doing this activity.

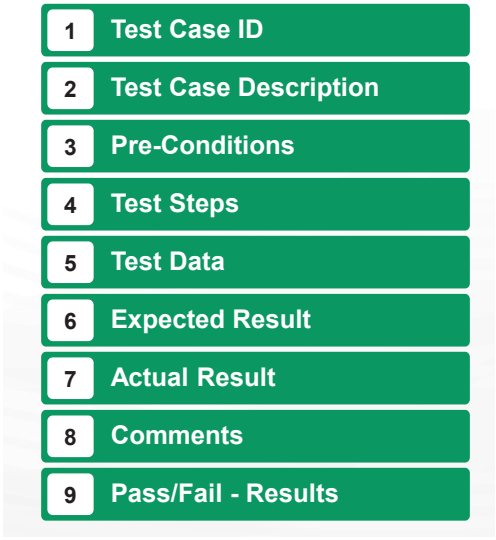


Test cases are like bread and butter of any software tester. A tester cannot be a good tester unless he/she learns and writes effective test cases. The designing of test cases actually showcases a tester’s skills, knowledge and ability to carry out testing in an effective manner. Also test cases are quite helpful while raising defects and during regression testing

Test cases are powerful artifacts that work as a good source of truth for how a system and a particular feature of software works.

Designing a Test Case:-

The basic parameters which comprise to design a test case are:-



Test Case ID : This might sound like a small representative of a larger action but this small representative is highly helpful in tracing.

Test Case Description : Gives a brief idea of what is going to be tested

Pre-Conditions : Any assumptions that apply to the test and any preconditions that must be met prior to the test being executed should be listed here.

Test Case Steps: The test steps should include the necessary data and information on how to execute the test. The steps should be clear and brief, without leaving out essential facts.

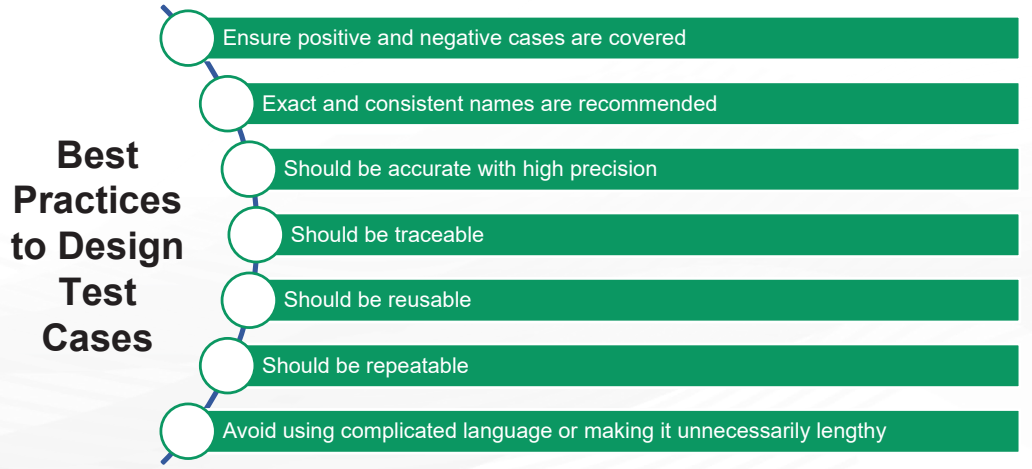
Test Data: It's important to select a data set that gives sufficient coverage. Select a data set that specifies not only the positive scenarios but negative ones as well.

Expected Result: The expected results tell the tester what they should experience as a result of the test steps.

Actual Result: They specifies how the application actually behaved while test cases were being executed.

Comments: Any other useful information such as screenshots that tester want's to specify can be included here.

Pass/Fail – Results : Post execution the results are mentioned in this column



Simplicity is of essence. Writing in clear and concise words without too much ambiguity is more acknowledged rather than using unnecessary jargons to design a test case. Testers should write test case in such a manner that someone new to the system get to read the test cases, very soon he/she should be able to grasp the content in the test case and work accordingly.

