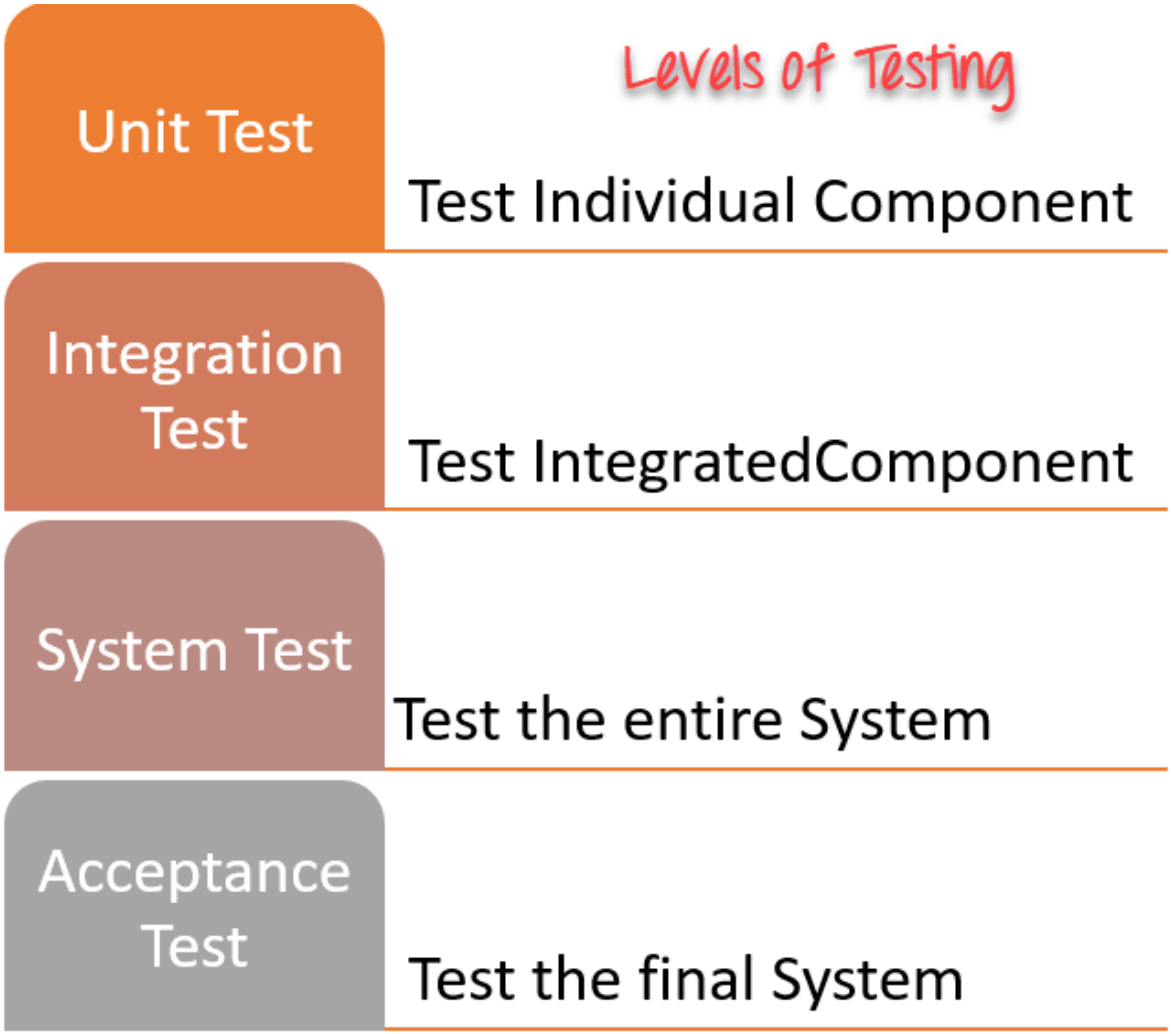
TESTING LEVELS



LEVELS OF TESTING

1	Unit Testing	Done by Developers
2	Integration Testing	Done by Testers
3	System Testing	Done by Testers
4	Acceptance Testing	Done by End Users

Testing LEVELS

- **Unit testing:**
  - test the functionalities of a single module or function.
- **Integration testing:**
  - test the interfaces among the modules.
- **System testing:**
  - test the fully integrated system against its functional and non-functional requirements.

