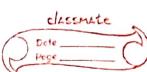


### Chapter - 3

#### Verification & Validation



Definition	Verification	Validation
The process of evaluating a system or component (to determine whether product of the given development phase satisfy the condition imposed at the start of that phase).	The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified business requirements.	The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified business requirements.

- IEEE

- IFSI

Objective: To ensure that the product is being built according to the requirements and design specifications.

To ensure that the product fulfills its intended use when placed in its intended environment.

Question: Are we building the right product?

Are we building the right product?

Evaluation Plan, Requirement specification item Design specification, code, Test cases.

Testing type Static Testing Dynamic testing

Responsibility It is done by QA team. It is done by UX/UI testers/ business users with the help of QA team.

Testing order Generally carried out before validation. - Generally follows verification.

Verification	Validation
Methods	Use methods like walkthrough, review, inspection etc.
Low Level activity	High level activity

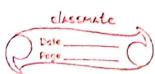
#### Advantages of Software Verification:

- Demonstration of consistency, completeness and correctness of the software at each stage and between each stage of the development life cycle.
- Verification helps in lowering down the count of the defect in the later stages of development.
- Verifying the product at the starting phase of the development will help in understanding the product in a better way.
- It reduces the chances of failures in the software application or product.

#### Advantages of Validation:

- During verification if some defects are missed then during validation process it can be caught as failures.
- If during verification some specification is misunderstood and development had happened then during validation process while executing that functionality the difference between the actual result and expected result can be understood.
- Validation helps in building the right product as per the customer's requirement and helps in satisfying their needs.

There are two types of testing techniques:



### I. Black Box Testing      II. White Box Testing

#### Black Box Testing:

Here, the software is treated as a "black box", and its internal logic for processing the data is not considered.

A set of input is fed to the software, and the outputs delivered by the software are compared with the expected outputs.

BBT is normally conducted from the user interface or command line.

The efficiency of BBT depends on the care with which the test cases and test data are designed.

If test cases were exhaustive, the testing is exhaustive and has a better chance of detecting failure.

#### Steps for conducting Black Box Testing:

(1) Create the executable file of the software unit that is to be tested.

Install it on the test system (environment).

(2) Prepare the master data that is required to run the test.

(3) Study the test plan and note the test objectives.

(4) Study the test case designed for the test.

(5) Run the program from either the command line or the user interface.

(6) Execute test cases in the sequence specified, log the result and determine whether test execution ~~whether~~ failed or passed the test.

(7) If the result is doubtful, restore the test data to the protest image and reexecute the test.

(8) Arrange the test result for a managerial review and submit the report to the originator of the test request.

(9) Provide clarification to the originator of the test request and the individuals involved in defect resolution.

(10) Carryout regression testing to ensure that the defects are resolved satisfactorily.

(11) If regression test uncovers defects, iterate steps of 7 to 10 until all defects are satisfactorily resolved.

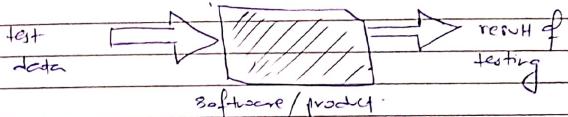


Fig. Black Box Testing

#### White Box Testing

- WBT considers the internal logic and program statements of the software.

- It involves stepping through every line of code and every branch in the code.

- To use this technique, the tester should be knowledgeable in the software programming language and should understand the structure of the program.

- WBT ensures that all program statements and all control structures are tested at least once.

- WBT can be conducted from the command line, user interface or from within the program.

- The other way to conduct WBT is by using the interactive development environment (IDE) or a language specific debugger.

- These tools have facilitate to perform the following:
  - \* Step through every line of code.
  - \* Set break points in the code.
  - \* Set the initial value or change the value of variables or constant.
  - \* Traverse through every path for control structures.
  - \* Stop execution at any point in the program and resume testing from the beginning or anywhere in the program.
  - \* Move execution from any point to any other point in the program.

#### # Steps for conducting WBT from the IDE or the debugger.

1. Prepare the master data that is required to run the test.
2. Study the test plan and note the objectives.
3. Study testing guidelines if any.
4. Obtain applicable checklist and keep them handy.
5. Study the test cases that is designed to test.
6. Receive the software unit to be tested from the development team and load it onto test system.
7. Launch the IDE or debugger and open the software unit to be tested in the IDE or debugger.
8. Set breakpoints where a pause in program execution is desired.
9. Execute the test cases in the sequence specified, log the result and determine whether test execution failed or passed the test.
10. Arrange the test result for managerial review and submit the report to the originator of the test.

- (i) Provide classification + the originator of the test request and individual involved in defect resolution.
- (ii) Carryout regression testing to ensure that the defects are resolved satisfactorily.
- (iii) If regression test uncovers defect, iterate step 10 to 12 until all defects are resolved satisfactorily.
- Sometimes an IDE or debugger might not be available to carryout WBT. In such cases, test software that will test the software unit under consideration need to be developed.
- Such test also performs the following functions.
  - i) It calls the software unit to be tested and passes the required parameters to it.
  - ii) It receives the final values produced by the software unit to be tested and presents them to the tester for evaluation of their efficiency.
  - iii) It allows iteration if required with the software unit to be tested during execution.

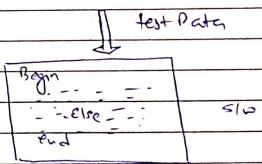


fig. White Box Testing.

- classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_
- 1 Following confirmations are ensured by verification:
  - 2 achievement of the core functionality set for the artifact.
  - 3 comprehensiveness and completeness of the artifact.
  - 4 conformance to standards and guidelines defined for the type of artifact in the project plan.
  - 5 efficiency and effectiveness of the solution presented by the artifact.
  - 6 clarity and correctness of the artifact.
  - 7 achievement of all ancillary functionality that is expected of the artifact.
  - 8 no unnecessary functionality is present in the artifact.
  - 9 no malicious functionality is contained in the artifact.
  - 10 the format of the artifact adheres to organizational standards or customer specification.
  - 11 fulfillment of the requirements of downstream functions of the artifact when it is to be used downstream.
  - 12 cases where testing would not be able to detect any defects.
  - i unnecessary declaration of variables and constants. may not be harmful but it ties up RAM and slows down the system.
  - ii inclusion of unwanted libraries may be harmless but it increases the size of setup file and disk space.
  - iii multiple algorithms can be used to achieve a given functionality. Efficiency of algorithm used cannot be detected by normal testing.
  - iv programmers tend to use the control structure. They are most comfortable with rather than them.

- right one.  
classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_
- v normal testing can't determine if the right control structure was used.
  - vi it cannot detect 'time bomb', 'event bombs' and 'random bombs' in a program.
  - vii it cannot detect ~~local code~~ (comment in the program)
  - viii the tools and techniques for verification adopted in software development organization include:
  - 1 walkthroughs
  - 2 inspections
  - 3 audits
  - ix walkthroughs (peer reviews)
  - the review of work product performed by peers during development of the work products to identify defects for removal.
  - the people (peer) conducting a walkthrough normally have experience and expertise similar to that of the author. They deliver a report at the end of the walkthrough known as the review report which contains
    - \* defects uncovered during the walkthrough
    - \* opportunities for improvement
    - \* suggestions for improvement
  - x there are five types of walkthrough:
  - 1 independent walkthroughs
  - 2 guided walkthroughs
  - 3 grouped walkthroughs
  - 4 expert reviews
  - 5 managerial reviews

- 1) Independent walkthroughs:
- Project leader chooses a pair (reviewer) for carrying out walkthrough and provides reviewer the artifact to be reviewed in (readonly mode.)
  - The reviewer reviews the artifact, prepares the review report and gives to the Project leader.
  - No guideline or interaction to the author is allowed.

Advantage: Author can allocate time to another activity while artifact is being reviewed.

Disadvantage: Reviewer may not understand the artifact if complex algorithm was used.

- 2) Guided walkthroughs:
- A guided walkthrough is conducted in the presence of the author of the artifact being reviewed.
  - The author walks the reviewer through the artifact, explaining the contents of it.
  - Opportunity for improvement is discussed among them and the author is will note it along with the best manner of improvement.

Advantage: Duration of review is much shorter than independent.

Disadvantage: -> The author might convince the reviewer that a defect is in fact not a defect.  
-> Author and reviewer may disagree about an opportunity for improvement.

- 3) Group walkthroughs:
- It is used when the knowledge of more than one person is necessary to review the artifact.
  - It can be conducted in three modes:
    - i) Postal review
    - ii) Meeting review
    - iii) Guided Meeting review
- 4) Postal review:
- Group of reviewer is made and a review coordinator is selected.
  - Artifact is provided to all the reviewer. They provide review report to the coordinator from their convenient location.
  - The coordinator prepares final report by looking at all those reports.

Advantage: Reviewers don't interact with each other, so their report will be independent.

Disadvantage: It takes longer time because the process cannot be completed until the slowest member finishes.

- 5) Meeting review:
- Group of reviewer is made and a review coordinator is selected.
  - Artifact is provided to all the reviewers.
  - The coordinator arranges a review meeting.
  - All reviewers come to the meeting with their review comment.
  - The review report is finalized in the meeting by discussing all the comments from all the reviewers.
- Advantage: faster than Postal Review.
- Disadvantage: Some reviewers may not fully contribute in their review.

### ii) Guided Meeting reviews.

- Similar to Meeting reviews, except the team members do not have to come to the meeting prepared with comments
- In the meeting, the author provides artifact to the reviewer, review team members discuss and offer their comments.
- The review coordinator collects the comments and prepare the review report.

Advantage: Turn around time is the shortest.

Disadvantage: Some reviewers may not fully contribute in the review process.

### iii) Expert Reviews

Expert review is necessary when:

- There is no one on the project team or organization who has sufficient knowledge and experience in particular domain or technology.
- complex mathematical algorithms are included in the artifact.

#### Types of experts:

- ✓ - Domain experts
- ✓ - Subject matter experts
- ✓ - Technology experts
- ✓ - Social experts

Advantages: Helps to develop robust system

Disadvantages: costly.

- should be scheduled properly.

### 5.3 Managerial Reviews

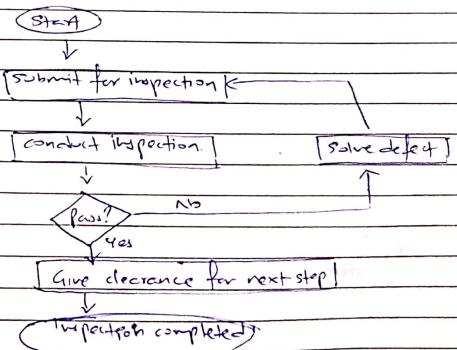
HiC performed by the person who directly supervises the author of the artifact.

- Managerial reviews start once the peer review has been conducted and all defects are fixed and closed.
- If defects are found during the managerial review, the work is reassigned either to the same person (or team) who conducted the peer review or another person to re-do the peer review.

### Inspection

It ensures that all required components are ready for next stage.

The output of the inspection activity is an inspection report which specifies the system is passed or failed.



### Types:

- 1) System Testing Readiness Inspection
- 2) Acceptance Testing Readiness Inspection
- 3) Delivery Readiness Inspection

#### 1) System Testing Readiness Inspection

- This inspection ensures that all the required components are ready for conducting system testing.
- This inspection is carried out by QA department personnel.
- Report is created and handed over to the PL.

It ensures that the server is equipped with following:

- i) Hardware (web server, database server, appserver)
- ii) System Software (operating system & other software)
- iii) Security software (antivirus, antispyware, firewalls)
- iv) High bandwidth network
- v) Master data etc

It also ensures the following conditions are met:

- i) All test cases and test plans are ready.
- ii) All testers are notified about the testing.
- iii) All bugs are fixed.
- iv) The closing criteria for testing.

