Software engineering Unit 4 (KCS601)

Software Engineering (Dr. A.P.J. Abdul Kalam Technical University)

UNIT-4

SOFTWARE TESTING :-

Software testing is the process of executing a program to locate an error. A Good test case is one that has a high probability of finding undiscovered error. So, losses due to it can be save. Testing is the fundamental of software's sucess. Testing is used to show incorrectness and considered to success when an error is detected.

Objectives of Software Testing :-

① Software (S/W) quality improvement :-

S/W testing is not only used to remove bugs, but also to find out design defect by the programmer. As for a complex system design perfection is one time is not possible. So, testing should be done to make the system perfect.

S/W quality mean that conformness to the specify the s/w design requirements. It include

(i) Being Correct

(ii) Minimum requirement of quality.

Now a days, computers and s/w are used in critical applications the outcome of a bug can be serve

(3) S/W reliability estimation :-

S/W reliability has important relationship with many aspect of S/W development. Its objective is to discover the designing error before delivery to the customer. The failure data during the testing process are taken down in order to estimate the S/W reliability. The testing process may function with regular feedback from the reliability analysis to the testors and designers as shown in figure 1

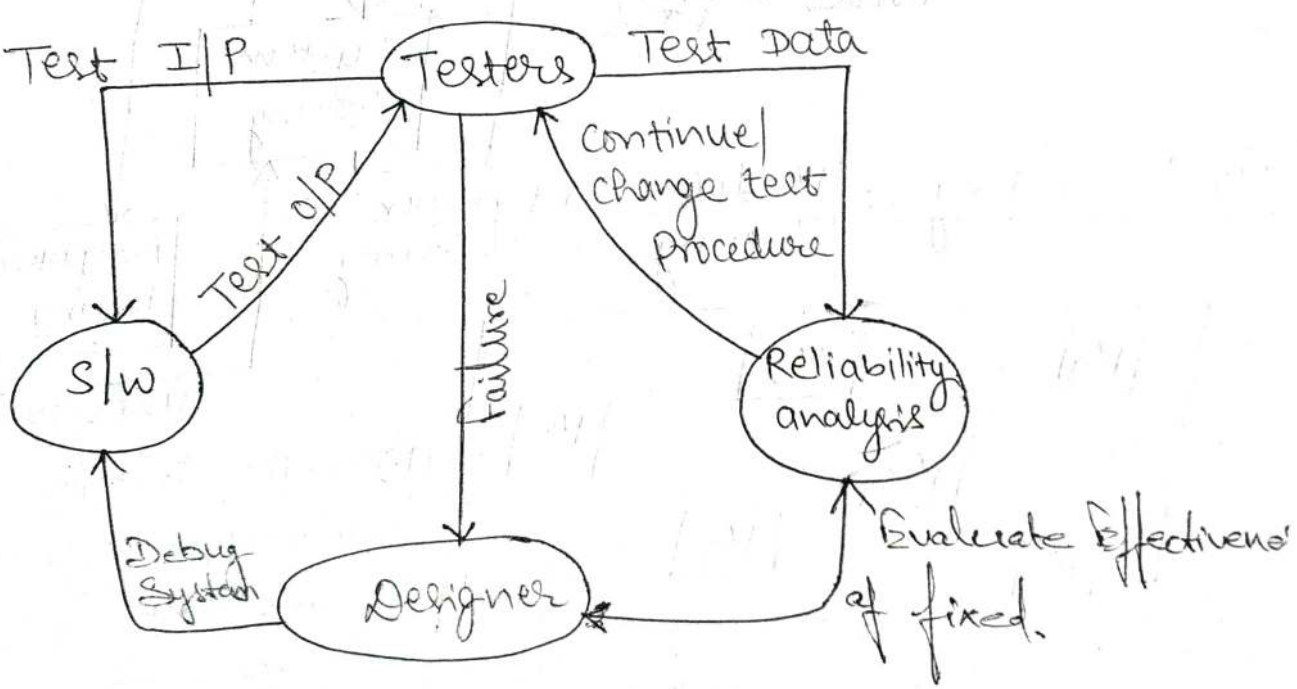
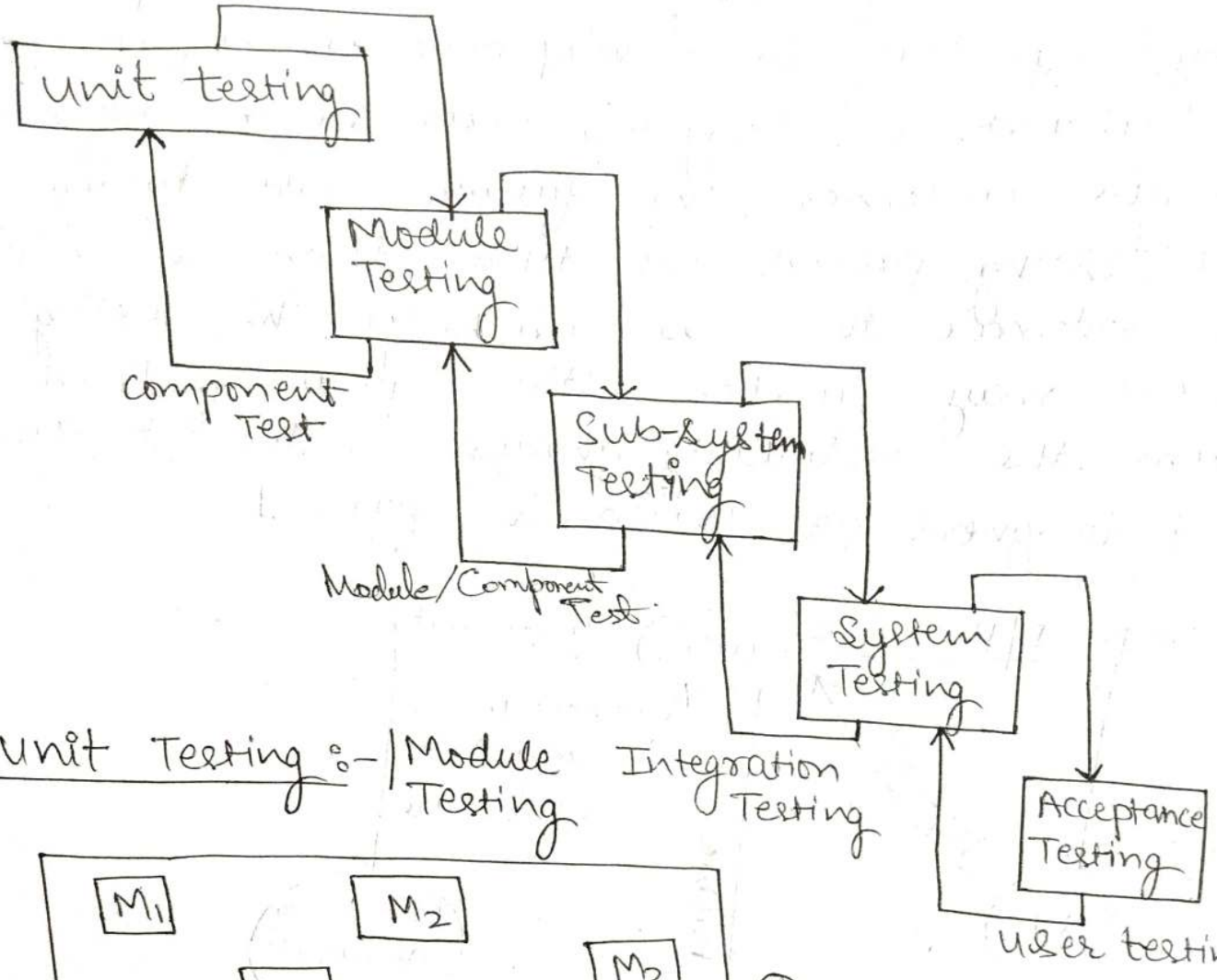
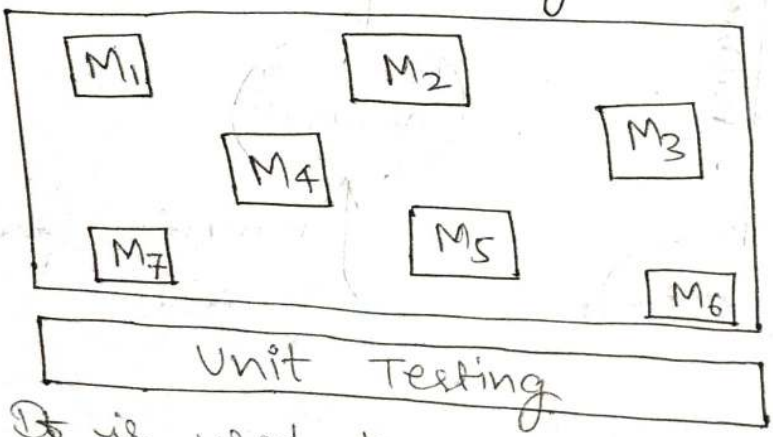


Figure No. 1

Level of Testing :-



① Unit Testing :- / Module Testing



1. Unit testing focus verification affort on the small unit of s/w design. The s/w component or module is used to uncover error with in the boundary of the module.
2. Unit testing is the white box testing oriented
3. In unit testing individual components are tested to ensure that they are working properly in all aspects as required. thats why, at start of the phase

it is also called component testing.