Aceptance Testing:

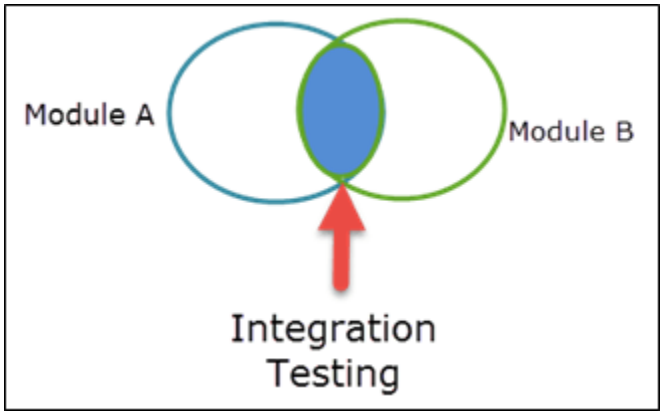
- **Test whether the application is accepted by the users or not**

UNIT TESTING

- **UNIT TESTING** is a type of software **testing** where individual units or components of a software are tested.
- The purpose is to validate that each **unit** of the software code performs as expected.
- **Unit Testing** is done during the development (coding phase) of an application by the developers.

Integration testing

- After different modules of a system have been coded and unit tested:
  - modules are integrated in steps according to an integration plan
  - partially integrated system is tested at each integration step.



