Lecture No - 30

UNIT - 4

Software Testing

Content: Basic concepts of testing

Testing -

→ It is a process of executing program with the intent of finding errors.

→ It identifies all the defects in a program.

→ Effective testing will lead to the * delivery of quality s/w.
* More satisfied users, * lower maintainence cost,
* more accurate & reliable results.

→ Ineffective testing will lead to opposite results.

→ Hence it is an essential activity.

Testing Objectives -

→ Detect S/w failures
→ Verification & Validation
→ data collection during testing provides indication of s/w reliability & quality.

S/w Testing Principles -

→ Should depend on users
→ It is impossible to test everything
→ Test planning should be done early.
→ Test for invalid and unexpected input conditions as well as valid conditions.
→ Testing time & resources are limited.
→ Testing should began at module.
→ Testing should be done by an independent 3rd Party.
→

b) Local Data Structure -

→ To ensure that the temporary stored data maintain its integrity while an algorithm is being executed. ~~at least~~ once.

c) Boundary Conditions -

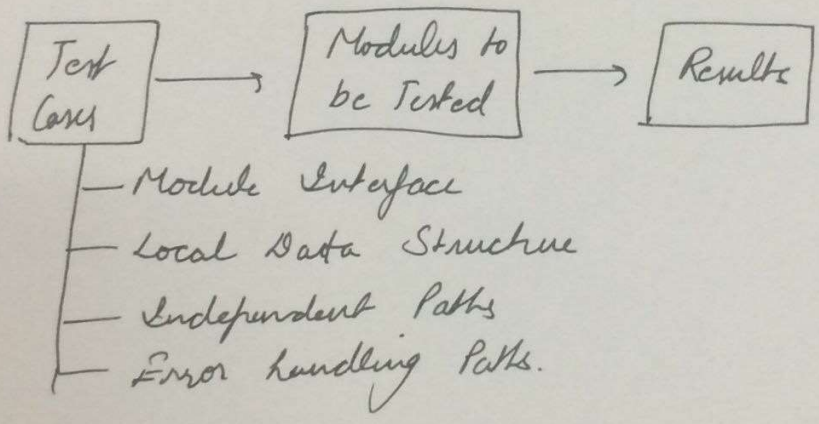
→ To ensure that the module operates as desired within the specified boundaries.

d) Independent Paths -

→ To ensure that all statements in a module have been encuted atleast once.

e) Error handling Path -

After completion of other it is done.



Advantages -

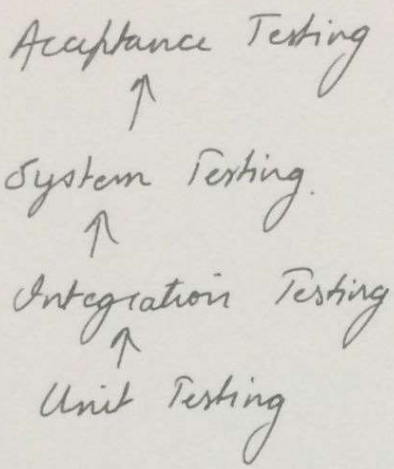
→ Facilitate changes
→ Simplifies Integration
→ Documentation
→ Design.

Disadvantage -

→ time consuming
→ It demands patience & thoroughness on part of development team.
→ Rigorous documentation is required

→ Assign best person to the task. of domg test.
→ Testing should begin in the small and progress towards the large.

Levels of Testing -



① **Unit testing** -
→ It is a s/w development process in which the smallest testable part of the application called units are individually and independently tested for proper operation.
→ It is taken after a module has been coded & reviewed.
→ It is a dynamic method for verification where the program is actually compiled & executed.
→ Different modules are tested against the specifications produced during the design of the module.
→ Goal is to test the modules or units not the whole S/w system and to test internal logic of the modules. (White Box oriented)

Types of Unit Testing -

@ **Module Interface** -
→ Tested to ensure that information flows in a proper manner into and out of the unit under test.

