1. The Fundamentals of Testing

A. Testing objectives

- Find and prevent defects
- Reduce risk of software failure in operation
- Gain confidence about level of quality
- Provide information for decision making
- To provide sufficient information to stakeholders to allow them to make informed decisions, especially regarding the level of quality of the test object.
- To reduce the level of risk of inadequate software quality (e.g., previously undetected failures occurring in operation)
- To comply with contractual, legal, or regulatory requirements or standards, and/or to verify the
- test object's compliance with such requirements or standards

B. Quality Assurance and Testing

Quality Management ties testing and quality assurance together. Includes both quality assurance and quality control.
Quality Control involves various activities, including test activities, that support the achievement of appropriate levels of quality. Test activities are part of the overall software development or maintenance process. Since quality assurance is concerned with the proper execution of the entire process, quality assurance supports proper testing
Quality Assurance is typically focused on adherence to proper processes, as to provide confidence that the appropriate levels of **quality** will be achieved (i.e. processes to prevent possible bugs during software development). When processes are carried out properly, the work products created by those processes are generally of higher quality, which contribute to defect prevention. In addition, the use of root cause analysis to detect and remove the causes of defects, along with the proper application of the findings of retrospective meetings to improve processes, are important for effective quality assurance.
Testing is a way of checking that the was the system operates and find the possible defects. Testing enables customers with the possibility to see if the developed product meets their expectations on its design, compatibility, functioning, etc.

	Quality Assurance	Testing
Definition	Activities designed to ensure the software corresponds to the correct specification	The process of exploring a system to find defects
Focus	Quality Control and meeting the requirements	System inspection and bug finding
Orientation	Process-oriented	Product-oriented
Activity Type	Preventative	Corrective
Aim	Assure Quality	Control Quality

C. Debugging vs testing:

Debugging:
Carried out by DEVELOPER Identifies and CORRECTS bugs in code

Testing:
Carried out by TESTER reporting DEFECTS

D. Errors, defects, failures:

A person can make an **error** (mistake), which can lead to the introduction of a **defect** (fault or bug) in the software code or in some other related work product.. If a defect in the code is executed this can lead to a **failure**.

E. Root Cause analysis

By identifying the defect early in the life cycle, it is a lot easier to identify why it was there in the first place.

F. 7 principles of testing:

1. **Testing shows the presence of bugs, not absence of bugs**
2. **Exhaustive testing is impossible**
3. **Early Testing**
4. **Defect Clustering**
5. **The pesticide paradox**
6. **Testing is context dependant**
7. **Absence of errors fallacy**

G. Test basis, conditions, cases, procedures & suite:

Test Basis: The body of knowledge used as the basis for test analysis and design
Test Conditions: An aspect of the test basis that is relevant in order to achieve specific test objectives
Test Case: A set of pre-conditions, inputs, actions (where applicable), expected results and post conditions, developed based on test conditions
Test Procedure: A sequence of test cases in execution order, and any associated actions that may be required to set up the initial pre-conditions and any wrap up activities post execution. Also referred to as 'test script'.
Test Suite: A set of test scripts or test procedures to be executed in a specific test run.

H. Testing's contributions to success

- Having testers involved requirements review allows early detection of defects in work products.
- Having testers work closely with system designers while the system is being designed can increase each party's understanding of the design and how to test it.
- Having testers work closely with developers while the code is under development can increase each party's understanding of the code and how to test it.
- Having testers verify and validate the software prior to release can detect failures that might otherwise have been missed.

I. The Psychology of Testing

Identifying defects during a static test such as a requirements review or user story refinement session, or identifying failures during dynamic test execution, may be perceived as criticism of the product and of its author. An element of human psychology called confirmation bias can make it difficult to accept information that disagrees with currently held beliefs.

J. Traceability between Test Basis and Test Work Products

For effective test monitoring and control (TMC), it is important to establish and maintain traceability throughout the test process (See section K) between each element of the test basis and the various test work products associated with that element. In addition to the evaluation of test coverage, good traceability supports:

- Analysing the impact of changes
- Making testing auditable
- Meeting IT governance criteria
- Improving the understandability of test progress reports and test summary reports to include the
- status of elements of the test basis (e.g., requirements that passed their tests, requirements that failed their tests, and requirements that have pending tests)
- Relating the technical aspects of testing to stakeholders in terms that they can understand
- Providing information to assess product quality, process capability, and project progress against business goals

K. The Test Process

There is no one universal software test process, but there are common sets of test activities without which, testing will be less likely to achieve its established objectives.

Test planning (TP)

Define the objectives of testing and the approach that will be taken:
Specifying suitable test techniques and tasks
Who does what
Creating test schedule for meeting a deadline
Define Exit Criteria (test completion criteria)
Work Products:
Test Plans
Test schedule

Test Monitoring & Control (TMC)

Monitoring: Involves the on-going comparison of actual progress against the test plan using any test monitoring metrics defined in the test plan
Control: Any necessary action is taken if there are any deviations (e.g. more tests may be needed).
Supported by the evaluation of exit criteria (definition of done).