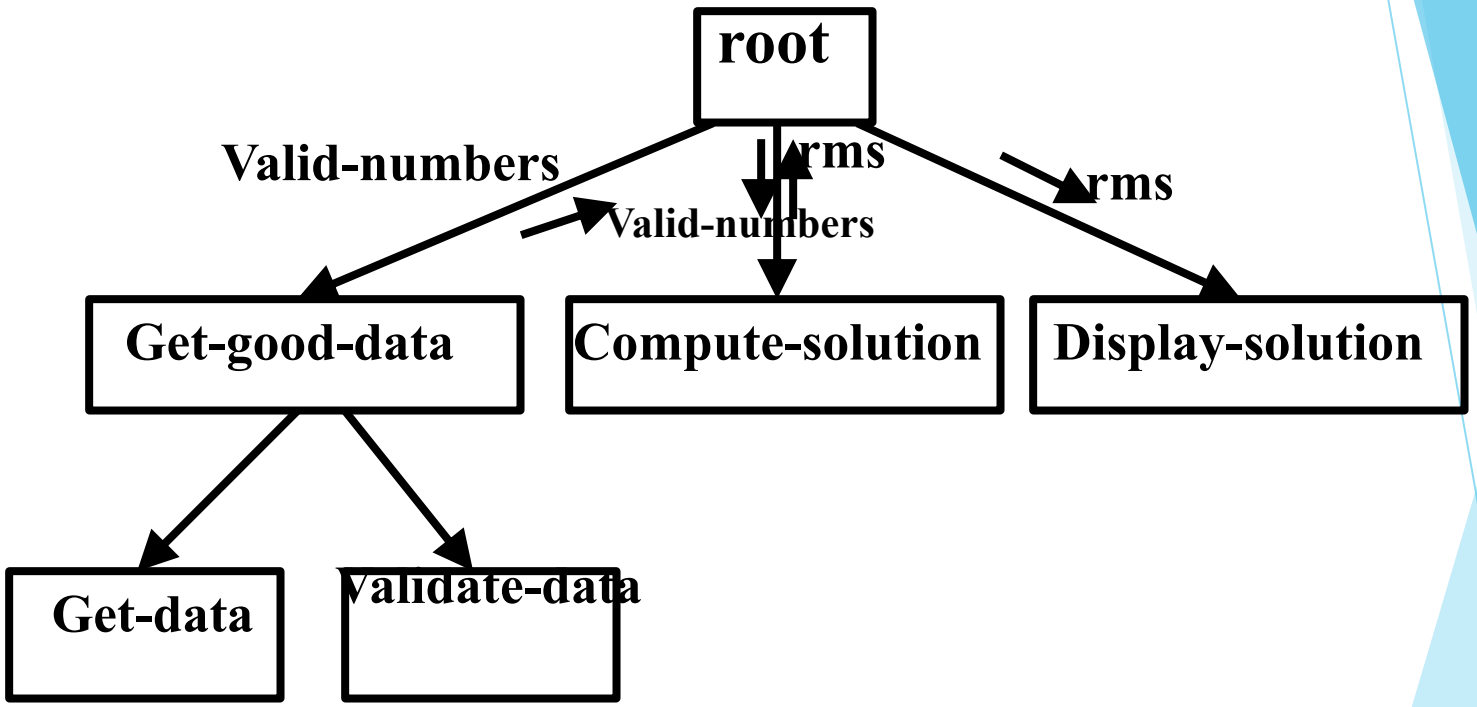
System Testing

- System testing:
  - validate a fully developed system against its requirements.

Integration Testing

- Develop the integration plan by examining the structure chart :
  - big bang approach
  - top-down approach
  - bottom-up approach
  - mixed approach

