

Assignment - 2

- What is Exploratory Testing?

- ➔ Exploratory testing is a concurrent process where test design, execution and logging happen simultaneously, testing is often not recorded, makes use of experience, heuristics and test patterns.
- ➔ It is not random testing but it is adhoc testing with purpose of find bugs and is structured and rigorous.

- What is traceability matrix?

- ➔ To protect against changes you should be able to trace back from every system component to original requirement that caused its presence.
- ➔ Types of RTM:
 1. Forward Traceability – mapping of req. to test cases
 2. Backward Traceability- mapping of test cases to requirements
 3. Bi-Directional Traceability – A good traceability matrix is the references from test cases to basis documentation and vice versa.
- ➔ Pros: => easy to identify the missing functionalities.

=>to make sure that all requirements included in test cases

=>If there is a change request for a requirement, then we can easily find out which test cases need to update.

=> To make obvious to the client that the software is being developed as per the requirements.

-> Cons: => Poor or unknown test coverage, more defects found in production.

=> Difficult project planning and tracking, misunderstandings between different teams over project dependencies, delays, etc.

- What is Boundary value testing?

- ➔ BVA is a methodology for designing test cases that concentrates software testing effort on cases near the limits of valid ranges.
- ➔ BVA is method which refines equivalence partitioning.
- ➔ Its generates test cases that highlight errors better than E.P.
- ➔ At those points when input values change from valid to invalid errors are most likely to occur.

- What is Equivalence partitioning testing?

- ➔ Aim is to treat group of input as equivalent and select one representative input to test them all.
- ➔ EP can be used for all levels of testing.
- ➔ EP always one positive scenario and two negative scenario.

- What is Integration testing?

- ➔ Testing performed to expose defects in the interactions between integrated components or systems.
- ➔ Integration testing is a level of the software testing process where individual units are combined and tested as a group.
- ➔ Two level of Integration testing :
 1. Component integration testing.
 2. System integration testing.
- ➔ CIT: testing performed to expose defects in the interfaces and interaction between integrated components.
- ➔ SIT: it tests the interactions between different systems and may be done after system testing.

- What determines the level of risk?

- ➔ It's depends on below evaluated points:
 1. Business levels.
 2. Technological level
 3. Project level
 4. Testing level – its depend on the where to start testing and where more testing is needed
 5. Financial cost of software
 6. Potential loss of life like safety systems
 7. Potential loss of face(improper implementations)
 8. Late delivery to the market

- What is Alpha testing?

- ➔ It is always performed by the developers at the software development site.
- ➔ It is always performed in virtual environment.
- ➔ It is not open to the market and public.
- ➔ It is also performed by the independent testing team.
- ➔ It is conduct for software application and project.
- ➔ It is always performed within the organization.
- ➔ It's come under the category of both white box testing and black box testing.

- What is beta testing?

- ➔ It is always performed by the customers at their own site.
- ➔ It is not performed by independent testing team.
- ➔ It is performed in real time environment.
- ➔ It is also the form of acceptance testing.
- ➔ It is only a kind of black box testing.
- ➔ It is usually conducted for software product.

➔ It is always performed outside the organization.

- What is component testing?

- ➔ The testing of individual software components.
- ➔ It is the first level of testing and is performed prior to integration testing.
- ➔ It's also known as unit testing, module testing or program testing.
- ➔ Unit testing is performed by using the white box testing method.
- ➔ Unit testing frameworks, drivers, stubs and fake objects are used to assist in unit testing.
- ➔ Unit testing are typically written and run by software developers to ensure that code meets its design and behaves as intended with debugging tool.
- ➔ Unit tests find problems early in the development cycle.

What is functional system testing?

- ➔ Testing based on an analysis of the specification of the functionality of a component or system
- ➔ Functional testing verifies that each function of the software application operates in conformance with requirement specifications.
- ➔ This testing mainly involves black box testing and it not concerned about the source code of the application.
- ➔ Type of functional testing:
 1. Unit testing
 2. Smoke testing
 3. Sanity testing
 4. Regression testing
 5. Integration testing
 6. Black box testing
 7. White box testing
 8. User acceptance testing

- What is Non-Functional Testing?

- ➔ Testing the performance, reliability, scalability and other non-functional aspects of the software system.
- ➔ It performed after functional testing
- ➔ It's describe how good the product works.
- ➔ Using tools will be effective for this testing.
- ➔ Type of non- functional testing:
 1. Performance testing
 2. Load testing
 3. Stress testing
 4. Security testing
 5. Scalability testing
 6. Migration testing
 7. Stress testing

8. Volume testing
9. Compatibility testing

- What is GUI Testing?

- ➔ GUI testing is the process of testing the system's GUI of the system under test. In this we involves checking the screens with the controls like menus, buttons, icons, and all types of bars – toolbar, menu bar, dialog boxes and windows etc.
- ➔ Common aspects verified in GUI testing:
 1. Layout
 2. Style and aesthetics
 3. Responsiveness
 4. Behaviour
 5. Error messages
- ➔ Type of GUI testing
 1. Manual testing
 2. Automated Testing
 3. Cross-platform testing

- What is Adhoc testing?

