



Manual Testing

Class 4

Agenda

Functional vs Non Functional Testing

Software Testing Hierarchy

Different Types of Testing:

- Regression Test
- Ad hoc Test
- Smoke Test
- 508 Compliance

Whitebox vs Blackbox Testing techniques

What is RTM?

- It is a document that maps and traces user requirement with test cases.
- The main purpose of RTM is to see that all requirements are covered so that no functionality should be missed while testing. RTM is the mapping between requirement to test cases and test cases to the defect. If there is defect found we can trace back to tell which specific requirement failed.
- It is also called requirement coverage, one on one mapping.

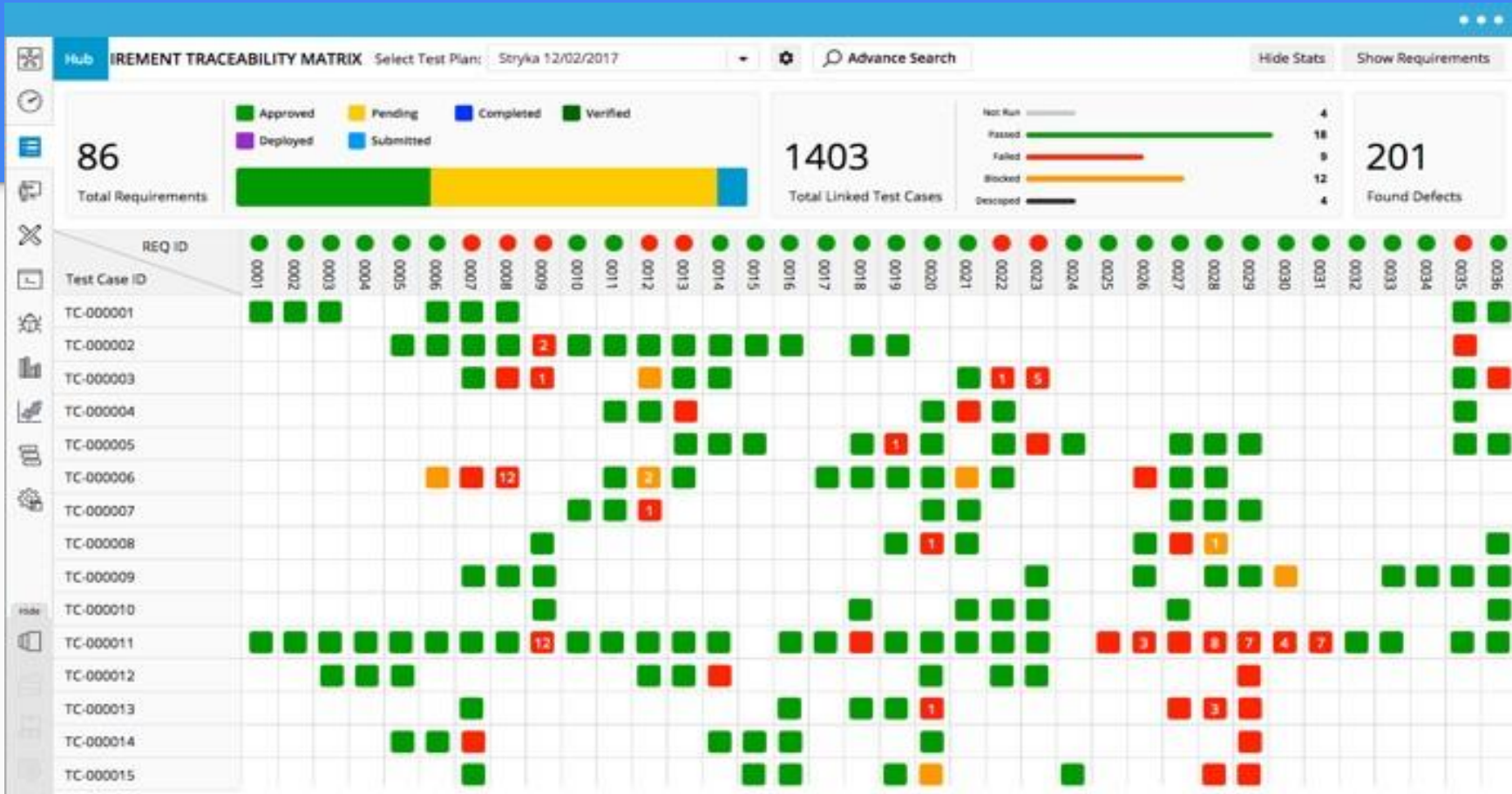
**REQUIREMENT
TRACEABILITY
MATRIX**

RTM Example

Requirement Traceability Matrix

Requirement	Status	Linked Tests	Linked Tests Defects
CALC-671 - Req v4	v3.0 - NOTRUN	CALC-672 - T v4	
CALC-670 - As a user, I can calculate the sum of 2 numbers	v3.0 - UNCOVERED		
CALC-653 - As a user, I can calculate the sum of 2 numbers	v3.0 - OK	CALC-658 - test addition in shell script CALC-657 - Calculate the sum of two numbers CALC-654 - user sums two integer numbers	CALC-667 - problem with 1 button CALC-660 - the calculator crashed
CALC-650 - As a user, I can calculate Requirement Coverage charts based on a given Test Plan	v3.0 - NOTRUN	CALC-652 - T2 CALC-651 - T1	