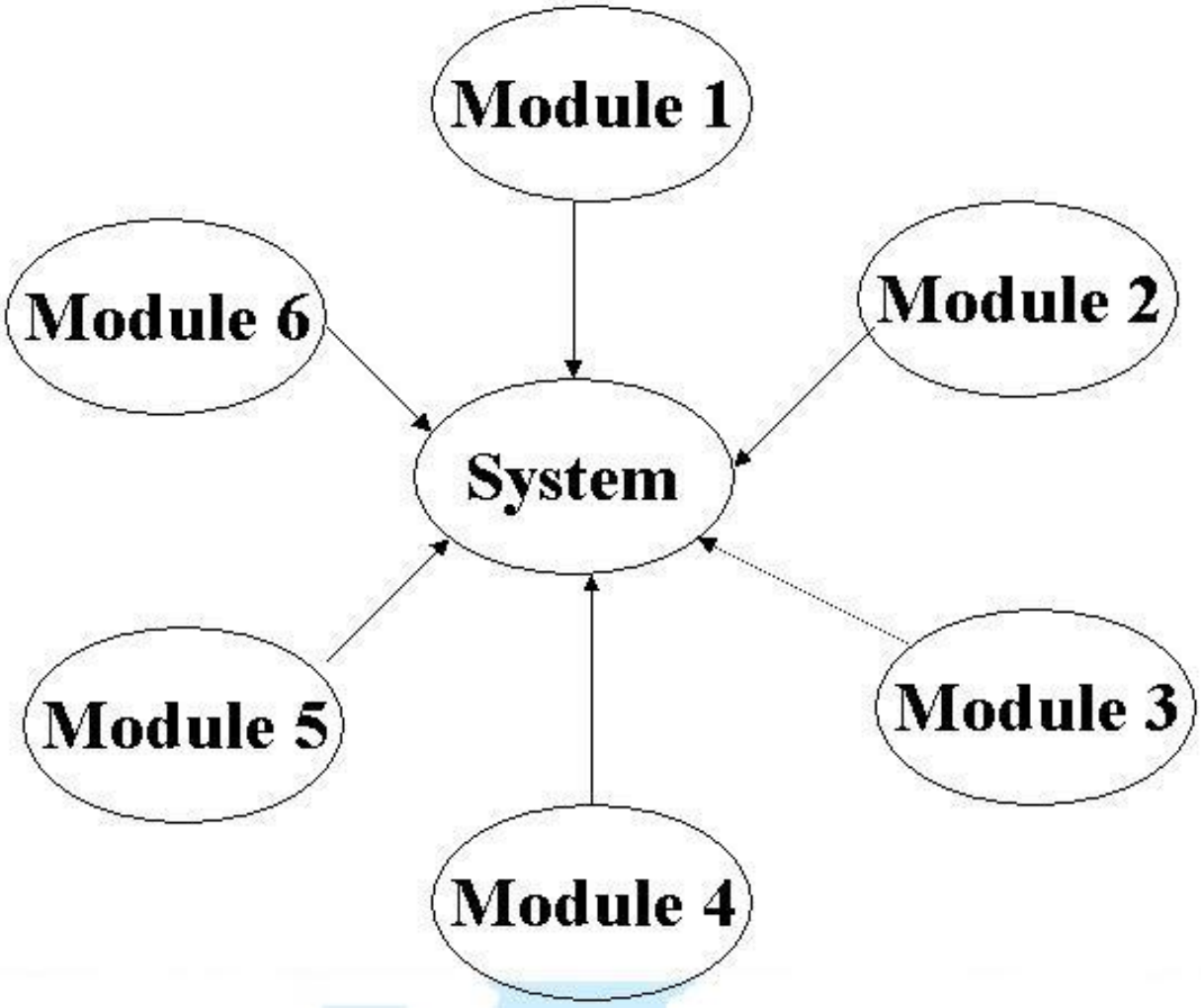
Example Structured Design



Big bang Integration Testing

- Big bang approach is the simplest integration testing approach:
  - all the modules are simply put together and tested.
  - this technique is used only for very small systems.