#### 2) Acceptance Testing Readiness Inspection:

Acceptance testing is conducted as a pre-requisite for obtaining the customer's acceptance of product

delivery and thus for receiving payment.

The following aspects are ensured during an acceptance testing readiness inspection.

- i) All the planned QA activities inspections, reviews and tests are conducted.
- ii) The software is complete in all aspects.
- iii) All the required hardware are configured.
- iv) All the system software, middleware and databases are properly located in hardware.
- v) fallback plans are ready for fixing any bugs.

#### 3) Delivery Readiness Inspection:

The purpose of delivery readiness inspection is to ensure that the delivery package has all the components and that the right versions are included in it.

- i) It ensures the following things:
  - ii) The right versions of source codes and libraries are used in building project.
  - iii) All components mentioned in the software delivery are included in the delivery set.
  - iv) All planned QA activities are performed on all components of the delivery set.
  - v) The delivery set is done to the right person.

There are 3 modes of project delivery:

- i) Single-shot delivery:
  - ii) - Here all components are delivered at once

#### ii) Interim delivery:

- It means one of multiple deliveries is delivered.

#### iii) Final delivery:

- It means the last delivery of a series of deliveries.

Imp

efficacy → ability to practice desired effect

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

### Audits

- Audit are document verification system in which project documents and records are compared with the organization's standards or defined process.
- They are short in duration (one to two hours).
- They ensure that a project is being executed in conformance with the organization's defined processes and that is ready for next phase of execution.
- Audit consists of auditors (the people who conduct the audit) and auditee (the people whose project is being audited).

Audits can be classified in a variety of ways:

- ✓ Conformance Audits versus Investigative Audits
- Conformance Audit focus on the efficacy of implementation of organizational process during project execution.
- ✓ They are conducted to compare and contrast the project document with the organizational process.
- Investigative Audits focus on finding the causes of failures but sometimes focus on finding the causes for an extraordinary success.

Vertical Audits Versus Horizontal Audits:

- Vertical Audits are performance audits conducted across the organization on either a few selected projects or on all aspects of these projects; they focus on all aspects of the project(s).

Horizontal Audits are also Conformance audits conducted across the organization on either a few selected projects or on all projects, but they focus on only one aspect of the project(s).

#### 3) Periodic Audits versus Phase-End Audits:

Periodic Audits are conformance audits that are conducted at the organizational level based on calendar duration. ISO-certified organizations do this every 2-3 months.

Phase-End Audits are conformance audits that are conducted triggered by project events.

#### 4) Internal Audits versus External Audits:

Internal Audits can be any type of audit, but they are conducted by people internal to the organization.

however the internal auditors are independent of the project being audited.

External Audits are conducted by an external agency that specializes in the audit process.

#### Implementation of verification activities in Projects.

It is better to use the planned approach rather than ad-hoc approach to implement verification activities in projects.

In planned approach, verification activities for a project are planned by the software project manager (SPM) during the project planning stage and they are recorded in the software quality assurance plan.

- The software QA plan records the following:
  - i) A list of all verification activities planned after implementation in the project.
  - ii) All project execution phases, after which phase-end audits are to be conducted.
  - iii) All stages of the development at which software inspections are to be carried out.
  - iv) All artifacts that are to be subjected to peer reviews.
  - v) All possible individuals who will conduct the reviews.
  - vi) All type of reviews that are to be conducted.
- The verification activities planned for a project need to conform to the organizational verification process.

#### Validation:

- It is the process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specific requirements  
- IEEE
- It answers the question "Are you building the right product?"
- To achieve full worth of validation, three factors must be applicable:
  - i) Validation is performed by an independent person who are not the same persons as those making the claim.
  - ii) Validation is performed not just against the specification of the claimant but also against external specifications.

3) Validation is performed for satisfying the claimant and providing confidence in stakeholders but not perform for self assurance.

- # Validation of Software design:
  - There is difference in approach of validating different nature of projects (Software, automobile, Electronics etc)
  - In case where the designs require production of large quantity of products, a prototype is built and subjected to all necessary tests.
  - There are two approaches for validating a design:
    - i) Scale down approach
    - ii) Prototype approach (Build and improve prototype - use and discard prototype)

- ii) Scale down approach:
  - This approach is basically used in automobile ship building and aircraft manufacturing industries where they cannot afford to build a wrong product.
  - In case of ship building industry, before building a large ship, a much smaller model of the ship is built and tested in the scale-down environment.
  - The result of the test will validate the design of the proposed ship.

- # Prototype approach:
  - In case where the design is used to produce a large quantity of products, a prototype is built and subjected to all necessary tests.

- In software development, there are two types of prototypes:

i) Build And Improve prototype:  
Here skeletons of the software product such as screen layout, report layout and simple navigation are built on the actual development platform. Software developers continue to improve the design based on the results of validation of the prototype.

ii) Use And Discard prototype:  
Here a prototype is built on a mockup using drafting tools. The requirements are validated and then the actual software product is built on the real development platform.

# Validating the product specification:

- Improper definition of specification leads to improper software design and ultimately, the right product cannot be built. So, product specification should be validated.

- Generally there are two ways of validating the product specification.

i) Group reviews

ii) Brain Storming

iii) Group Reviews

The group reviews for validation of software product specification is mostly achieved through the use of domain experts. However, the postal reviews and meeting reviews can also be used.

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

iv) Brain Storming

- Here, concerned experts gather in an informal meeting and deliver it to validate the artifact at hand.  
- It is used when the activity is performed for the first time.

# Validating the software Product.

-> Software is defined as "the process of executing a software item to detect the differences, if any, its behaviour and the desired behavior".

-> Testing is carried out to assure the customer that the product conforms to the specification and functionality initially agreed upon.