CALC-640 - As a user, I can calculate the sum of 2 numbers	v3.0 - NOK	CALC-645 - teste soma em shell script CALC-644 - teste automatizado para soma 2 numeros CALC-641 - usuario soma dois numeros CALC-634 - generic automated test	CALC-647 - calculadora crashou ao fazer igual apos soma
CALC-629 - As a user, I can calculate the sum of 2 numbers	v3.0 - NOK	CALC-633 - automatic cucumber addition CALC-630 - calculate the sum of two numbers CALC-324 - Calculate the sum of 2 numbers	CALC-636 - the calculator crashed
CALC-619 - As a user, I can calculate the sum of 2 numbers	v3.0 - OK	CALC-620 - Calculate the sum of 2 numbers	CALC-626 - Def2 CALC-625 - Def1
CALC-614 - As a user, I can calculate the sum of 2 numbers	v3.0 - NOTRUN	CALC-618 - automatic test for the addition CALC-615 - Calculate the sum of 2 numbers	

RTM Example



Two Types of Testing

- Functional Testing
- Non-functional Testing

Functional Testing

Is a type of testing when we test the functionality of the application.

First thing we do when a code or new functionality comes from developers so make sure that the functionality is working fine according to business requirements, and doesn't have any major bugs.

Functional testing is performed by manual testers and by automation team.

Examples: **Can user login? Can user logout?**

Functional testing is designed to determine that the application's features and operations perform the way they should.