Big Bang Integration Testing

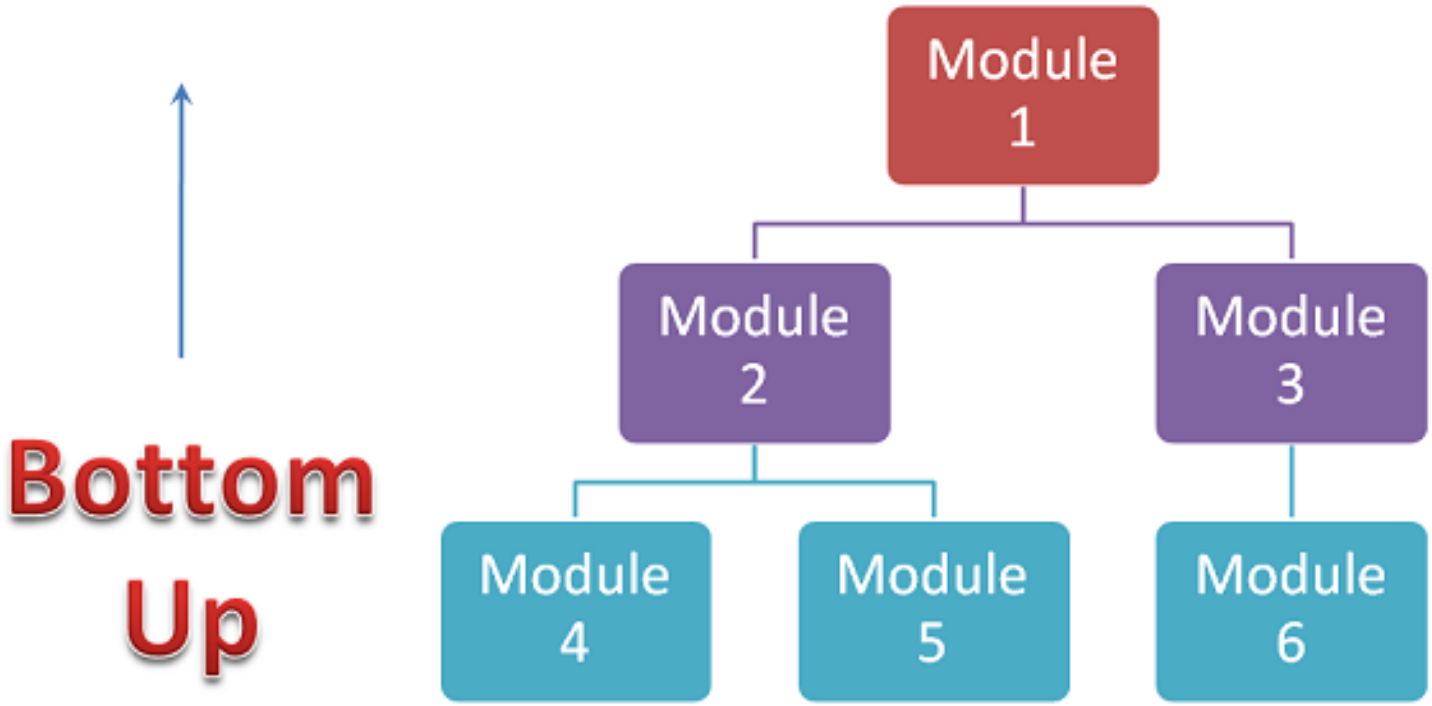


Big bang Integration Testing

- Main problems with this approach:
  - if an error is found:
    - it is very difficult to localize the error
    - the error may potentially belong to any of the modules being integrated.
  - debugging errors found during big bang integration testing are very expensive to fix.

Bottom-up Integration Testing

- Integrate and test the bottom level modules first.
- A disadvantage of bottom-up testing:
  - when the system is made up of a large number of small subsystems.
    - This extreme case corresponds to the big bang approach.


Bottom
Up
Module
1
Module
2
Module
3
Module
4
Module
5
Module
6

Top-down integration testing

- Top-down integration testing starts with the main routine:
  - and one or two subordinate routines in the system.
- After the top-level 'skeleton' has been tested:
  - immediate subordinate modules of the 'skeleton' are combined with it and tested.

Mixed integration testing

- Mixed (or sandwiched) integration testing:
  - uses both top-down and bottom-up testing approaches.
  - Most common approach

Integration Testing

- In top-down approach:
  - testing waits till all top-level modules are coded and unit tested.
- In bottom-up approach:
  - testing can start only after bottom level modules are ready.

Phased versus Incremental Integration Testing

- Integration can be incremental or phased.
- In incremental integration testing,
  - only one new module is added to the partial system each time.

Phased versus Incremental Integration Testing

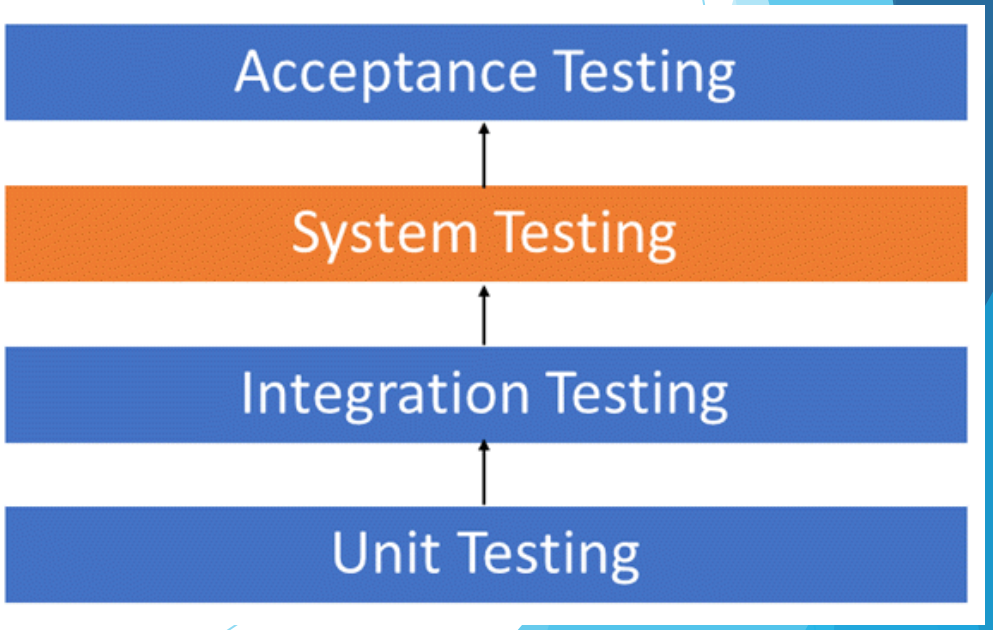
- In phased integration,
  - a group of related modules are added to the partially integrated system each time.
- Big-bang testing:
  - a degenerate case of the phased integration testing.