-> Testing is used to confirm quality rather than achieve it.

-> In software domain, however, software can be subjected to any type of testing without the fear of damaging the product or database or injuring the testers.

-> This flexibility allows extensive use of software testing as the main tool for quality assurance in the software development domain.

-> Independent testing has been gaining wider acceptance in software testing.

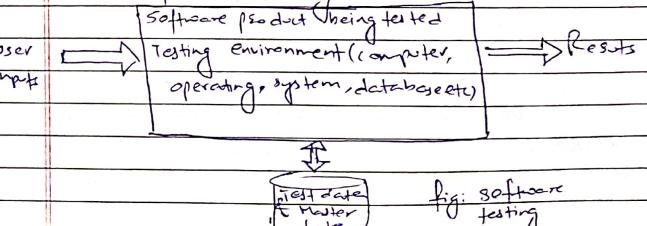


fig: software testing

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

## Testing different types of Software products

### ① Batch Processing Systems:

- Here, data is entered offline so when data is submitted to the software, it is expected to have been validated, accurate and in right form.
- Here, blackbox testing technique has been used so test data that forced the software to traverse each path in the code must be created for uncovering all possible errors.
- Examples: monthly processing system, early processing system in bank etc.

- If an error is encountered, a batch processing system fails.

### ② Online Systems:

- Online Systems are also referred as event-triggered systems
- To test online system
  - Test cases are prepared that send positive inputs which prove that the software does what it is supposed to do.
  - Test cases are prepared that send negative inputs which shows that the S/W has all the data-validation and error handling routines to prevent the wrong data from entering or failing the system.

### ③ Real-Time Systems:

- Real-time systems interact with machines and control them to perform desired functions but interface with hardware so machine responses have to be simulated during testing so that a

wrong signal from the software does not damage the machine or its surroundings

### ④ Scientific Applications:

- Scientific Application systems are built around complex mathematical equations that process large algorithms with few inputs.
- They are processing intensive rather than data intensive.
- So, to test this system, data with a solution that is known and has been worked out on a reliable existing system can be used.

### ⑤ Mobile Applications:

- To test mobile software, either an interface to the actual mobile hardware is needed or test software that simulates the hardware responses has to be developed.

### ⑥ Software Simulators:

- Simulators interface with the input hardware of the actual hardware system.
- To test this software, either the actual hardware must be brought in or a test bed that generates signals identical to those generated by the actual hardware must be prepared.

#### # Testing Basis:

- i) Black Box testing
- ii) White Box testing

#### # Approaches to testing:

- There are two basic approaches to testing

1. Intuitive Testing
2. Process driven Testing

#### 1. Intuitive Testing

- It is carried out by an experienced tester who uses his or her common sense.
- Testing is carried out using the experience and intuition of the tester.
- Here certain creativity and common sense is expected from tester.

User: [ ]

Phase: [ ]

Enrolled from: [ ]

Enrolled To : [ ]

[ Go ]

Note: Explain this figure by considering Enrolled from and Enrolled To

i.e. Enrolled from must be smaller than Enrolled To which is known by experience and common sense.

Advantages: i) It is quicker to test and much of the time usually spent on test planning and test case

design is saved.

#### Disadvantage 8:

- i) It requires an experienced tester to conduct the tests and uncover all defects.
- ii) It is almost impossible to test all functionalities without a test plan.
- iii) It is almost impossible to conduct thorough testing even for critical functions.

#### d. Process-Driven testing:

In this approach, software testing is conducted according to a defined validation process using a test plan and a set of test cases.

- for each project, a software test strategy is decided upon during the project planning stage, and it is recorded in the test plan or software quality assurance plan.

The test strategy is described below:

#### e. Test Strategy:

Test strategy is concerned with uncovering as many defects as possible within the allocated time and cost budgets and with maximizing the impact of such testing.

- The first step in finalizing the test strategy is to set testing objectives such as the following:

#### 1. Quality objectives:

- uncover all defects, irrespective of time and cost.

#### 2. Customer acceptance objectives

- The main objective of testing is to convince the

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

customer that the product is built in accordance with the customer's requirements to obtain customer's requirements sign-off and be paid by the customer for the product.

## 2. Product certification objectives:

- The tests are carried out as specified by the customer, and the product is certified as requested by the customer.
- Such as virus and spyware-free, product rating, usability, functionality etc.

In addition to objectives, the following also are part of test strategy.

- types of tests to be included in testing
- How to test
- Regression testing
- criteria for successful completion of testing
- Mechanism for defect closure and escalation, when necessary.
- Progress reporting
- Defect analysis

## Test case Design:

Some of the techniques for test case design are:

- ① Equivalence Partitioning (explained later)
- ② Boundary Value Analysis (explained later)
- ③ Error Guessing

In this technique, the test case designer uses experience, intuition and common sense

- classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_
- to design test cases that are likely to detect errors.
  - Using this technique requires many years of experience in developing and testing software.
  - There is some kind of guessing involved. So, software engineers are likely to commit errors.

Example: date → February 30, month → 13, year → 5555 etc should be rejected by the system.

## ④ Logic coverage:

- Here, the logic of the SW design is used to derive test cases. The test cases evaluate the SW to ensure that the logic is producing the desired result.
- The test case may either prove or disprove the logic built into the SW design.

## ⑤ Consistency checking:

- It involves designing test cases that check for consistency of processing from different points in the software.
- It is applicable whenever there is a requirement to display the same information through different functionalities of the SW.

Example:

The number of items available should be displayed consistently in all POS computers even though the number is increased or decreased from other sources.

## ⑥ Requirement Tracing:

Here, the requirements from the user requirements specification are traced and the test cases to

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

test the software product are designed to confirm that the SW does in fact meet and fulfill the user requirements.

- ① Response Time checking
  - Response Times are extremely important in real-time software systems
  - Test cases are designed to test the response time of a software product to ensure it is within acceptable limits.