Non-Functional Testing

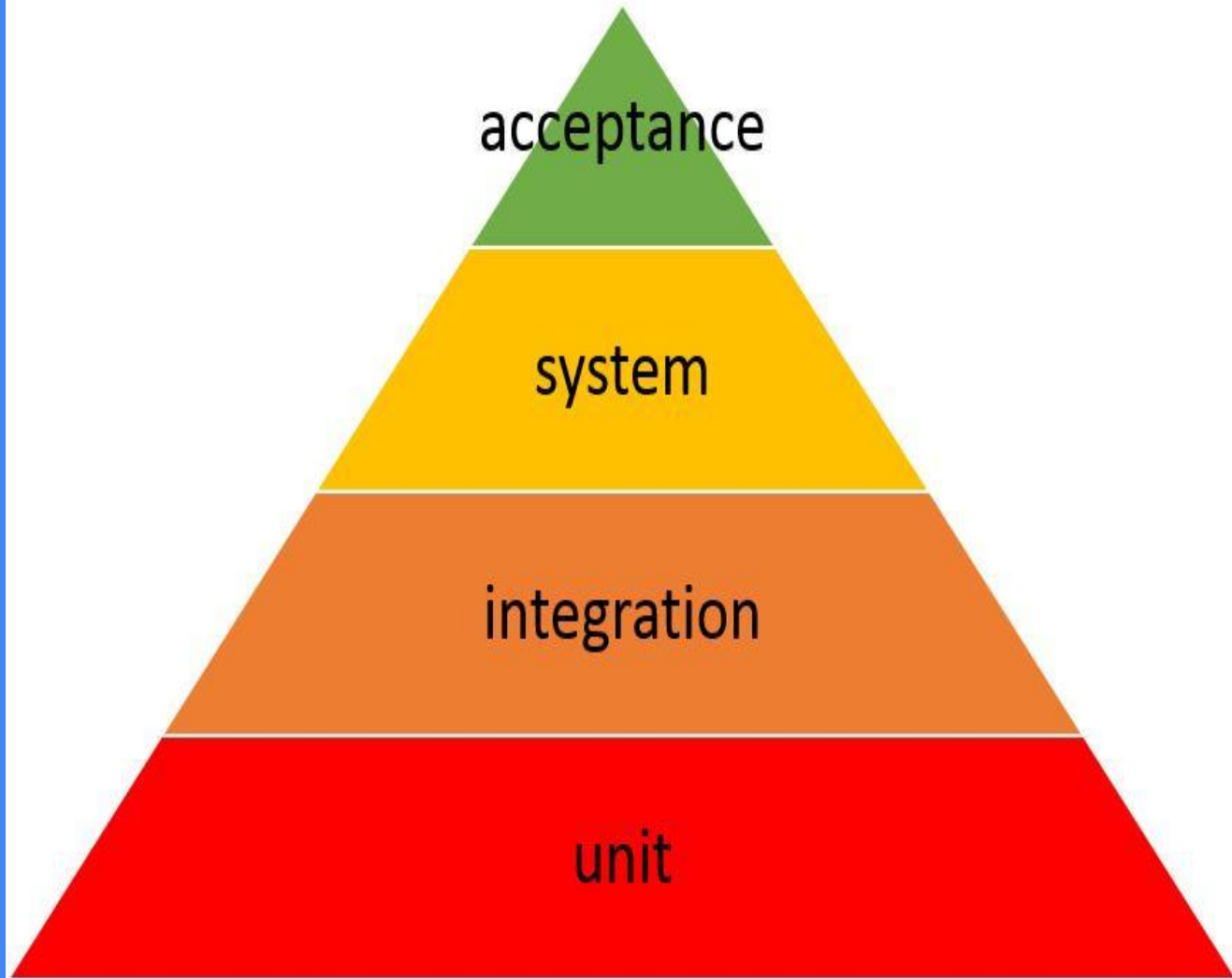
Testing software application or system for its non-functional requirements: The way a system operates, rather than specific behaviors of that system.

Non-functional testing is designed to figure out if your product will provide a good user experience. For example, non-functional tests are used to determine how fast the product responds to a request or how long it takes to do an action (performance testing)

Non-functional testing wants to see if the product stands up to customer expectations.

Look and Feel are also part of non-functional testing

Software Testing Hierarchy



Unit Testing

Unit is smallest testable part of an application like functions, classes etc.

It's part of the white box testing.

It's done by developers. They execute and evaluate their code and if they find any bugs they fix it before it come to QA environment and testers finds it.

As a tester we do not perform unit testing

Developer – DEV Environment

Integration Testing

Validate the interaction of related components, modules and systems to ensure they work together.

Test individual software components to verify interaction between various software components and detect interface defects. After the integration testing has been performed on the components, they are readily available for system testing.

For Ex. Amazon video will work fine as a standalone component. But when you put Amazon video code together with the rest of the website it is no longer working. We need to make sure that all the components when integrated together are able to interact with each other and work properly.