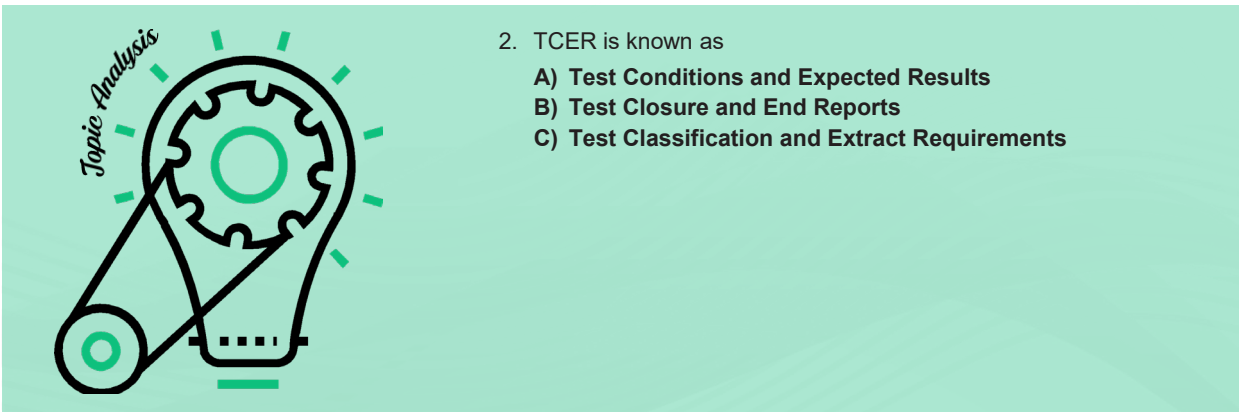
Avoiding repetition is also important. Imagine repeating the same thing and making It further lengthy. Generally, a lot of copy paste happens in real time designing so these kind of human mistakes do happen. Just that we have to be careful and keep these things in account.

Do not assume functionality and features of the software application while preparing test case. Stick to the Specification Documents.

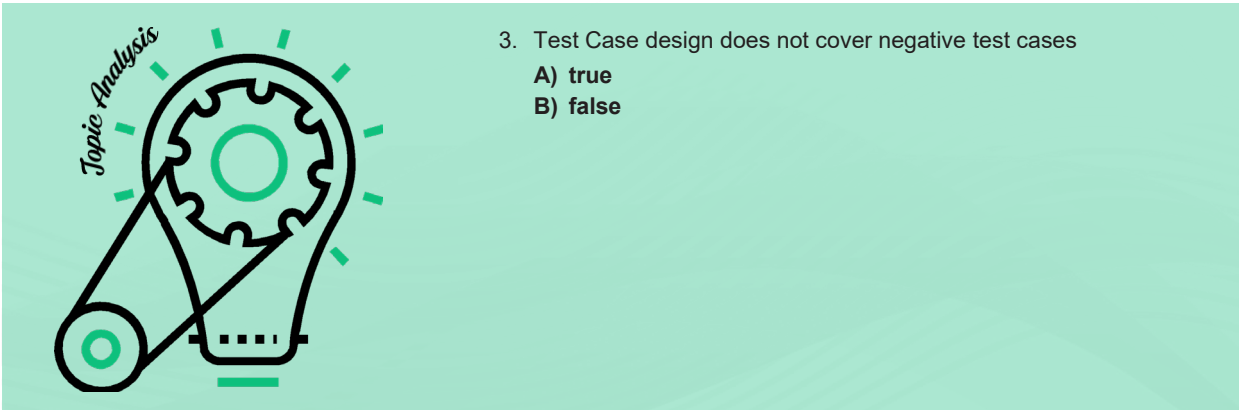
What have we learnt so far ?