- ➔ It is informal testing type with an aim to break the system.
- ➔ It does not follow any test design techniques to create test cases.
- ➔ In fact it does not create test cases altogether.
- ➔ This testing is primarily performed if the knowledge of testers in the system under test is very high.
- ➔ Main aim of this testing is to find defects by random checking.
- ➔ Ad-hoc testing can be achieved with the testing technique called error guessing.
- ➔ Types of ad-hoc testing
 1. Buddy testing
 2. Pair testing
 3. Monkey testing

- What is load testing?

- ➔ Load testing is to test the system behaviour under normal workload conditions, and it is just testing or simulating with the actual workload.
- ➔ Load testing does not break the system
- ➔ Load testing identifies the bottlenecks breaking the system under various workload and checks how the system reacts when the load is gradually increased.
- ➔ Load testing gives confidence in the system & its reliability and performance.

- What is stress Testing?

- ➔ Stress testing is to test the system behaviour under extreme conditions and is carried out till the system failure.
- ➔ In the stress testing determines the point of the system to reveal the maximum point after which it breaks.
- ➔ Stress testing tries to break the system by testing with overwhelming data or resources.
- ➔ Stress testing is done in order to check when the application fails by reducing the resources such as RAM, HDD etc. and also the keeping the number of users constant.

- What is white box testing and list the types of white box testing?

- ➔ Testing based on an analysis of the internal structure of the component or system.
- ➔ Structure-based testing technique is also known as white-box testing or glass box testing technique because here the testers require knowledge of how the software is implemented, how it works.
- ➔ White box testing done by the developer ends.
- ➔ Types of white box testing:
 1. Statement coverage
 2. Decision coverage
 3. Condition coverage

- What is black box testing? What are the different black box testing techniques?

- ➔ Black box testing either functional or non-functional, without reference to the internal structure of component or system.
- ➔ Testers have no knowledge of how the system or component is structured inside the box.
- ➔ In black box testing the tester is concentrating on what the software does, not how it does it.
- ➔ In this testing tester does not require coding knowledge.
- ➔ Type of Black box testing techniques:
 1. Equivalence partitioning
 2. Boundary value analysis
 3. Decision tables
 4. State transition testing
 5. Use-case testing

- Mention what are the categories of defects?

- ➔ Functional defects or Critical functional defects
- ➔ Performance defects
- ➔ GUI Defects
- ➔ Security defects
- ➔ Logical defects
- ➔ Base on lifecycle phase defects like requirement defects, design defects, coding defects, testing defects, deployment defects etc.

➔ Database defects

- Mention what big-bang testing is?

- ➔ Big-bang testing is one type of integration testing.
- ➔ In this testing all components or modules is integrated simultaneously, after which everything is tested as a whole.
- ➔ In this first we integrated all modules and then after we tested.
- ➔ It's convenient for small systems.
- ➔ Main problem is fault localization is difficult.

- What is the purpose of exit criteria?

- ➔ Successful testing of integrated application.
- ➔ Executed test cases are documented.
- ➔ All high prioritized bugs fixed and closed.
- ➔ Technical documents to be submitted followed by release notes.

- When should "Regression Testing" be performed?

- ➔ In this testing of a previously tested program following modification to ensure that defects have not been introduced or uncovered in unchanged area of the software, as a result of the changes made. It is performed when the software or its environment changed.
- ➔ When we use Regression testing :
 - =>Change in requirements and code is modified according to requirement.
 - =>New feature is added to the software
 - => Defect fixing
 - => Performance issue fix

- What is 7 key principles? Explain in detail?

- ➔ Testing principles :
 1. Testing shows presence of defects:
 - Testing shows the defect are present, but cannot prove that there are no defects.
 - Testing reduce the probability of undiscovered defects remaining in the software but, even if no defect are found, it is not a proof of correctness.
 - However testing cannot prove that there are no defect present.
 2. Exhaustive testing is impossible
 - Testing everything including all combinations of inputs and preconditions is not possible
 - We don't have too much time to check all modules in the software or project so we must priorities our testing effort using a risk based approach.
 3. Early testing
 - Testing activities should start as early as possible in the software or system development life cycle, and should be focused on defined objectives.
 4. Defect clustering

- A small number of modules contain most of defects discovered during pre-release testing, or are responsible for the most operational failures.
 - Defects are not evenly spread in system
 - They are clustered.
5. The pesticide paradox
- If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new defects.
 - In this stage we need to be regularly reviewed and revised, and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects.
6. Testing is context dependent
- Testing is basically context dependent.
 - Testing is done differently in different contexts
 - Different kinds of sites are tested differently
7. Absence of errors fallacy
- If the system built is unusable and does not fulfil the user's needs and expectations then finding and fixing defects does not help.
 - The defect have been resolved it may still be unusable and does not fulfil the user's needs and expectations.