④ It is lowest level of testing.

⑤ Unit testing is typically conducted by developme team and programmer who coded the unit. It for ensure that unit meets its requirement

The main reason for doing unit testing are :-

(i) The size of single module is small enough that can locate errors very easily.

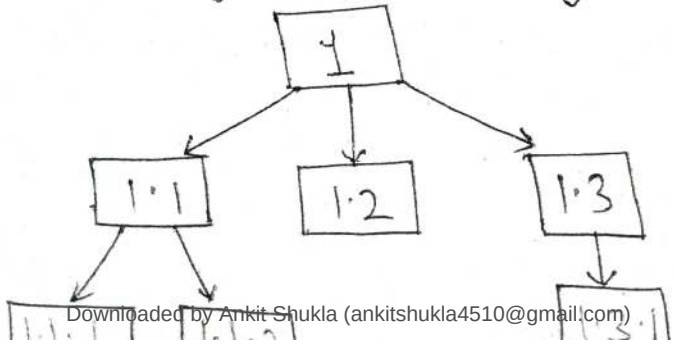
(ii) Confusing interaction of multiple errors which occure when many parts tested together is eliminated.

② Integration Testing :-

Once individual program component | modules have been tested, they must be integrated to create a partial system

It is used to check the module interface that is a way to interaction of module, is not having any kind of errors. The main integration technique are as follows :-

(i) Top-down Integration Testing :-



Top-down testing is an integral part of a top-down development process, where, the development process starts with high level components and work down the component hierarchy.

After the top level component has being programmed and tested, its sub components are implemented and tested in the same way. This process continue until the bottom level componew are implemented or Tested.

Advantages :-

(i) Design error are detected as early as possible, saving development time or cost because correctio in the module design can be made before their implementation.

(ii) The s/w system can be tested through-outly from the start with test cases. without providing (expensive) test environment.

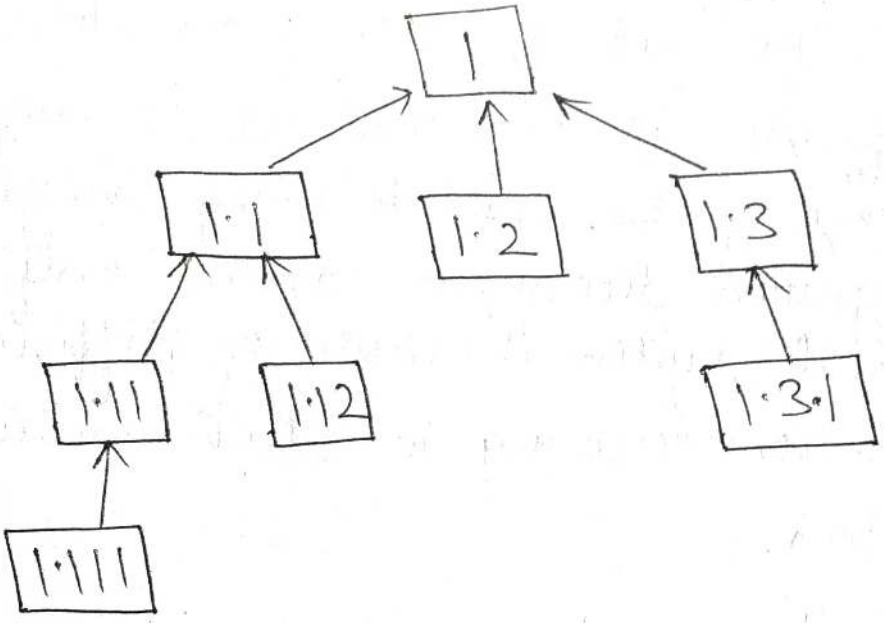
Disadvantages :-

(i) strict top-down testing proves extremely difficult.

(ii) Error in lower hierarchy levels are hard to localized.