Lecture No - 31

Contents :- Integration Testing, Bottom-up Integration Testing

Integration Testing -

→ Primary objective is to test module interface in order to ensure that there are no errors in the parameter passing, when one module is invoked by another module.

→ It is a systematic technique to build the program structure and at the same time test are done to uncover errors related to interfacing.

Target → testing the interface

Types of Integration Testing -

- Big Bang
- Top-down
- Bottom-up
- Sandwitch Integration

Big Bang - (Non Incremental)

→ In this all or most of the developed modules are coupled together to form a complete S/w system. or major part of the system & then integration is done.

Advantage -

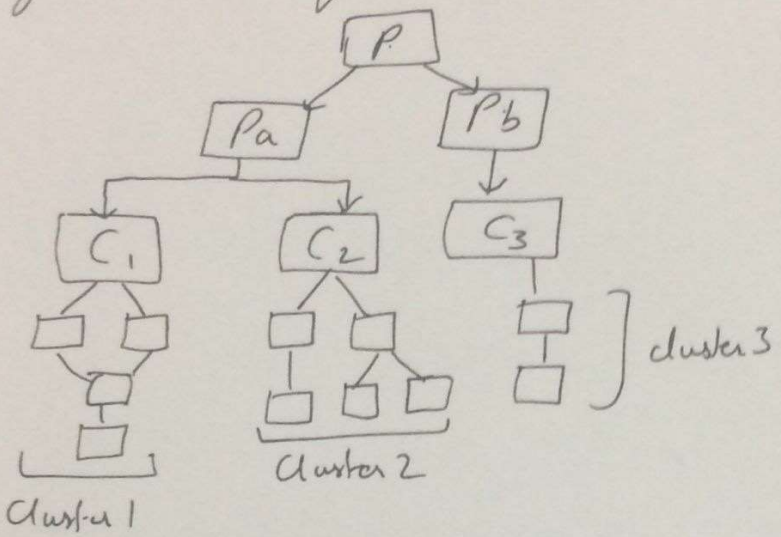
All modules are integrated at once. in a single step.

Disadvantage -

Not suitable for large products (locating a fault is difficult)

Bottom-Up Testing:-

→ lower level components are tested first then used to facilitate the testing the higher levels. components.
→ The process is repeated until the component at the top of the heirarchy is tested.



Advantages

→ More satisfying
→ Behaviour of modules at lower level are verified earlier.
→ No stubs are required if relies on routines only.
→ Uncover errors that occur at lower level in S/w.
→ Modules are tested in isolation from other modules.

Disadvantage

→ Delays in verification of modules at higher level.
→ Complexity in integrating large S/w.
→ Expensive

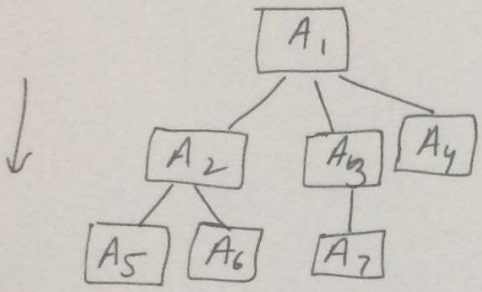
Lecture No -32

Content:- Top down Integration Testing.

Top Down Integration Testing-

→ It is an incremental approach for building program structure as we move downward through control hierarchy.

→ Modules are integrated and subordinates to the main control module are included into structure.



Advantages-

→ Behaviour of modules at high level is verified early.
→ None or only one driver is required.
→ Modules can be added one at a time in each step.
→ Supports both depth first and breadth first method.

Disadvantages

→ Delays verification of behaviour of modules present at lower level.
→ Large no. of stubs are required in case the lowest level of S/W contains many functions
→ Modules can't be tested in isolation because one has to invoke other module.