#### #1 Test Environment

- First software development environment is set up during project initiation stage which may include:
  - ✓ Server for the database
  - ✓ client machine, one for each developer
  - ✓ Networking for all the machines
  - ✓ Local and SW information artifacts (Requirements document, design document, standards, and guidelines etc) into a common area accessible to all team members.
  - Set up configuration management procedures to deal with change control and code promotion (code to unit testing to integration testing to system testing to acceptance testing.)
  - Set up communication mechanism.
- Then test environment is created which should be a replica of the development environment or its scaled-down version.

\* It needs server for database.

\* few client machines to conduct the testing which have the same software development toolkit as the development client machines have

\* Server should have test data and backup facility whenever a component of SW needs to be tested,

\* a copy is moved into test environment.

\* component is tested.

\* after it passes all test cases, it is removed from the test environment and test data is reset to the pretest image.

#### 1.1

##### #1 Testing Scenarios:

There are two testing scenarios:

- ① Project Testing or Embedded Testing
- ② Product Testing

##### ① Project Testing

These are the testing activities carried out as a part of SW development project to ensure that development work is defect free.

- It is also called embedded testing, because testing progress along with development progresses and testing work completed soon after development work is complete

following tests are carried out in Project Testing:

- i) Unit Testing
- ii) Integration Testing
- iii) System Testing
- iv) User Acceptance Testing

- Unit is the smallest testable part of an application like functions, classes, procedures, interfaces etc.
- makes sure that code meets its design and requirements.
- Pros: - issues are found at early stage and reduces the cost of bug fixes.  
- helps in maintaining the code.

#### Unit Testing:

- carried out by the person who wrote the code and additionally by an independent peer using white box testing.

#### Steps:

- ① Programmer constructs a component according to the design document and informs the project manager.
  - ② Project Leader assigns an independent peer for unit testing along with the programmer.
  - ③ A copy of code is made available to the tester along with test cases, guidelines and checklists.
  - ④ Tester loads the code in the testing environment.
  - ⑤ Tester executes all the test cases and logs the result.
  - ⑥ Tester hands over the test log to the project leader.
- The project leader tells the developer to solve the defects and request the tester to conduct regression testing.
- ⑦ If regression testing reveals new defects then repeat steps 5 to 7 until all defects are closed satisfactorily.

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

- Integration Testing carried out either as a
  - one-off (when all integration is completed)
  - incrementally (whenever one unit of SW is integrated, its integration is tested until all units are integrated and tested)
- Black-Box is used for one-off integration testing.
- White-Box is used for incremental integrated testing.
- There are two approaches to product integration:
  - Top-down approach
  - Bottom-up approach

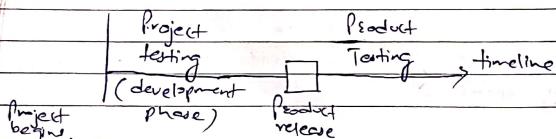
In top-down approach, the top level component from where the product functionality would begin is developed first. Each component is developed and integrated with the top-level components of their respective module.

In bottom-up approach each component of lowest level is constructed and tested first. When all the components of sub-module and are completely constructed and tested, the top level component of the submodule is constructed and all its components are integrated with it.

#### System Testing:

- It should investigate both functional and non-functional requirements of the system.
- Carried out to ensure that the software works on all intended target systems.
- System testing is carried out in following applications

- In mainframe application, development usually takes place in a simulated environment. Once development is completed, the SW is loaded onto the mainframe computer and tested in actual environment.
  - In client-server applications, system testing is carried out on all intended operating system version: win 7, win 2000, win XP, etc and also on database servers as: SQL server 7, SQL server 2000, etc.
  - In web applications, system testing includes testing on various client systems such as Windows based machines, Unix or Linux based machines and Macs. Also, in various browsers such as chrome, firefox, internet explorer and in different versions too.
- User Acceptance Testing:
- Carried out to obtain customers sign-off so that the SW can be delivered and payment received
  - conducted using Black Box Testing technique
  - Based on the test cases designed to prove that the SW performs all the functions specified in the user requirement document provided by the customer/approved by the customer.
  - It is positive testing.
  - Done by the user for customer.
  - if passed, it shows that the system is ready for deployment



② Product Testing:  
When a product is sold without customer's specific customization, it is referred to as a commercial off-the-shelf (COTS) product.

Product Testing is carried out on a COTS SW product to ensure that the product works w/o any defects in a variety of customer scenarios.

following tests are carried out in Product Testing:  
Load Testing:

Large number of users are logged in & use the SW in a random manner, objective of load testing is to find out how the SW manages multiple requests and whether it serves up accurate results or mixes the results up.

ii) Volume Testing:

Volume testing subjects the SW to a high volume of data to see if performance degrades as the amount of data increases.

iii) functional Testing:

It tests the software to ensure that all its functional features/functions are working correctly.

Two types of functions:

Main function: It fulfills product and customer needs.

Auxiliary function: Ensures security, protection against misuse, data integrity etc.

It is positive testing.

#### iv) End-to-End Testing:

Here, an entity in the application is tracked from birth to death.

example:

Tracking the employee from when he joins the system to when he leaves the system.

#### v) Parallel Testing:

Here a number of users access the same function and are either inputting or requesting the same data.

It helps to ensure data consistency.

#### vi) Concurrent Testing:

Concurrent testing ensures that the software product has sufficient concurrency controls built in. It is carried out to identify any issues that occur when two or more users access the same functionality and update or modify the same data with different values at the same time.

#### 7) Stress Testing:

Stress testing stresses the S/W by:

- making expected resources unavailable.
- causing deadlock scenario
- not releasing resources
- disconnecting the network etc.

To ensure that the software can handle such stress.

#### 8) Positive Testing:

- To ensure that all defined functions are performing as expected.

- not designed to uncover defects.

#### 9) Negative Testing:

- It uncovers deficiencies in the S/W focused on error handling and facilities improvement.