(iii) ~~Bot~~

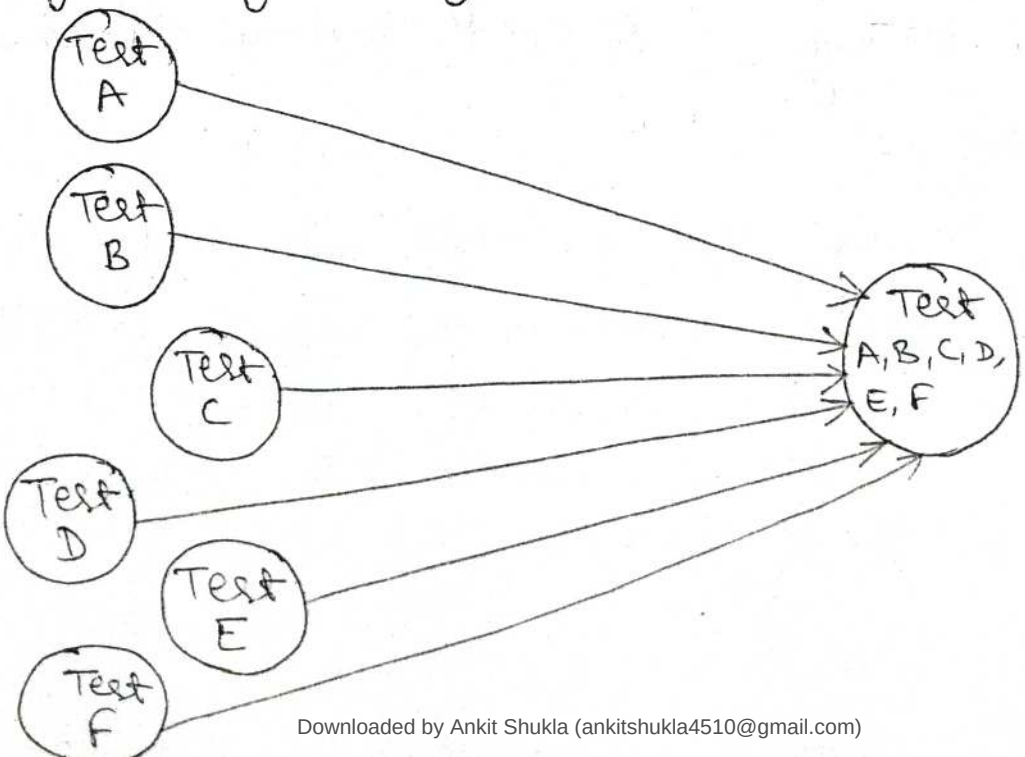
(ii) Bottom-up Integration testing :-



Disadvantage :-

1) Design error in the upper level are detected very late.
2) Testing individual level also inflicts high cost for providing a suitable test environme

(3) Big-Bang Testing :-



It is non-incremental integration approach where all modules making a complete system are integrate in single step. The set of error encountered there are very difficult to localised as it may potentiall belongs to any of the module being integrated. So, this integration techniques can be only used for very small system because of difficulty in identify the error occuring in which module interfac of large system.

Describe α and β testing :-

The acceptance testing is a kind of testing conducted to ensure that the s/w works correctly in the user work environment.
The acceptance testing can be conducted over the periods of weaks and month.

The type ~~acceptance~~ testing are :-

1.) Alpha testing 2.) Beta testing 3.) Acceptance Testin

Alpha Testing :-

A lot of testing s/w products is done to ensur about the system performance before delivering the s/w product to the customer. But, it is not possik to predict how the customer will really use program/module or system.

while operating manuals and user manuals, are explain and given to user. Even then their are possibility of misinteruption / of misinterpretation of instructions, complex data combination that is no understable to user, the output produce by system in user environment is not up to the mark though tester found it ok when he/she tested the system.

To solve this problem customer is called at developer site to test the system. This type of testing is called alpha testing. In alpha testing the s/w is used in its future environment on which it has to work on user site. It is done at developer site. So, it is described as internal testing process.

Meaning :-

The alpha testing is a testing in which the version of complete s/w is tested by the custom under the supervision of developer This testing is perform at developer site.

(2) Beta-Testing :-

The beta testing is a testing of s/w is done at user site in real world environment. This type of testing is done when developer is

quite confident on the performance of their system and think it is ready for final delivery. Means ready to release to real customer. This testing is done out of developing environment of software. So, it is described as external testing.

Means :-

The Beta testing is a testing in which the versio of s/w is tested by the customer without a developer being present. This testing is perform at customer site, as their is no presence of developer during testing, it is not control by developer. The end user records the problem and repost them to the developer. The developer then make appropriate modification.

Regression Testing :- Testing is used to find out erro in the s/w due to which it is not working proper when these error are find out then different methods are used to connect these error but while removing process of current errors some new faults or error may be generated. Regression testi is used to identify new error or fault. In other means, regression testing identify new faults that may have been introduced as correct ones are being corrected.

means:-

Thus, Regression testing is used to ~~while operating~~ reduce the site effect of changes.

Regression testing be applying in development phase as well as maintainence phase of s/w life cycle.