- Looking at exit criteria for test execution for a stage of testing may include:
- Check test results to see if the required test coverage has been achieved
- Determining the component or system quality, by looking at both test results test logs
- Seeing if more tests are required (for example, if there are not enough tests to reach the level of risk coverage that is required).

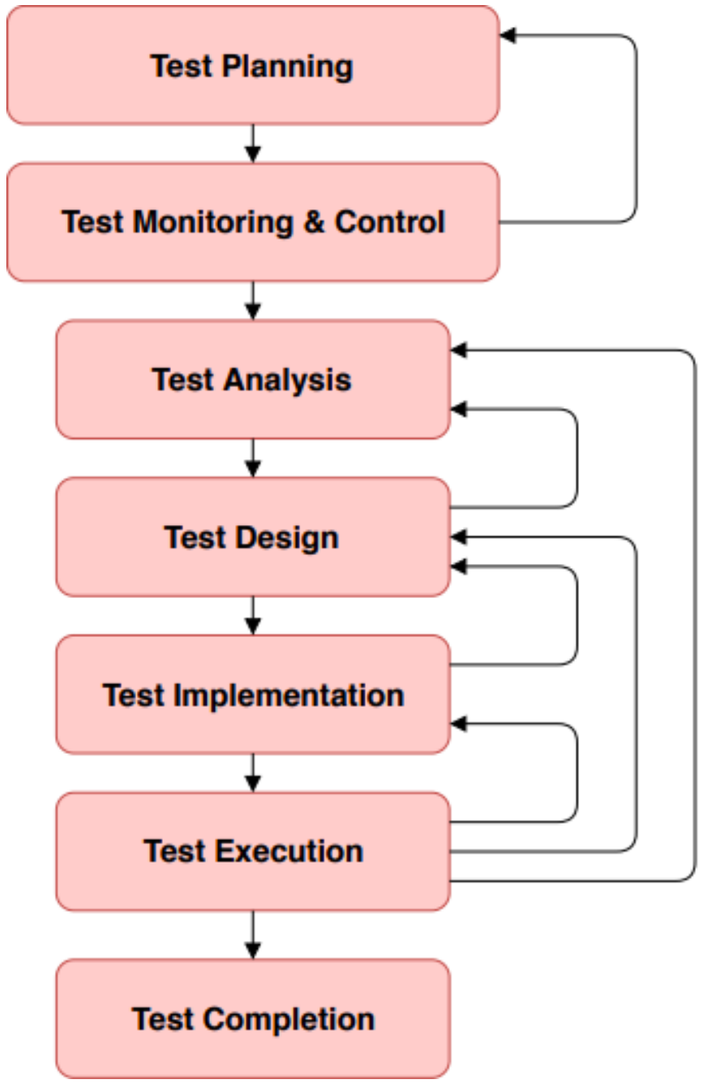
Work Products:
Test Progress Reports
Test Summary Reports

Test Analysis (TA)

"What is going to be tested?"
Analyse test basis to identify testable features and define associated test conditions (i.e. what are we going to test in terms of measurable coverage criteria).
Sometimes, test conditions produced as part of test analysis are used as test objectives in test charters.
Work Products:
Test Conditions (prioritised)
Test Charters

Test Design (TD)

"How to test?"
Design and prioritise tests cases
Identify test data to support said test cases (derived from test conditions)
Identify test data to be used with test cases
Design test environment.
Capture bi-directional traceability between the test basis, test conditions, test cases and test procedures.
Test conditions are further refined.
Work Products:
Test cases and sets of test cases to exercise the test conditions defined in test analysis.
Test data design/identification
Test environment design



Test Implementation (TI)

"Is everything in place so that we are able to test?"
Test design and test implementation tasks are often combined.
Creating and prioritising test procedures
Creating test suites from the test procedures
Arranging the test suites within a test execution schedule
Building the test environment
Preparing test data
Building test environment
Checking and updating the bidirectional traceability between the test basis, test conditions, test cases, test procedures and now to include test suites.
Work Products:
Test procedures and the sequencing of those test procedures
Test suites
Creation and verification of the test environment and test data
A test execution schedule

Test Execution (TE)

Executing tests and logging results.
Record the identification of what is being tested e.g. test items or test objects etc)
Comparing actual with expected results.
Logging and reporting defects
Analysing anomalies to establish their likely causes
Retest and/or regression test when defect has been corrected
Repeating test activities either as a result of action taken for an anomaly, or as part of the planned testing
Work Products:
Documentation of the status of individual test cases or test procedures (e.g., ready to run, pass, fail, blocked, deliberately skipped, etc)
Defect reports Documentation.

Test Completion (TC)

Collect data from completed test activities
Finalising and archiving the test environment, the test data, the test infrastructure, and other testware for later reuse
Ensure defect reports are closed as necessary
Analyse discrepancies to determine their cause
Handing over the testware to the maintenance teams
Lessons learned (improve future tests!)
Work Products:
Test summary reports (for project stakeholders)
Actionable items for improvement in subsequent projects (Lessons Learnt)
Finalised testware