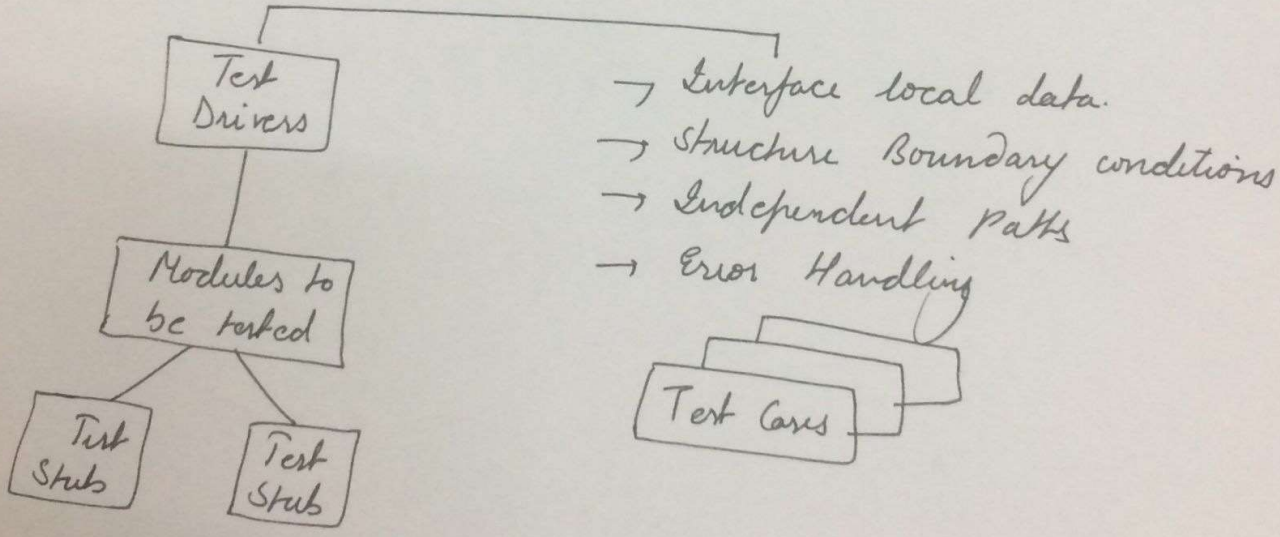
Sandwitch Testing-

→ It is an approach to combine top-down & bottom-up testing. The combination takes advantages of best features of both the approaches, while eliminating the weaknesses.

→ Whole system is divided into 3 layers just like sandwitch. The target is in middle. 1 layer is above and second is below the target.

→ Both stubs and drivers are requried in the testing.

Stubs & Drivers-



→ Interface local data.
→ Structure Boundary conditions
→ Independent Paths
→ Error Handling

Lecture No -33

Contents :- Acceptance Testing, Regression Testing, System Testing

Acceptance Testing -

→ Also called as user acceptance testing.
→ It is carried out to verify if the product is developed as for the standards and specified criteria.
→ It checks for all the requirements specified by the customers.
→ It is carried out by the users / customers when the product is developed externally by another party.
→ It falls under the category of black box testing methodology where the users are not very much interested in internal coding of a s/w.

Advantages -

→ Gives the user an opportunity to ensure that s/w meets the user req requirements beforehand. (actual delivery)
→ Enable both user & s/w developers to identify and resolve problems in a s/w.
→ Determines the readiness of a s/w to perform operations.
→ Decrease the possibility of a s/w failure to a target extent.

Disadvantages -

→ Although user provide a valuable feedback. They don't have the detailed knowledge of the s/w code.
→ Since testing is not user primary operation they mail fail to observe or accurately report some s/w failures.

Regression Testing -

→ It involves re-execution of the same test-cases to ensure that there have been no side effects because of the changes.

→ It is the process of ~~changing~~ testing changes to computer programs to make sure that the older programming still works with the same new changes.

→ It attempts to verify that app works as specified even after the changes additions / modifications were made it.

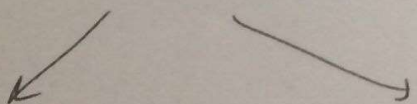
→ It maintains the original functionality and continues to work as specified even after changes / addition / modification to the S/w application.

System Testing -

→ This testing is performed ~~&~~ when S/w integration is over and successfully tested. The S/w product is now ready for inspection from the customer point of view.