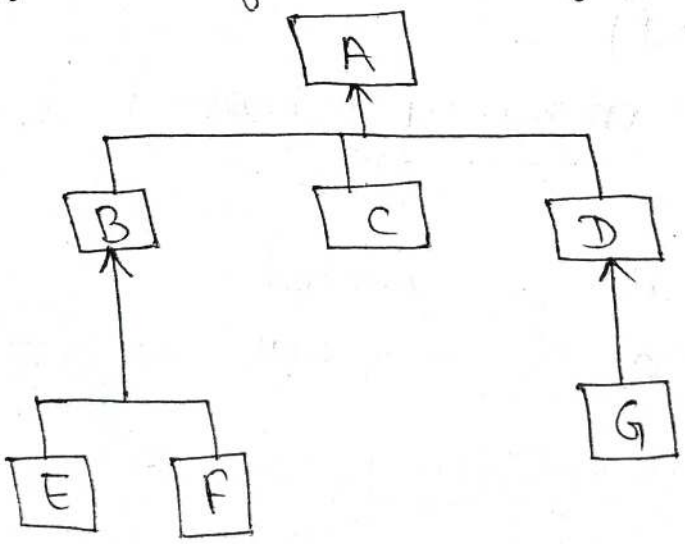
(1) Development ~~Phase:~~ In it regression testing is done after correcting the error found during the testing of s/w.

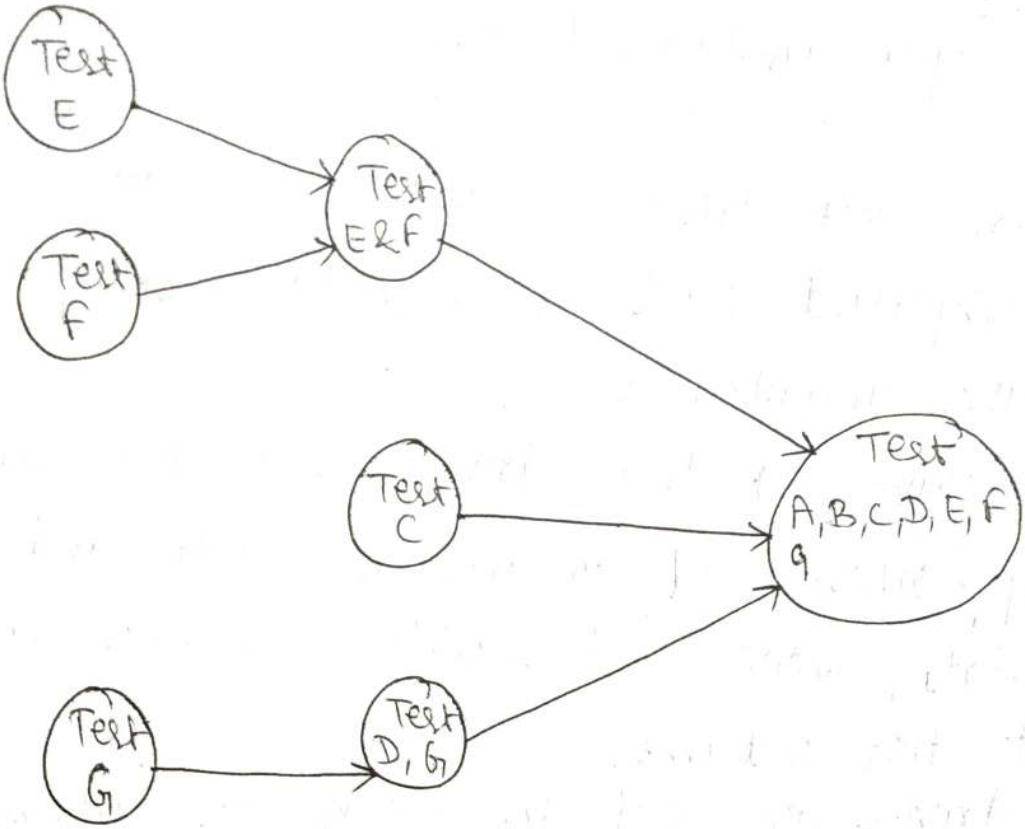
(2) Maintainance phase:- In this phase of s/w life cycle adaptive, corrective and maintainence is done due to which some modification is done in the s/w. and these modification may be cause of new errors. To find out these errors regression testing is used in maintainence phase of the software (s/w).

Q. What do you understand by Test driver & Test stub ?

Ans The term test driver & test stub play an important important roll in integration of system. They are described as ....

1) Test Driver → A test driver is a S/W module or application used to invoke a test and provide test data, control and monitor execution an report test outcome.
Test driver are used for testing of sub modules in the absence of main control module.
A component driver routine calls a particular component and passes test cases to it. The driver may be written for a unit (module) or for integration (for combining) the module





If each individual model E, F, G are working correctly, we move to next level. Unlike the lowest level component the next level compone are not tested separately. Instead, they are combined with components they ~~call (which has~~ already tested).

Advantage of bottom-up (Driver base integration technique) :-

1) No test stub is needed.
2) Error in critical modules are found easily.