System Testing

System testing is testing complete and fully integrated software product. Everything together like integrated component, Front end, back end, database, server etc.

Test that the behavior and functionality of the complete and fully integrated application complies with specified requirements.

Also called End to End scenario testing.

It also tests overall user experience.

QA Manual/Automation- TEST/SIT Environment

Acceptance Testing

Once the application has completed it is delivered to user or customer and tested by them for acceptance testing.

Acceptance testing is basically done by the user or customer although other stakeholders may be involved as well. Work with subject matter experts and business stakeholders to guide them through the user acceptance testing process, ensuring that the application meets business needs

Also known as UAT User Acceptance Testing.

Companies can also have UAT Team to perform User Acceptance Testing. Once QA team and Dev team completes the testing then UAT team tests the product. UAT team assures that the product works fine, as expected before it goes to business.

USERS - STAGE/UAT Environment

Different testing types

There are different testing types that we as testers can perform

- Smoke
- Regression
- Performance
- Ad Hoc

Smoke Test

A smoke test is a quick, lightweight test or set of tests, usually automated, that confirm that the basic functionality of system is working correctly. It tends to emphasize broad, not deep tests, and is usually done before launching a more extensive set of tests.

Smoke test quickly scans the application through multiple high level scenarios under 30 mins to make sure that the application is working fine (Environment is UP or Down).

If something does not work, it sends out an email to everyone, letting them know that smoke test did not pass. This is mostly done through automated script.

Automated Smoke test can be scheduled multiple times a day to give timely report to the team.

Regression Testing

- Regression testing is a software testing practice that ensures an application still functions as expected after any code changes, updates, or improvements.
- Regression testing is responsible for the overall stability and functionality of the existing features. Whenever a new modification is added to the code, regression testing is applied to guarantee that after each update, the system stays sustainable under continuous improvements.
- Regression testing is performed before every application release.
- Regression test cases are perfect for test automation because it is repetitive and very time consuming.

Regression Testing

When to apply regression testing:

- A new requirement is added to an existing feature
- A new feature or functionality is added
- A modifications/changes been applied to the existing feature
- The codebase is fixed to solve defects
- The source code is optimized to improve performance

Performance Testing



Performance testing, a non-functional testing technique performed to determine the system parameters in terms of responsiveness and stability under various workload.

It measures the quality attributes of the system, such as scalability, reliability and resource usage etc.

Example: Can 2000 user login to the application at the same time? Can user move to the next page in 1 second?

508 Compliance Testing

Section 508 is a requirement for government websites, and in general, for all federal departments and agencies to ensure that their electronic information & technology (EIT) is accessible to people with disabilities.

508 Compliance testing is to make sure disable people can use it as well.

Mandatory for government websites.

Popular tool used for 508 testing is JAWS(Job Access With Speech)



Ad-Hoc Testing

Testing performed without planning and documentation.

Informal testing OR least formal test method.

A lot of Bugs can be find in ad-hoc testing.

Performed mostly after official testing is done and still sometime is left for testing the application.

It can be also called Exploratory testing, Random Testing

Retesting

Regression testing is done to find out the issues which may get introduced because of any change or modification in the application. The purpose of regression testing is that any new change in the application should NOT introduce any new bug in existing functionality.

Verification of bugs are During retesting only failed test cases are re-executed. Retesting is carried out to ensure that the original issue is working as expected. Retesting is executed in the same environment with same data but in new build.

Different testing techniques

There are different testing techniques could be used such as:

- White box testing
- Black box testing
- Positive testing
- Negative Testing
- Boundary Value analysis

White-box testing techniques

Requires detailed programming skills.

Mostly done by developers where they execute their code before it goes to QA. environment like unit testing

Internal structure is known to the Developer who is going to test the software.