→ In this overall modules are tested as a whole (complete system as one module)

Types of User acceptance Testing -



(α - testing)
Alpha - testing

(β testing.)
Beta testing

→ Alpha - testing

→ It is done at developers site by the customer.

→ It is conducted in a controlled environment with developer looking over the shoulder.

→ During α testing developers records errors and usage problem.

Beta testing -

→ It is done at one or more customer's site by the end user of the S/W.

→ It is conducted in environment that ~~consists~~ can't be controlled by the developers (not an ideal environment)

→ During Beta testing customer records all problems and submit report to the developers for modifications at regular intervals.

Lecture - 34

Contents :- White-Box Testing, Various Types of white Box testing.

Structural Testing (White Box Testing)

→ It is an approach to testing where the test-cases are generated on the basis of the actual code of the program or module to be tested.

→ Also called as white box, glass-box, logic driven or path-oriented, clear box testing.

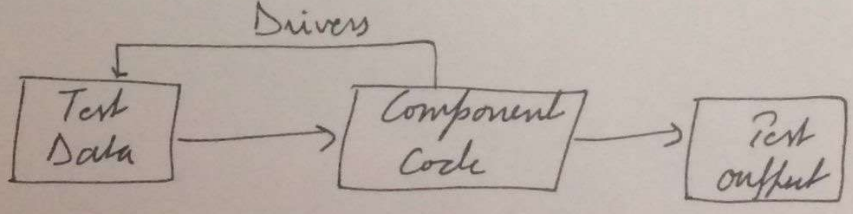
→ The testers should have complete knowledge of the internal structure of the s/w.

Advantages -

→ Covers the larger part of the program code while testing.
→ Uncovers typographical error.
→ Detect design errors.
→ Help to optimize the code.

Disadvantages

→ Time consuming
→ expensive
→ Test based on design may miss other system problem.
→ Test cases need to be changed if implementation changes.
→ Test that covers most of the program code may not be good for accessing the functionality of unexpected behaviour and other testing goals.



White Box Testing Techniques -

→ **Statement Coverage -**

This technique is aimed at exercising all programming statements with minimal tests.

Statement Testing = (No of Statements Executed / Total No of Statements) X 100 %.

→ **Branch Testing -**

This technique is running a series of tests to ensure that all branches are tested at least once.

BT = (No of decisions outcomes tested / Total No of decisions outcomes) X 100 %.

→ **Path Coverage -**

= (No of paths exercised / Total No of paths in the program) X 100 %.

This technique is corresponds to testing all possible paths which means that each statement and branch is covered.

→ **Data Flow Testing -**

→ Track the specific variables through each possible calculation, thus defining the set of intermediate paths through the code.

→ It reflect dependencies, mainly through sequences of data manipulation.

→ Each data variable is tracked and its use is verified.

→ **Loop Testing -**

→ It relate to testing single loops, concatenated loops and nested loops.

→ Independent and dependent code loops and values are tested by this approach.

Black Box or Functional Testing -

→ Also called as exterior / specification / data driven or input / output driven testing.

→ It is an approach of testing where the test are derived on the basis of requirements of the program or module.

→ Focuses on functional requirement.

→ Dosnot bother about internal structure. (No analysis of code)

→ It serves to detect deviations of a test object from its specification.

→ Performed in later testing phases.

→ It only observes the o/p for some i/p values which does not analyse the code.

Advantages -

→ Tester requires no knowledge of implementation

→ Efficient for large S/w.

→ Test are done from users point of view.

→ Test cases can be designed as soon as the req^{n} is complete

→ Reveals any ambiguities & inconsistencies in the functional specification.

Disadvantages -

→ Test cases difficult to design.

→ Leaves many program paths untested.