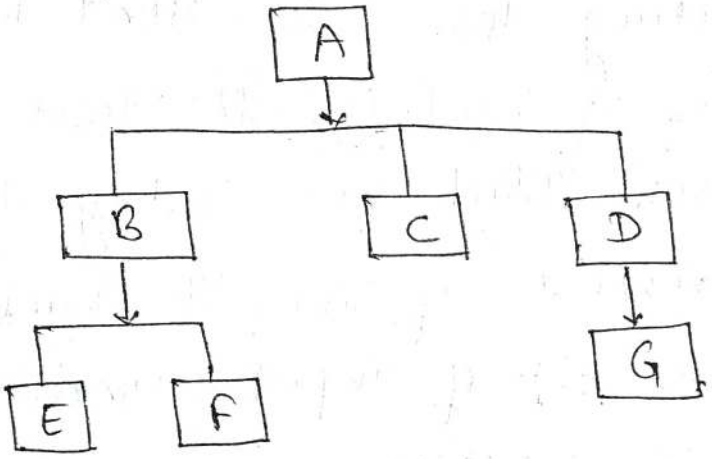
Disadvantage of bottom-up :-

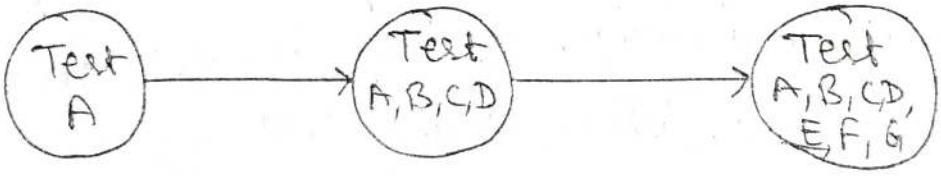
1) More test driver needed.
2) Interface error are discover later.

Test Stub :-

It is specialized implementation of element used for testing purpose, which are dumy of real component. Test stubs are programmes or components that have deterministic behaviour and are used to interface with Sub system in order to take care of dependancies.

Basically, stub are used in top-down approach. In it the main control module is tested in the absence of sub-modu





For example :-

If we want to test a module A but it needs module B, C, D. In this case, we creates dumy modules for B, C, D. These all are called stub and used to run A.

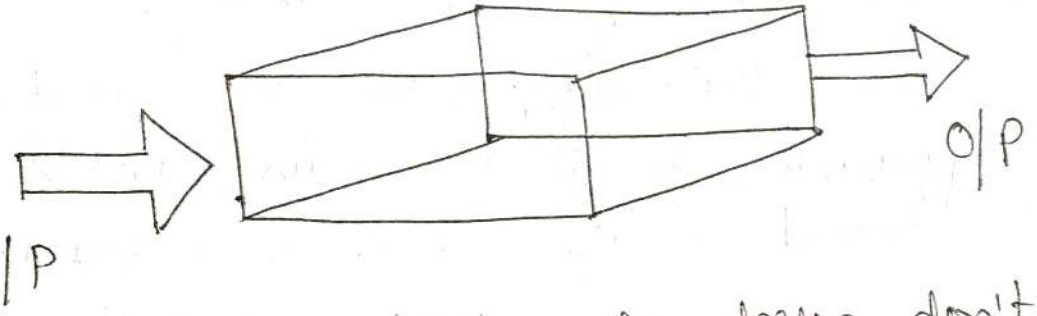
Advantages of top-down (stub base integration technique)

1.) No need of test driver.

Disadvantage of top-down (stub base integration technique)

2.) It requires lot of stub writing.

Black Box Testing (Functional Testing) :-



In black-box testing the tester don't know the internal structure of module. It tests only for I/P / O/P behaviour. Black box testing focus on the functional requirement of S/W. It enable the S/W engineer to drive set of inputs condition that will fully exercise all functional requirement for a program, without knowledge of its inner structure.

The black-box testing is also known by its name by —

1.) functional test
2.) Experior testing.
3.) Specification testing.
4.) Data driven testing. and
5.) I/P & O/P driven testing.

White-Box Testing Technique:- (Structural Testing / Glass Testing)

- Basic-Path Testing
- Control Structural Testing
  - conditional Testing.
  - Data flow Testing.
  - Loop testing.

Basic-Path Testing:-

This method enable the designer to drive a logical complexity major of procedural design and use it as a guide for defining a basis set of execution path. Exp. matter of cyclomatic Complexit

Control Structural testing:-

The following are some important type of control Structural testing

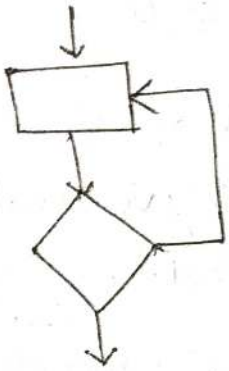
(1) **Condition testing:-** Its testing is a test case design method that is used for checking a logical condition contain in a program module. A simple condition is a Boolean Variable or relational expression which include NOT (~), EQUAL To (=), LESS-THAN (<), GREATER-THAN (>), LESS-THAN EQUAL To (<=), GREATER THAN EQUAL To (>=). Therefore, types of condition include the

following Boolean operator. Boolean variables error, Boolean paranthesis error, relational operator error, Arithmatic expression error.