System Testing

- System tests are designed to validate a fully developed system:
  - to assure that it meets its requirements (SRS).

System Testing

- The purpose of a system test is to evaluate the end-to-end system specifications.



System Testing

- During system testing, in addition to functional tests:
  - performance tests are performed.

Performance Testing

- Addresses non-functional requirements.
  - May sometimes involve testing hardware and software together.
  - There are several categories of performance testing.

Types of SYSTEM (Performance) TESTING

1. Usability Testing- mainly focuses on the user's ease to use the application, flexibility in handling controls and ability of the system to meet its objectives

2**. Regression Testing-** involves testing done to make sure none of the changes made over the course of the development process have caused new bugs.

It also makes sure no old bugs appear from the addition of new software modules over time.

3. Storage Testing: Testing type that verifies the program under test stores data files in the correct directories and that it reserves sufficient space to prevent unexpected termination resulting from lack of space. It is usually performed by the testing team

4. Recovery testing - is done to demonstrate a software solution is reliable, trustworthy and can successfully recoup from possible crashes.

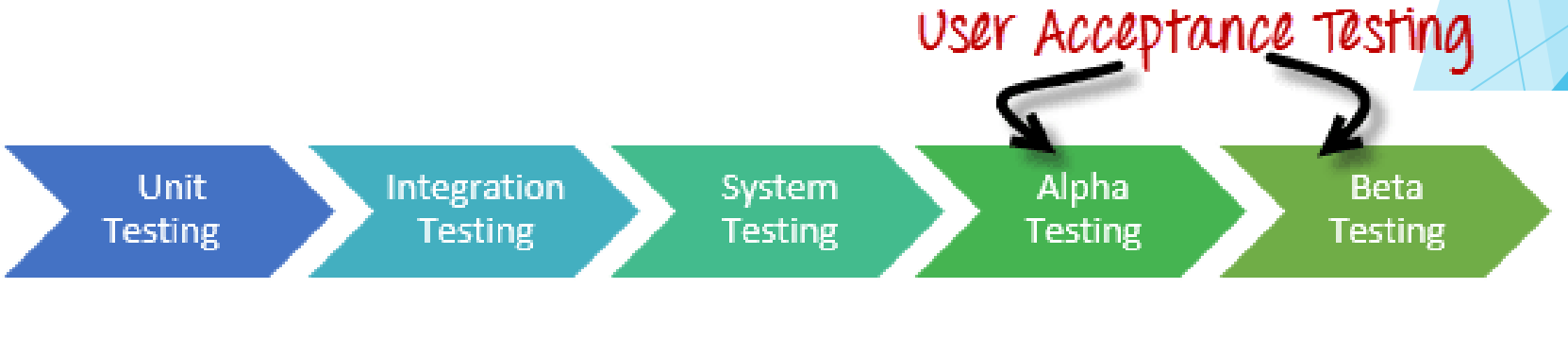
5. Migration testing- is done to ensure that the software can be moved from older system infrastructures to current system infrastructures without any issues.