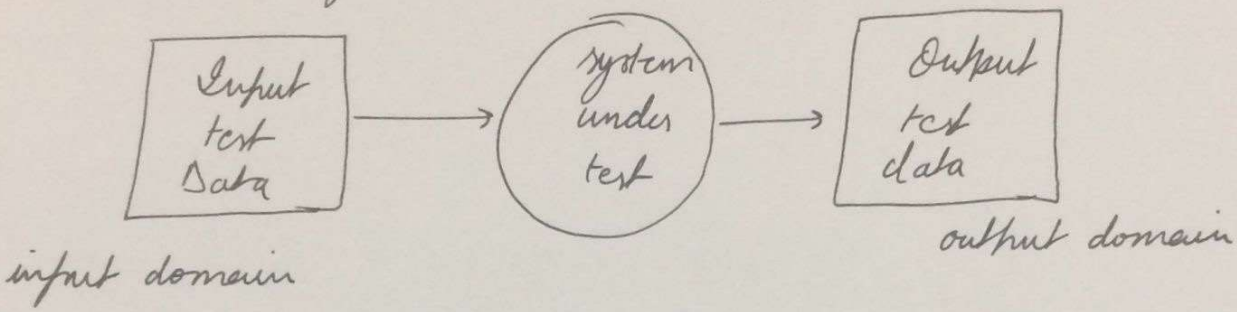
→ Cann't be directed towards specific segment of code. hence it is more error prone.

→ Only a small no. of possible inputs can be tested as testing every possible input consumes a lot of time.

Lecture No - 35

Contents: Black Box Testing & Various Types of Black Box Testing.

Continuation of Black Box testing -



Black box testing

- Different techniques of Black Box testing -

① Boundary Value Analysis -

→ test cases that are close to boundary conditions have higher chances of detecting errors.

→ Boundary conditions means an I/P value may be on the boundary.

ie just bellow the upper limited and just above the lower limit

- eg → 2 i/p real variables x & y.

I/P variable x with a range from 1 to 100.

BV = 1, 2, 99, 100

$a \leq x \leq b$, $c \leq x \leq d$.

x bounded by interval [a, b]

y bounded by internal [c, d]

→ a, a+1, b-1, b

→ c, c+1, d-1, d.

② Robustness Testing -

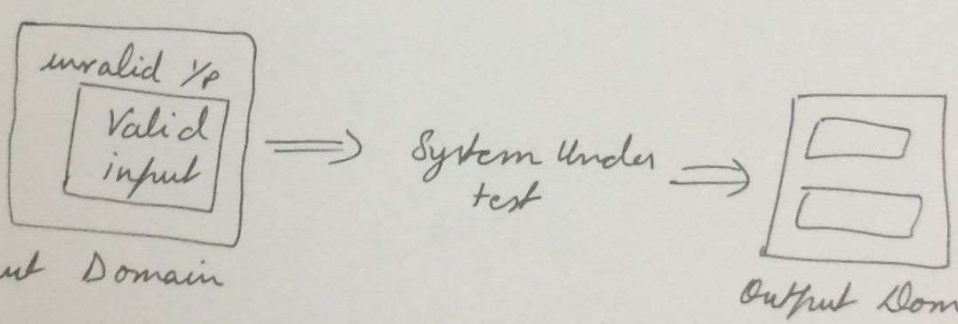
→ Extension of boundary value analysis.

→ In this type of testing the extreme values are encluded with a value slightly greater than maximum value & slightly less than the minimum value.

$a \leq x \leq b$
$c \leq x \leq d$
→ $a-1,\ b+1$
→ $c-1,\ d+1$

(3) Equivalence Partitioning -

→ In this I/P domain of a program is partioned into a finite number of equivalence classes such that one can reasonably assume, not be absolutely sure that the test of a representive value of each class is equivalent to a test of any other value.



for a program that is supported to accept any number between 1 and 99., there are atleast 4 equivalence classes from I/P side. They are:-

i) Any number b/w 1 and 100 is valid.
ii) Any number less than 1. (0 and all -ve value)
iii) Any number greater than 99.
iv) If it is not a number, it should be accepted.

Equivalence classes to be defined on the basis of I/P & O/P domain.

Lecture No -36

Contents:- Test data, Test Case, Test Suite

Test Case -

Test case is the triplet [I, S, O]

I = data input to the system

S = State of the system at which data is input

O = the expected output of the system.

→ Test cases describes an input description and an expected output description.