(ii) Data-Flow Testing :- Data flow testing focus on the point at which variable receive the values and the point at which these values are used.

(iii) Loop-Testing :-

Simple Loop

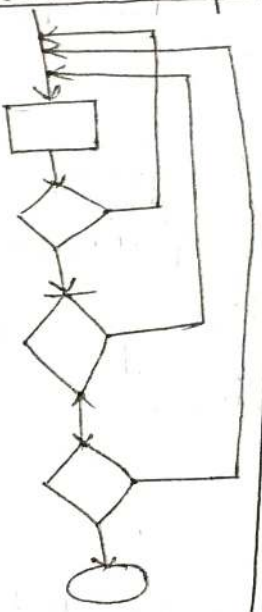


(i) if n=0 that means skip the loop completely.

(ii) if n=1 that means, 1 pass through the loop is tested.

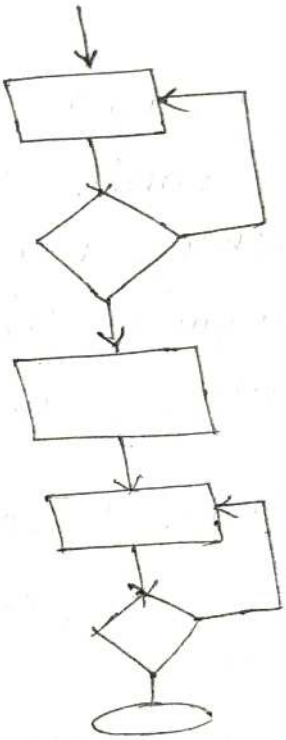
(iii) if n=2 that means, 2 pass through the loop.

Nested Loop



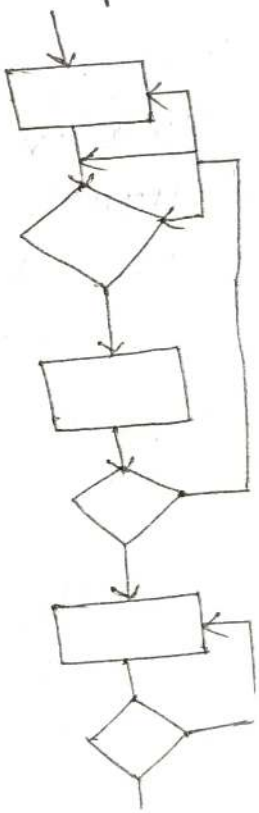
i) Testing begins from the inner most loop first. At the same time set all the other loops to minimum values.

Concatenated loop



The concatenated loop can be tested in the same manner as simple loop test.

Unstructured loop.



The testing can not be effective conducted for H unsaturated loc

n=m that means testing is done when there are n passes.	(ii) The simple loop test for inner most loop is done (iii) Conduct the loop testing for the next loop by keeping the outer loop at minimum value, and other nested loops at some specified value. (iv) This testing process is continue until all the loop have been tested.	These type of loo need to be redisigned:

Static Program / Testing Analysis :- Static Testing Analysis.

~~Seeing~~ To detecting error without direct execution of the test objective. The goal is to localised as early as possible error part of test objects.

The s/w developer normally, make use of static testing while developing their code.

The static testing can be done by various methods:-

(i) Formal technical review.
(ii) Walk through.
(iii) Code inspection.

(i) Formal technical reviews :-

It involves the analysis of ~~an~~ facts by a group technically skill people, following a specify a documented process. This is a quality ensurance activity perform by s/w engineering. ~~The objective~~

Objectives of formal technical reviews :- (FTR)

(i) Two uncover error in function, logic or implementation for any representation or o/w FTR is an effective way.

(ii) FTR is done to verify that s/w being reviewed meets its requirement.

(iii) It ensure that s/w is represented according to relevant standard.

(iv) The FTR help to acheive uniform s/w developmen and make project more manageable

FTR includes walk through code inspection and other s/w group and s/w assignment.

The FTR is conducted in following stages :-

(1) Preparation :- In it there is group of 4-5 people. one is team ~~leader~~ and other are review the product, and establish the report, for the reviewed meeting, and structure a meeting time.

(2) Meeting :- Meeting is affected by producer, review ~~leader~~ and all reviewers. Reviewers notes all important issues raised during the revie

(3) Review Reporting & Recording :-

After the review meeting the recorder produce a review summary report.