1. Which is not a Test Document?
   - **A) Test Policy**
   - **B) Test Case**
   - **C) PIN (Project Initiation Note)**
   - **D) RTM (Requirements Traceability Matrix)**

Answer: 1 : (c)



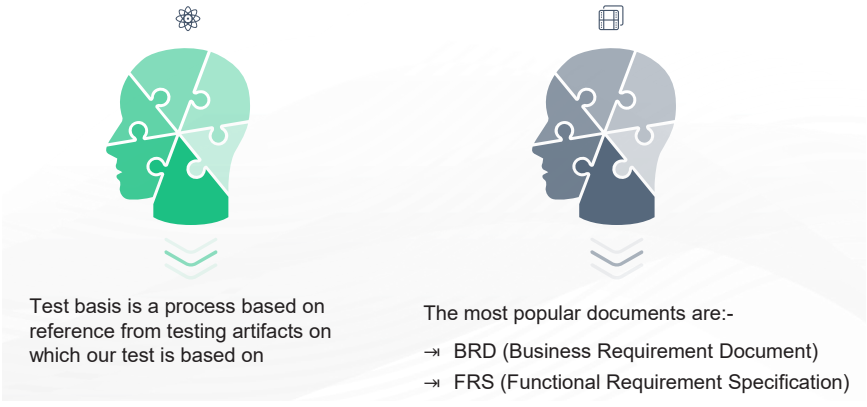
Answer: 2 : (a)



Answer: 3 : (b)

1.5.3 Test Basis

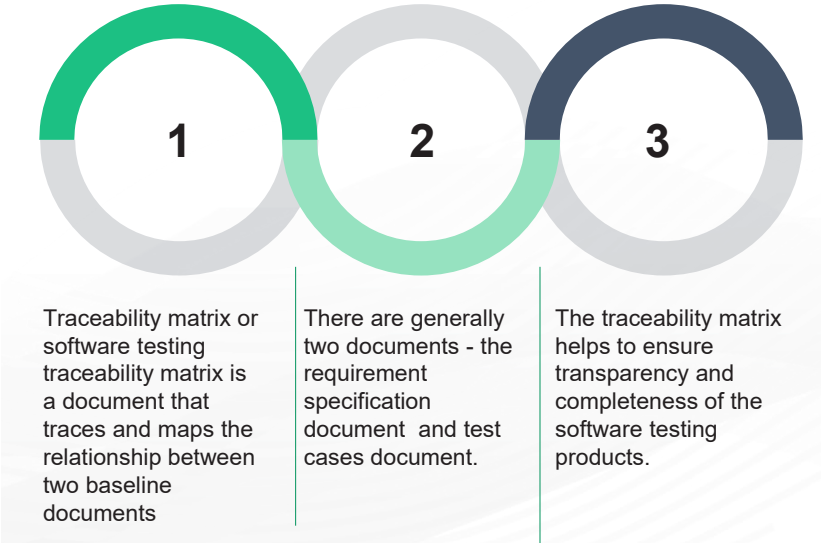
Test basis is also known as Test Analysis.



Business Requirement Document : In some cases gets directly transferred to the testing team which tells about the changes that have been implemented and it highlights the areas which needs to be tested.

Functional Requirement Specification: is more specific to a certain functionality or group of functionalities. It is irrelevant whatever document a testing team gets. What matters most is that they should be smart enough to deduce from these documents what needs to be tested and design their test cases accordingly.