Test Suite -

→ A set of all test cases with which a S/W product is tested is called a test suite.

→ A good test case has a high probability of finding an error and hence the same goes with the test suite.

→ We may have a test suite of effective or good test cases.

→ We may have a test suite of all possible test cases.

Hence any combination of test cases may generate a test suite.

+ There are essentially 2 main approaches which we have studied to systematically design the test cases.

* Black Box testing
* White Box testing

→ Who should do testing?

→ Testing requires the developers to find the errors from their software.

→ It is difficult for S/W developer to point out errors from own code creations.

→ Many organisations have made a distinction between development & testing phase by making different people responsible for each phase.

→ What should we test?

We should test the programs responses for every possible input. It means we should test for all the valid and invalid inputs

Different terminologies -

→ Errors → syntax error or misunderstanding of specification
→ Bugs → When developer makes mistake while writing code
→ Failure → It occurs when fault executes.
→ Fault → manifestation of an error (bug/defect).

Lecture No-37

Contents:- Static Testing Strategy:- FTR, Walk Through.

Static Testing Strategy

→ Static Testing means that the analysis of the program is performed with executing them.

→ A static Analyser operates from pre-computed database of descriptive information derived from the source test of the program.

Various S/w Testing strategies are -

① Formal Technical Review - (FTR)

→ A FTR is a form of he review in which a team of qualified person examine the suitability of the S/w product for its intended use and identifies any conspiracy between specifications and standards.

→ It can be conducted at any stage of the S/w development life cycle.

→ Purpose of conducting review is to minimize the defect ratio as early as possible in S/w Development life cycle.

→ Review can be formal or informal.

→ Informal reviews are referred as walk through and FTR is actually a class of reviews that includes Walkthrough, code inspection, round robin review and other small groups technical assesment of the S/w.

Objective of FTR- (Peer Reviews)

→ To uncover errors in function, logic or implementation for any representation of the S/w.

→ To verify that the S/w under review needs its requirements.

→ To ensure that the S/w is developed in a uniform manner.

→ To make project more manageable.

→ To ensure that the S/w has been represented according to pre-defined standards.

② Walk-Through-

→ Method of conducting informal group / individual review is called walk-through.

→ It can be pre-planned or can be conducted at need basis.

→ Generally people working on the work product are involved in the walk through process.

→ The purpose of walkthrough is to find problems and discussion over the alternate solution.

→ IEEE recommends 3 specialist roles in a walk-through focusing on demonstrating that work product meets all the requirements. They are :-

① Leader -

→ One who conducts the walk through.

→ Handles administrative tasks and ensures orderly conduct.

ii) Recorder -

→ Who note down all potential defects, decision and action items and identify during the walk through meeting.

→ He/She generates minutes of meeting at the end of walk-through session.

iii) Author -

→ One who represents the S/W product in step by step manner at the walk through meeting and is probably responsible for computing most action items.

Lecture No - 38

Contents:- Code Inspection, Coding Standards & Guidelines

Code Inspection -

→ A code inspection is a formal, rigorous, indepth group review designed to identify problems as close to their point of origin as possible.

Objective of Code Inspection -

→ Finds problems at the earliest possible point in the s/w development process.
→ Verify that the work product meets its requirements
→ Ensure that the work product has been presented according to pre-defined standards. Provide data on product quality and process effectiveness.
→ Increases the effectiveness of s/w testing.

IEEE represents following role in an inspection -

* **Inspection Leader -**

It ensures that the inspection dod is conducted in an orderly and meets its objective.

* **Recorder -**

The recorder should record inspection code is conducted in an orderly and meets its objective. is required for process analysis. The inspection leader may also be the recorder.

* **Reader -**

Lead the inspection team through the s/w

product in a comprehensive and logical fashion.

* Author -

He is responsible for the S/w product meeting its inspection, entry criteria for contributing the inspection based as special understanding of the S/w product and for performing any rework required to make the S/w product meets its inspection criteria.

* Inspector -

→ He shall identify and describe anamolies in the S/w product and shall be chosen to represent different view points at the meeting.