#### 10) User Manual Testing:

- It is done to ensure that the user manual of S/W are in synchronization with each other.

#### 11) Deployment Testing:

- It simulates the target environment and deploys the S/W to ensure that deployment specified is appropriate.  
- Here, only the failures in deployment are considered.

#### 12) Sanitizing Security Testing:

- ensures that the components of S/W package are complete and the versions appropriate.

#### 13) Retesting Testing:

- carried out after defects uncovered by earlier testing has been fixed.  
- previous test is not repeated but only reported defect is retested.

#### 14) Regression Testing:

- completing an entire test again  
- to ensure the process of fixing defects has not introduced any fresh errors in S/W.

15) Security Testing:  
- check against the threat of viruses, spyware.

16) Performance Testing:  
- evaluates overall performance of the system indicates response time, turn around time, report generation etc

17) Usability Testing:  
- involves testing SW for different types of usage to ensure that it satisfactorily fulfills the requirements of specified functional areas.

18) Install- Uninstall Testing:  
- to ensure that install/uninstall functions are satisfactorily performed.

19) Comparison Testing:  
- involves testing of the product with competing products to identify differences & to determine the relative position of the product against competing products.

20) Regulation Conformance Testing:  
- ensures that the product conforms to all government regulations meeting to minimum standards.

21) ~~Alpha~~ Testing - Testing within organization where SW was developed by the actual users.

22) Beta Testing:  
Testing done by actual users at their respective locations outside the organization that developed the SW product.

1T Test case Design Template.

Test Case Template

Project Name:

Project ID:

Test Case ID:

Test priority:

Module Name:

Test Title:

Description:

Precondition:

Dependencies:

S-N	Test steps	Test Data	Expected Result	Actual Result	Status Note

Post condition:

Test case Defn:

A test case is a document which has a set of test data, preconditions, expected results, and post conditions, developed for a particular test scenario in order to verify compliance against a specific requirement.

## Chapter 3:

Software Verification:

do the right thing = validation

do the things right = verification

"The process of evaluating a system or component to determine whether the products of the given development phase satisfy the condition imposed at the start of that phase."

— IEEE

"Confirmation that works products properly reflect the requirements specified for them. In other words, verification ensures that you built it right."

— CMMI

following specification confirmations are sought from verifications:

1. achievement of the core functionality set for the artifact.
2. comprehensiveness and completeness of the artifact.
3. conformance to standards and guidelines defined for the type of artifact in the project plans.
4. efficiency and effectiveness of the solution presented by the artifact
5. clarity and correctness of the artifact.
6. achievement of all ancillary functionality that is expected of the artifact.
7. no unnecessary functionality is present in the artifact.
8. no malicious functionality is contained in the artifact.

1. the format of the artifact adheres to organizational standards or customer specification.
2. fulfillment of the requirements of downstream functions of the artifact when it is to be used downstream.

Tools and Techniques for verification:

- 1. Walkthrough (Peer Review)
- 2. Inspection - 3 types
- 3. Audit

#### Walkthrough

5 types:

- 1. Independent review
- 2. Guided review
- 3. Group review
  - \* postal review
  - \* meeting review
  - \* guided meeting review
- 4. Expert review
  - \* Domain expert (years of experience in same field)
  - \* Subject matter expert (expert in subject field like math, probability, botany, biology, physics...)
  - \* technology expert (highly skilled in development platform)
  - \* social expert (expertise in social behaviour, market forces, economy)
- 5. Managerial review:
  - \* performed by person who directly supervises the author of artifact.

This review is the final step before promoting the artifact to the next level in configuration management.

Objectives of IR

- using well developed groups of supervisors.
- ensures no required info is missing.
- ensures that all essential preceding activities have been performed.
- ensures consistency with other artifacts in the project
- ensures traceability with both upstream & downstream artifacts.

Artifact ready  
for walkthrough

Artifact passed in  
config. control

Identify persons to  
conduct walkthrough

conduct walkthrough

review report

No  
Any defect?

Yes

fix defects

No

Are defects  
fixed?

Yes

Place review in  
project records

walkthrough  
(completed)

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

fig: Independent  
walkthrough process.

**Inspection:**

- ensures that all required components are ready for the next stage.
- output of an inspection activity is an inspection report which specifies whether the system passed or failed.

Types:

- ① System Testing Readiness Inspection
  - ensures that all required components are in place for conducting the System testing.
- ② Acceptance Testing Readiness Inspection.
  - conducted as a prerequisite for obtaining the customer's acceptance of product delivery & thus for receiving payment.
- ③ Delivery Readiness Inspection
  - purpose is to ensure that the delivery package has all the components & that the right versions are included in it.
  - can be carried out in 3 modes
    - ⓐ Single shot delivery
    - ⓑ interim delivery
    - ⓒ final delivery

**Audit**

- used mainly in organizations that have a defined software development process that has been implemented in their projects
- audits are document verification system in which project documents & records are compared with the organization's standard

or defined processes.

- are short in duration (two hours)
- they ensure that a project is being executed in conformance with organization's defined processes.
- audits are conducted for the purpose of uncovering non-conformance (NC) in a project.
- an NC list lists all NCs uncovered during an audit.
- Auditor: people who conduct audit.
- Auditee: people whose project is being audited.

Types:

- ① Conformance Audit vs Investigative Audit:
  - focus on the efficiency of implementation of organizational process during execution.
  - to compare and contrast the project documents with organizational process.

while Investigative Audit

- focus on finding the causes for a failure or extraordinary success.