1.5.4 Traceability Matrix

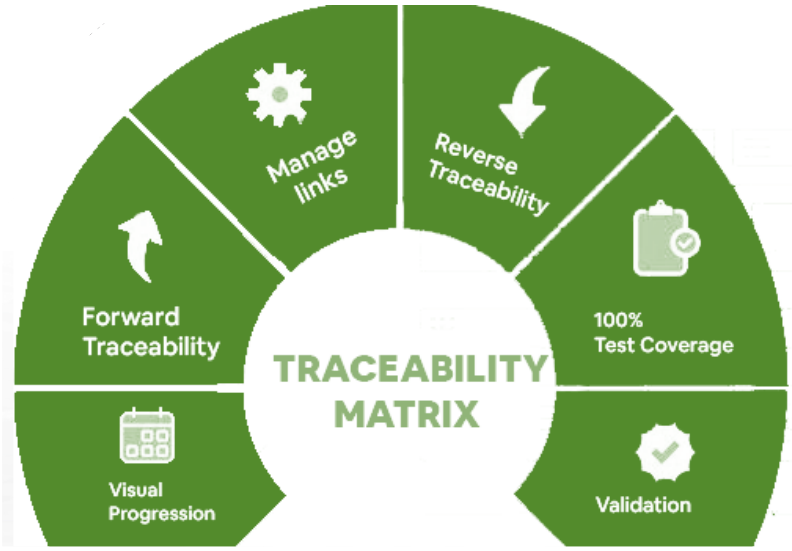


Traceability matrix is formulated out of Requirement Document. It is a formal artifact to track project requirements. in terms of testing “Requirements Traceability Matrix (RTM) is a document which is used to validate that all the requirements are linked to test cases”

The basic body of an RTM gives information about:-

- Scope of the Project
- RTM Table
- Comments Section

Design of a Traceability Matrix



Visual Progression: This is where we visualize our goals. Something like why are we making an RTM ? What purpose will it serve ? Will it be used, maintained and documented in the current version? What will happen to the RTM in the next iteration ? Hence visualizing all these, an RTM is designed and edited.

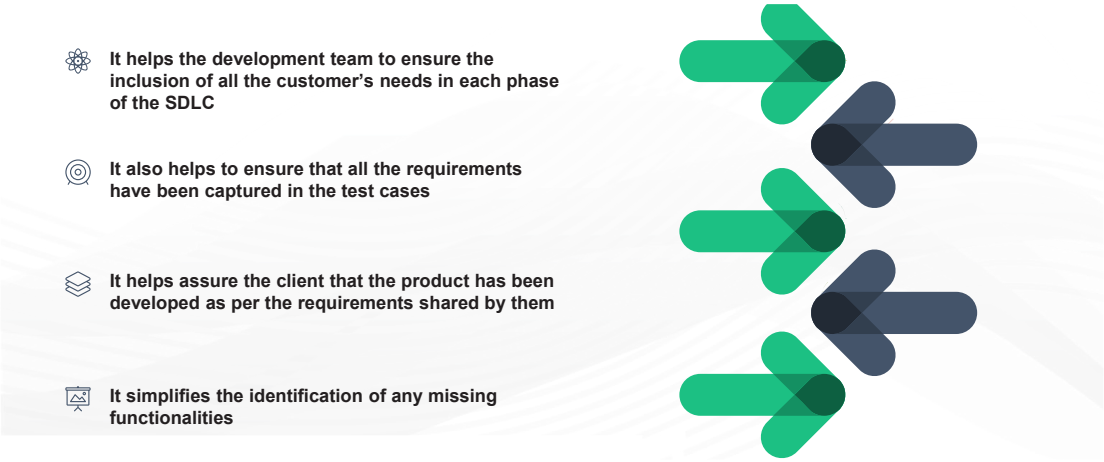
Forward Traceability: Sometimes this is marked within the same RTM artifact. Sometimes it

is a separate artifact. This document is used to map the requirements to the test cases

Backward Traceability: Mapping test cases with requirements in RTM is called Backward Traceability Matrix.This restricts expansion of requirements which have not been in scope of the project.