Black-box testing techniques

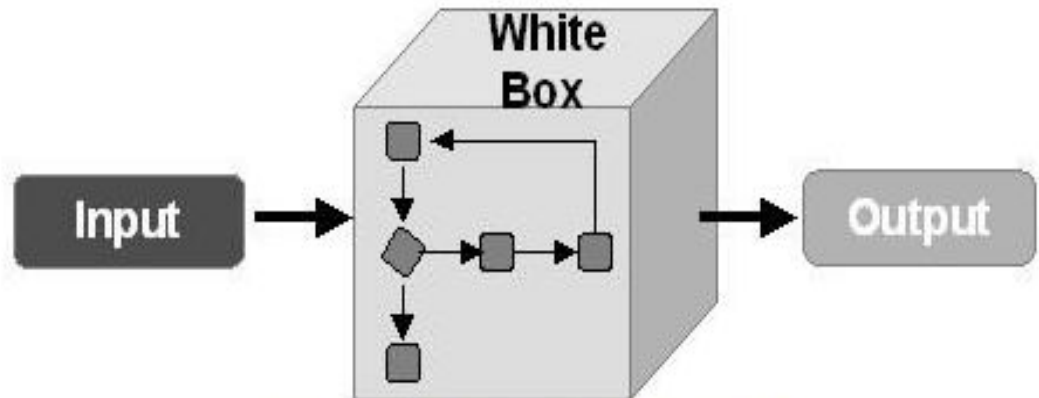
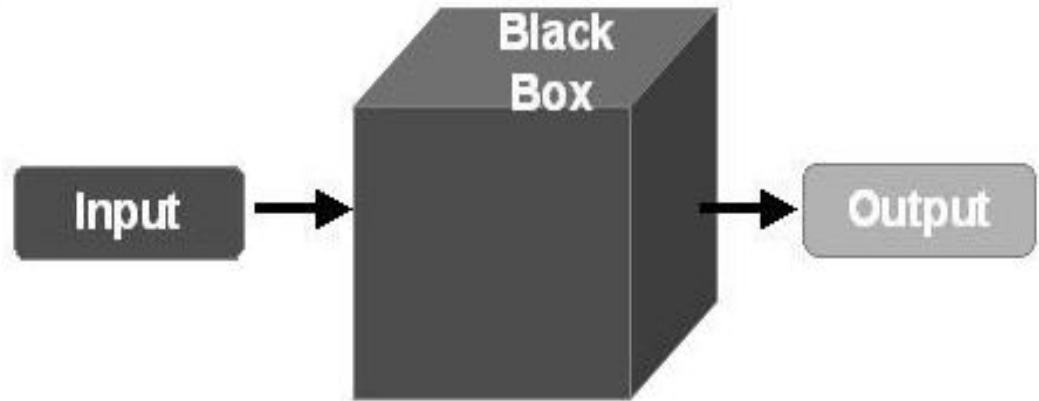
Requires no knowledge of internal paths, structures, or implementation of the software being tested.

Tester doesn't know what is inside.

All the tester knows is what the software is supposed to do and if the software is doing its job correctly

Black-box testing only cares about input / output

White-box v.s Black-box



Positive Testing

Testing the application with valid input.

Example: If you are doing positive testing for buying functionality, you would use only valid credit card number and expiration date.

Enter Only Numbers

999999

Positive Testing

Negative Testing

Test the application with invalid data as input.

Checks whether application behaves as expected with invalid / negative input.

Test the application to see that it does not do anything that it is not supposed to do.

Example: Login with invalid username and password. You go to ATM today and try to take 1000 dollars out or you can put in Costco card and see what happens. Give your own ...

Enter Only Numbers

Negative Testing

Boundary Value Analysis

Boundary testing is to make sure the software accepts valid data inside the valid Boundary (Lower

Boundary and Higher Boundary) and rejects invalid data outside the valid Boundary.

Example: Tax return period is from January 1st until April 15th. If you spend less than \$50 and amazon.com still did not charge you for shipping, it means there is a defect in the application. Many defects can be found with this method.