② Vertical Audit vs Horizontal Audit:

- |  |  |
|--|--|
| conducted across the organization on either a few selected projects or all projects, they focus on all aspects of projects | conducted across the organization on either a few selected projects or all projects; but they focus on only one aspect of project. |
|--|--|

③ Periodic audits vs Phase - End audits:	
- conducted at org level	- triggered by project events
- based on calendar duration	- executed when a project execution (major) phase is completed
- normally (2-3 months)	
- at end of every cycle of audit findings are presented to mgmt	Phases: x project initiation x S/W requirement analysis x S/W design x S/W construction x System testing x project closure

④ Internal Audit vs External Audits:	
- conducted by people internal to the organization	- conducted by an external agency that specializes in the audit process.
- independent of project being audited.	- certified personnel
- to ensure conformance or investigate the occurrence of special event.	can check the artifact classification: x precertification audits x certification & recertification audits. x Surveillance audits.

#### II Validation:

- The process of evaluating a system or component during or at the end of the development process to determine whether it satisfied specified requirements.

IEEE (standard 610)

"Confirmation that the product / as provided (or as it will be provided), will fulfill its intended use."

or

"The purpose of validation is to demonstrate that a product or product component fulfills its intended use when placed in its intended environment."

- CHM

"you built the right thing".

To achieve the full worth of validation, following 3 factors must be applicable:

- ① Validation is performed by independent persons who are not the same persons as those making the claim.
- ② Validation is performed not just against the specifications of the claimant, but also against external specifications.
- ③ Validation is a planned and coordinated effort performed for the purpose of substantiating a claim and instilling confidence in stakeholders; it is not performed by self-assurance.

\* Validation of Software Design:  
- difference in approach of validating different nature of projects (Software, automobile, electronics)

- Scale-down approach.
- Prototype approach / model
  - ↳ build and improve prototype
  - ↳ use and discard prototype.

\* Validating the software specification

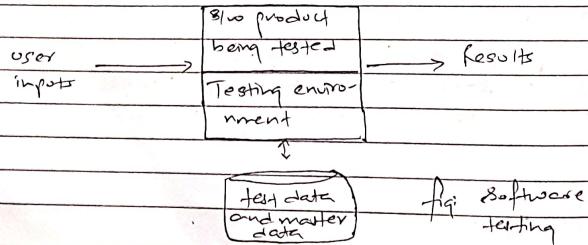
\* Validating the software product

Testing: "Process of executing a s/w item to verify that it satisfies requirements and to detect faults."

- British Standards  
Institution's standard

Software testing:

"Process of executing a s/w item to detect the differences, if any between its behaviour and derived behaviour".



- \* Testing different types of software products:
- \* Batch processing system: (data is submitted / entered offline; are process focused)
- \* Online system: (a.k.a event-triggered sys; interaction with GUI; focuses on test cases)

\* Real-Time System: (interacts with machines / hardwares  
have to use simulator in some cases; or have to create one)

\* Scientific applications:  
(built around complex mathematical equations  
that process large algorithm; are process intensive)

\* Mobile application:  
(either Simulator is needed or real devices)

\* Software Simulators:  
(type of processing system using special h/w  
simulates a real-life scenario).

\* Testing basics:

Black box testing:  
internal logic for processing the data is not considered

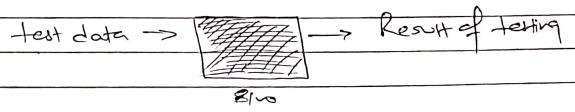


fig: Black box testing

White box testing / Glass box testing  
considers the internal logic & program statements of the S/W.

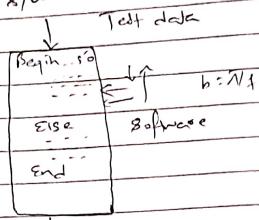


fig: white box testing

#### Approaches of testing

- Intuitive testing
- Process - driven testing

##### Intuitive testing:

- carried out by an experienced tester who uses his common sense.
- certain creativity and common sense is expected from tester.

User	<input type="text"/>
Phone	<input type="text"/>
Enrolled from	<input type="text"/>

##### Process - driven testing:

- testing is carried out in adherence with a defined validation process using a test plan & a set of test cases.

Software strategy is decided upon during project planning stage and is recorded in test plan.

##### Software Strategy:

- \* Productive objectives
- \* Customer acceptance objectives
- \* Product certification objectives

In addition to objectives, following are also part of test strategy:

- \* types of tests to be included in testing
- \* how to test
- \* regression testing
- \* criteria for successful completion of testing
- \* mechanism for defect closure & escalation when necessary.
- \* progress reporting
- \* defect analysis

##### # Test case Design:

"Test case is a set of test inputs, execution conditions and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirements, and as documentation specifying inputs, predicted results and a set of execution conditions for a test item."

- IEEE Standard 610.

### \* Test Plan formed (Book)

⇒ "condition table" is another way to describe test cases & describes the behaviour of the system for different combinations of inputs.

Condition	User enters information & clicks of
Valid ID & valid pwd	Accept
" " & invalid pwd	reject & prompt for valid pwd
Invalid ID & valid "	" " " " ID
" " & invalid "	" " " " ID & pwd
Empty ID and valid "	" " " " ID
Valid ID & empty "	" " " " pwd
Empty ID & empty "	" " " " ID & pwd

Table: Condition table for login screen.

- 1) Test Case design techniques
- 2) Equivalence Partitioning
- 3) Boundary Value Analysis
- 4) Error Guessing
- 5) Logic Coverage
- 6) Consistency checking
- 7) Requirements tracking
- 8) Response time checking

### 1) Equivalence Partitioning:

→ Input Space is partitioned into valid inputs & invalid inputs.

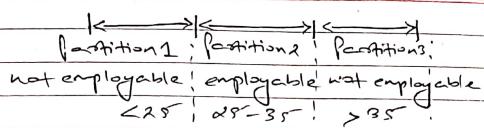
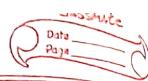


fig: Equivalence Partitioning

### 2) Boundary Value Analysis

①  $< 28$

②  $= 28$

③  $26 - 34$

④  $= 35$

⑤  $> 35$

(here  $= 35$  and  $> 35$ ) two test case are added.