• Difference between QA v/s QC v/s Tester

	QA	QC	Testing
1	Activities which ensure the implementation of process, procedures and standards in context to verification of developed software and intended requirements.	Activities which ensure the verification of developed software with respect to documented requirement.	Activities which ensure the identification of bug/error/defects in the software.
2	Focus on processes and procedures rather than conducting actual testing on the system.	Focus on the actual testing by executing software with intend to identify bug through implementation of procedures and process.	Focuses on actual testing.
3	Process oriented activities.	Product oriented activities.	Product oriented activities.
4	Preventive process	Corrective process	Preventive process.
5	It is subset of software test life cycle	QC can be considered as the subset of QA.	Testing is the subset of quality control.

- Difference between Smoke and Sanity?

	Smoke Testing	Sanity Testing
1	It's performed only the critical functionalities of program is working fine.	It is done to check the new functionality/bugs have been fixed
2	It is performed by the developer or tester	It is performed by the tester
3	This testing usually documented or scripted.	This testing usually not document and unscripted.
4	This is subset of regression testing	This is subset of Acceptance testing.
5	Smoke testing is exercises the entire system from end to end.	Sanity testing exercises only the particular component of the entire system.

- Difference between verification and Validation

	Verification	Validation
1	The process of evaluating work-products of a development phase to determine whether they meet the specified requirements for that phase.	The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements.
2	To ensure that the product is being built according to the requirements and design specifications.	To ensure the product actually meets the user's needs, and that the specifications were correct in the first place
3	Activities- Review, walkthroughs, Inspections	Activities- Testing
	Evaluation of plans, Requirement specification, design, coding, test cases.	Evaluate the actual product/software.

- Explain types of Performance testing.

- ➔ Performance testing is a meaning of quality assurance. It involves the testing software application to ensure they will perform well under their expected workload.
- ➔ Type of performance testing.
 1. Load testing
 2. Stress testing
 3. Endurance testing
 4. Spike testing
 5. Volume testing
 6. Scalability testing

- What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?

Aspect	SDLC (Software Development Life Cycle)	STLC (Software Testing Life Cycle)
Purpose	Focuses on the overall development of the software.	Focuses on testing to ensure software quality and defect identification.
Scope	Covers all phases of software creation: planning, design, coding, testing, deployment, and maintenance.	Covers only the phases related to software testing.
Phases	1. Requirement analysis 2. Feasibility study 3. System design 4. Coding 5. Testing 6. Deployment 7. Maintenance	1. Requirement analysis (from a testing perspective) 2. Test planning 3. Test case development 4. Test environment setup 5. Test execution 6. Test closure
Participants	Involves business analysts, developers, designers, testers, and project managers.	Primarily involves testers and QA analysts, with occasional developer support.
Outcome	Results in a functional and deployable software product.	Results in validated, verified, and defect-free software.
Relationship	A broader framework that encompasses STLC as one of its phases.	A subset of SDLC focused specifically on quality assurance.
Main Deliverables	Software product with documentation, code, and user manuals.	Test plans, test cases, test scripts, defect reports, and test summary reports.

- What is the difference between test scenarios, test cases, and test script?

Aspect	Test Scenario	Test Case	Test Script
Definition	A high-level idea or description of what to test.	A detailed step-by-step document that defines how to test a specific functionality.	A set of instructions written in code or a scripting language to automate a test case.
Detail Level	High-level, broad outline.	Medium-level, includes specific inputs, actions, and expected outcomes.	Low-level, precise, and machine-executable.

Aspect	Test Scenario	Test Case	Test Script
Purpose	Ensures coverage of all major features and flows.	Validates specific aspects of functionality.	Automates repetitive or complex testing processes.
Who Uses It?	Test managers, QA leads for planning.	Testers during manual testing.	Test automation engineers for automated testing.
Format	Informal, written in plain language.	Formal, structured format with predefined sections like ID, steps, expected result, etc.	Written in programming or scripting languages (e.g., Java, Python, JavaScript).
Tool Dependency	No tools required.	Usually documented in test management tools like JIRA, TestRail.	Requires automation tools like Selenium, Appium, or UFT.
When Used?	During the initial stages of test planning.	During the execution phase of manual testing.	During the execution phase of automated testing.

- Explain what Test Plan is? What is the information that should be covered?

➔ A document describing the scope, approach, resources and schedule of intended test activities.

Key information in a plan

1. Test objectives
2. Test scope
3. Test strategy
4. Entry and exit criteria
5. Test deliverables
6. Test environment
7. Role and responsibilities
8. Risk and mitigation

- What are the different Methodologies in Agile Development Model?

1. Scram
2. Kanban
3. Extreme programming(XP)

Scram frameworks:

- ➔ Product backlog
- ➔ Sprint planning
- ➔ Sprint backlog
- ➔ Daily scram
- ➔ Sprint retrospective
- ➔ Sprint review