- **Backward Compatibility Testing:** Testing method which verifies the behavior of the developed software with older versions of the test environment. It is performed by testing team.

6. Load Testing: Testing technique that puts demand on a system or device and measures its response. It is usually conducted by the performance engineers.

Acceptance Testing

- 1. ALPHA TESTING:
- Testing is carried out by the test team within the developing organization.



Beta Testing

- Beta testing is the system testing:
  - performed by a selected group of friendly customers.

	Alpha Testing	Beta Testing
Testing kind	acceptance testing	acceptance testing
Main goals	• to evaluate the software quality • to make sure everything works properly • to check readiness for Beta testing	• to evaluate the users' satisfaction with the product • to receive a valuable feedback • to ensure release readiness
Performed by	company employees	customers
Performed in	a lab or testing environment	in a real-time environment
Applied techniques	black-box and white-box testing	black-box testing
When to conduct	after system testing	after Alpha testing
Test run	many	1 - 2
Reliability and security	not checked	checked
Functionality and usability	checked	checked
Developers	engaged	not engaged

Some more types of performance testing:

Volume Testing

- Addresses handling large amounts of data in the system:
  - whether data structures (e.g. queues, stacks, arrays, etc.) are large enough to handle all possible situations
  - Fields, records, and files are stressed to check if their size can accommodate all possible data volumes.

Configuration Testing

- Analyze system behavior:
  - in various hardware and software configurations specified in the requirements
  - sometimes systems are built in various configurations for different users
  - for instance, a minimal system may serve a single user,
    - other configurations for additional users.

Compatibility Testing

- These tests are needed when the system interfaces with other systems:
  - check whether the interface functions as required.

Compatibility testing Example

- If a system is to communicate with a large database system to retrieve information:
  - a compatibility test examines speed and accuracy of retrieval.

Recovery Testing

- These tests check response to:
  - presence of faults or to the loss of data, power, devices, or services
  - subject system to loss of resources
    - check if the system recovers properly.

Maintenance Testing

- Diagnostic tools and procedures:
  - help find source of problems.
  - It may be required to supply
    - memory maps
    - diagnostic programs
    - traces of transactions,
    - circuit diagrams, etc.

Maintenance Testing

- Verify that:
  - all required artifacts for maintenance exist
  - they function properly

Documentation tests

- Check that required documents exist and are consistent:
  - user guides,
  - maintenance guides,
  - technical documents

Documentation tests

- Sometimes requirements specify:
  - format and audience of specific documents
  - documents are evaluated for compliance

Usability tests

- All aspects of user interfaces are tested:
  - Display screens
  - messages
  - report formats
  - navigation and selection problems

Environmental test

- These tests check the system's ability to perform at the installation site.
- Requirements might include tolerance for
  - heat
  - humidity
  - chemical presence
  - portability
  - electrical or magnetic fields
  - disruption of power, etc.

Test Summary Report

- Generated towards the end of testing phase.
- Covers each subsystem:
  - a summary of tests which have been applied to the subsystem.

Test Summary Report

- Specifies:
  - how many tests have been applied to a subsystem,
  - how many tests have been successful,
  - how many have been unsuccessful, and the degree to which they have been unsuccessful,
    - e.g. whether a test was an outright failure
    - or whether some expected results of the test were actually observed.

Regression Testing

- Does not belong to either unit test, integration test, or system test.
  - In stead, it is a separate dimension to these three forms of testing.

Regression testing

- Regression testing is the running of test suite:
  - after each change to the system or after each bug fix
  - ensures that no new bug has been introduced due to the change or the bug fix.

Regression testing

- Regression tests assure:
  - the new system's performance is at least as good as the old system
  - always used during phased system development.