→ used combination of equivalence partitioning & boundary value analysis (taking care of boundary points)

### 3) Error Guessing:

→ uses experience, intuition & common sense to design test cases that are likely to detect errors  
→ involves guessing technique

### 4) Logic Coverage:

→ to ensure that the logic is producing the desired results.

→ test cases either prove or disprove the logic

### 5) Consistency checking:

→ checks for consistency of processing from different points in a s/w.

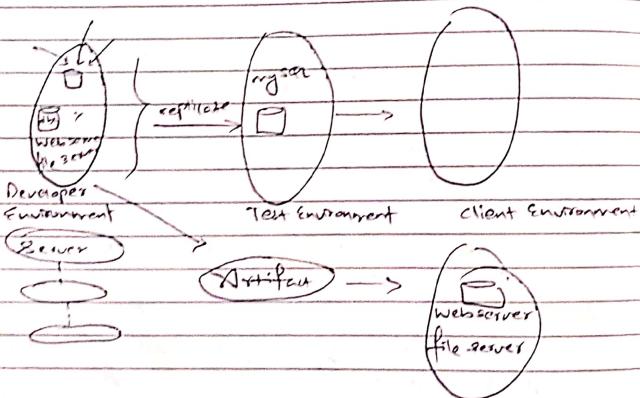
→ If there is difference in the values produced, it is a

clear indication of (cu). error in those S/W units.  
→ designed to ensure consistency of values.

x Requirements tracing:  
the requirements from the user requirements specification to the S/W detailed design are traced and the test cases to test the S/W product are designed to confirm that the S/W does indeed meet & fulfill the user requirements.

# Response time checking  
Response time - the elapsed time between when the customer gives a command and the system begins to display a response.

#### (e) Test Environment:



#### y Testing Scenario:

##### y Project Testing:

Carried out as part of a S/W development project to ensure that development work is defect free.

##### z Product Testing:

Carried out on a COTS (commercial off the shelf) S/W product to ensure that the product works up without any defects in a variety of customer scenarios.

##### x Project Testing: ~~unit testing~~

###### ↳ Unit Testing:

\* Always carried out by the person who wrote the code or by a peer using white box testing techniques.

###### ↳ Integration testing:

- carried out either as a one-off (when all integration is completed) or incrementally (one integration at a time)

\* one-off: black box

\* incrementally: white box

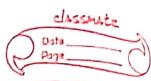
###### ↳ System Testing:

- carried out to ensure that the S/W works on all intended target systems

###### ↳ User-Acceptance testing:

- carried out to obtain customer sign-off so that the S/W can be delivered and payment

Received



- x Product Testing : ~~Hi-Fi testing~~ Extra Q3  
- The product is to be marketed to multiple customers & delivered with or without customization.
- + a product is developed as a project first and undergoes all tests that a project normally undergoes (unit integration, system)

y Load Testing: large no. of users are logged in & use the software in a random manner.

z Volume Testing: subjects the software to a high volume of data to see if performance degrades as the amount of data increases.

4. functional testing: to ensure that all its functions are working correctly.

5. End-to-end testing: one entity in the application is tracked from birth to death. e.g.: life insurance company's Software.

6. Parallel testing:  
a number of users access the same function with same data.

7. Concurrent testing:  
to eradicate any issues that occur when two or more users access the same functionality with different data at same time.

7.1 Stress Testing:

Stresses the system by making expected resources available causing deadlock scenarios, not isolating resource disconnecting network to ensure that software handles such stress/ conditions.

7.2 Positive Testing: to ensure that all defined functions are performing as expected  
- not designed to uncover defects.

7.3 Negative testing: uncovers deficiencies in the software focused on error handling and facilities improvement.

7.4 User Manual testing:  
involves using the S/W in conformance with the user manual to ensure that manual and S/W are in synchronization with each other.

7.5 Deployment Testing:  
Simulates the target environment and deploys the S/W to ensure that deployment specified is appropriate.

(clear)  
7.6 Sanity testing:  
ensured that that the components of software package are complete and versions appropriate.

13) Regression Testing:  
Carried out after defects uncovered by earlier testing has been fixed - previous test is not repeated but only defect report is generated.

14) Retesting: conducting an entire test again  
- to ensure that the process of fixing defects has not introduced any fresh errors in software.

15) Security Testing:  
check against the threat of viruses, spyware

16) Performance Testing:  
- evaluate overall performance of the system.  
- indicates response time, turn around time, report generation.

17) Usability Testing:  
involves testing S/W for different type of usages to ensure that it satisfactorily fulfills the requirements of specified functional areas.

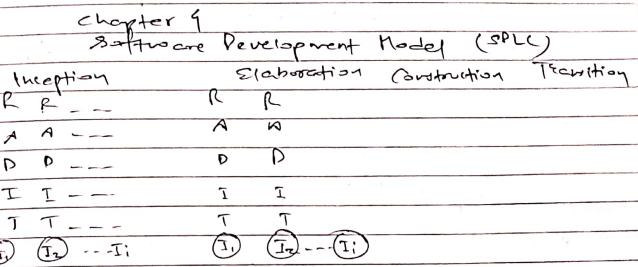
18) Install-Uninstall Testing:  
- to ensure that install/uninstall functions are satisfactorily performed.

19) Comparison Testing:  
involves testing of the product with competing products to identify difference and to determine the relative position of the product against competing product.

20) Regular Conformance testing:  
ensure that the product conforms to all government regulations pertaining to minimum standards.

21) A-testing: testing within orgn.

22) B-testing: testing without orgn.



⇒ 4 phases in SDLC (Inception, Elaboration, construction, Transition).

⇒ 6 processes/phases within each above phase (Requirement gathering, Analysis, Design, Implementation, Testing, Deployment, Maintenance).

⇒ Various methodological or S/W development model like:

- ① Waterfall model
- ② Iterative model
- ③ V model / verification / validation
- ④ Incremental model.
- ⑤ Spiral model
- ⑥ RAD model
- ⑦ Prototype model
- ⑧ Agile model