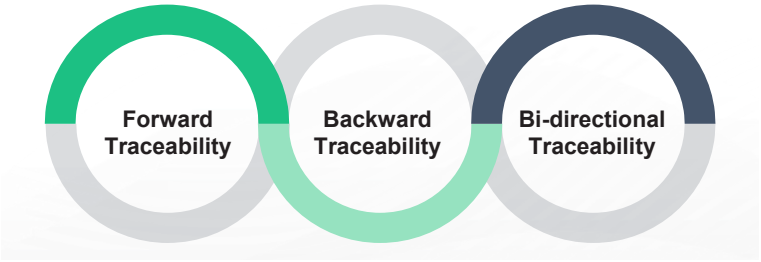
Bidirectional Traceability: When mapping the requirements to test cases for Forward traceability and test cases to requirements for Backward traceability in a single document, it is called a Bidirectional Traceability Matrix. This document will help in ensuring that all the specified requirements have corresponding test cases and vice versa.

Here are some advantages of Traceability Matrix



Traceability Matrix has its unique advantages and generally this also acts as a tracking document but somehow it is not widely used in past half a decade as methodologies in testing have changed and with the fast pace activities this just remains a concept rather than a proper testing artifact.

Type of Traceability Matrix

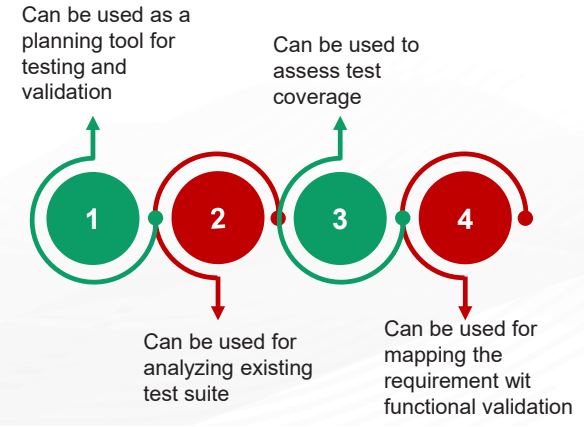


Forward Traceability : Mostly this is marked within the same RTM artifact but sometimes it is a separate artifact. This document is used to map the requirements to the test cases.

Backward Traceability : Also known as reverse traceability, mapping test cases with requirements in RTM is called Backward Traceability Matrix.This restricts expansion of requirements which have not been in scope of the project

Bi-directional Traceability : This type of traceability matrix has both forward and backward traceability. When mapping the requirements to test cases for forward traceability and test cases to requirements for backward traceability in a single document, it is called a Bidirectional Traceability Matrix. This document will help to ensure that all the specified requirements have corresponding test cases and vice versa.

How is it useful in Testing ?



Traceability Matrix is a very important document but with the fast pace Methodologies, slowly this practice is diminishing. It gives quite a lot of information in the beginning related to Forward, Backword and Bi-directional traceability with respect to a project requirements along with the test cases or test scenarios.

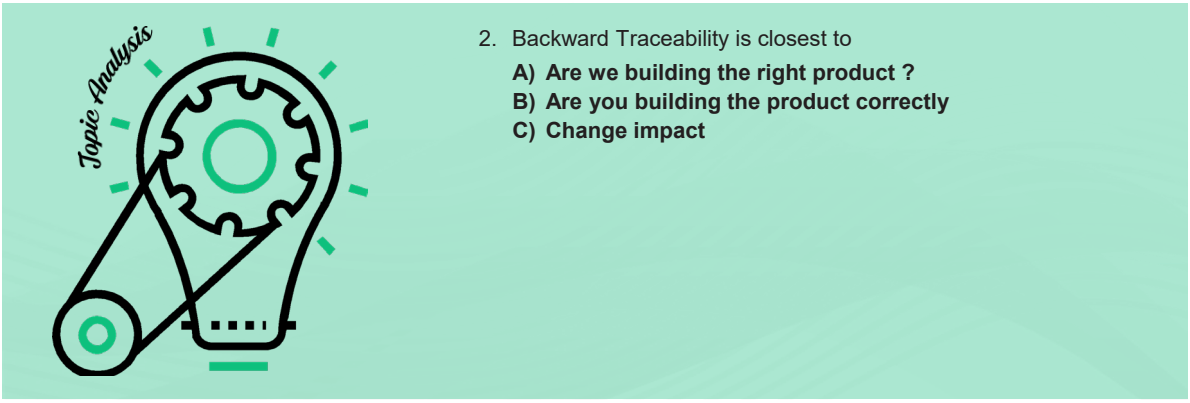
It also gives another round of thorough check between the mapping of requirements and the test cases documented to determine the scope coverage

It also restricts un-necessary editions or additions of furthermore requirements which are not in the definition of scope

What did you grasp ?

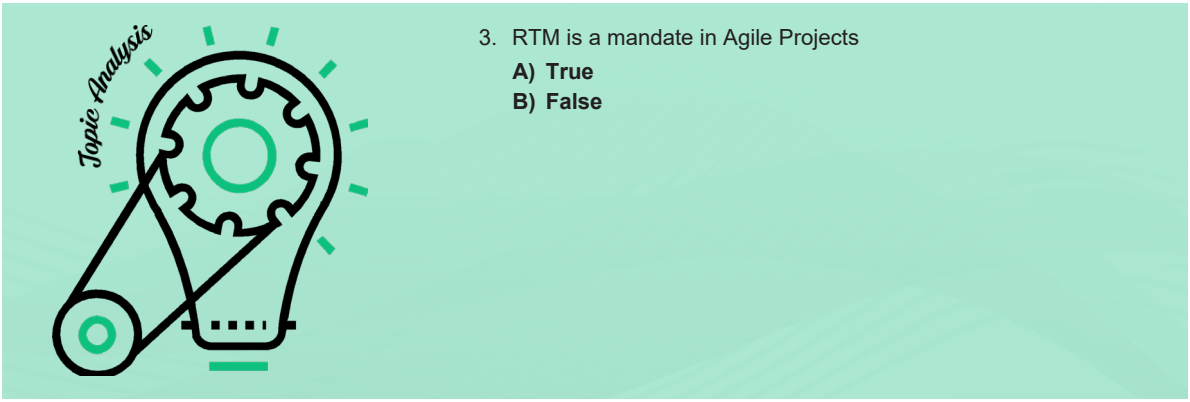
1. Are you building the product correctly is an example of ?
   - **A) Test Plan**
   - **B) forward traceability**
   - **C) backward traceability**

Answer: 1 : (b)



2. Backward Traceability is closest to
   - **A) Are we building the right product ?**
   - **B) Are you building the product correctly**
   - **C) Change impact**

Answer: 2 : (a)



3. RTM is a mandate in Agile Projects
   - **A) True**
   - **B) False**

Answer: 3 : (b)

In a nutshell, we learnt:

1. Test Scenario
2. Test Case Design
3. Test Basis
4. Traceability Matrix

Now, you have reached the end of the module, In this module, you have learned:

1. Test Scenario
2. Test Case Design
3. Test Basis
4. Traceability Matrix

Release Notes

B. TECH CSE with Specialization in DevOps

Semester Six -Year 03

Release Components.
Facilitator Guide, Facilitator Course Presentations, Student Guide, Mock exams and relevant lab guide.

Current Release Version.
1.0.0

Current Release Date.
19 Jan 2020

Course Description.
Xebia, has been recognized as a leader in DevOps by Gartner and Forrester and this course is created by Xebia to equip students with set of practices, methodologies and tools that emphasizes the collaboration and communication of both software developers and other information-technology (IT) professionals while automating the process of software delivery and infrastructure changes.

The language used in this course is US English. Our sources of reference for grammar, syntax, and mechanics are from The Chicago Manual of Style, The American Heritage Dictionary, and the Microsoft Manual of Style for Technical Publications.

Bugs reported	Not applicable for version 1.0.0
Next planned release	Version 2.0.0 Jan 2021