Answering the question —

(a) What was reviews?
(b) Who reviewed it?
(c) What were the finding & conclusion?

(4) Walk through :-

In a walk through author describe & explain the work product in the informal meeting to supervisor to get feedback. It to analysis of product for it effectiveness. In design phase of the product the purpose of walk through is to find out as many as possible problem. In product design while the design is on paper It is cheaper to make changes. Generally, walk through can be done at any stages of product development as given below.

(i) At the time of deciding schedule for different phases.
(ii) At the time of problem specification.
(iii) Design data structure.
(iv) Program Designing.
(v) Preparing documentation and user manuals.
(vi) Test plan, data and result
(vii) Maintainence changes

So, walk through can start in early state of soft. development, as design, planning. It is an static method quality assurance, walk through are informal meetings but with purpose is to ensure the high quality.

Objectives of walk through:-

Its main objective are to find:-

(a) Bugs (b) Misinterpretation (c) Error

(d) Inconsistances and any thing i.e., unclear.

Any thing i.e., complex and difficult to modify the purpose of walk through is to only to find out the problem not to correct them. The correction is the field of developer.

(ii) Code Inspection:-

It is a kind of review which is more in deta then the walk through.

For code inspection a team of following members is required.

(a) Moderator:- Manager, leather of inspection team

(b) Designer:- Team responsibility of current phase.

(c) Implementor:- Team responsibility for next phase

(D) Testor:- Preferable from s/w quality assuran team.

The code inspection is carried out into 5 stages

(i) Overview :- In this phase, all important document such as SRS, design document, code, plan are reviewed.

(ii) Preparation :- Understand the document in detail then list all the fault type found in inspection

(iii) Inspection :- Walk through the document and ensure that each item is covered every branch is taken at least one. Find fault and document them.

(iv) Debug :- Resolve all faults and problems.

(v) Follow up :- Moderator must ensure that every issue has been resolved if there large no. of fault then the code inspection team suggest the redesigning.

Difference b/w Static Testing & Dynamic Testing :-

Static Testing	Dynamic Testing
1) It means conducting all the verification activities. During static testing it ~~is~~ check whether all the set of standards of organization (coding, integration) are followed or not. Ex- Technical review, code inspection, walk through are static testing method.	1) Its means conducting validation activities. Unit testing, integration testing acceptance testing, system testing are dynamic testing methods.

2.) It is cost effective.

3.) It can be carried out within short time.

2.) Actual testing will involve more cost. Some time changes may need to be introduced in order to remove the bugs. Hence, it is not cost effective process.

3.) It is run for longer time.

Halstead's S/W Science :-

Halstead's Complexity measurement was developed to measure a program module's complexity directly from source code.

- n_1 is number of distinct operator.
- n_2 is " " " operands.
- N_1 is total numbers of operators
- N_2 " " " " operands.

Measure	Symbol	Formula
Program length	N	$N = N_1 + N_2$
Program Vocabulary	n	$n = n_1 + n_2$
Volume	V	$V = N * (\log_2 n)$
Difficulty	D	$D = \left(\frac{n_1}{2}\right) \times \left(\frac{N_2}{2}\right)$
Effort	E	$E = D * V$
Estimated length		$\Rightarrow n_1 \log n_1 + n_2 \log n_2$

```
void swap (int a[], int i)
  {
int temp;
 temp = a[i];
 a[i] = a[i+1];
 a[i+1] = temp;
```

operands	occurance	operators	occurences
swap	1	( )	1
a	5	{ }	1
i	5	void	1
temp	3	int	3
1	2	[ ]	5
		,	1
		;	4
		=	3
		t	2

$n_1 = 9$	$n_2 = 5$	$N_1 = 21$	$N_2 = 16$

Program length $N = N_1 + N_2$

$= 21 + 16 = 37$

Program vocabulary $n = n_1 + n_2$

$= 9 + 5 = 14$

Volume(V) $= N * (\log_2 n)$

$= 37 * \log_2 14 = 42.40$

Difference b/w Black box testing & white box testing

Black-Box Testing	White-Box Testing
1) It is also called functional or behaviour testing.	1) It is also called Glass box testing.
2) It examines some fundamental aspects of the system. With little regards for internal logical structure of the s/w.	2) In white box testing, the procedural detail, all the logic path, all the internal data structure all closely examine.
3) During Black box testing the program cannot be tested 100%.	3) White-box testing lead to test the program throughoutly.
4) This type of testing is suitable for large projects.	4) This type of testing is suitable for small project

Distinguish among error | fault | failure | defect | debuggi

(1) Error – It is a state that can lead to a system behaviour that is unexpected by the system user.

(2) fault – It is a characteristic of s/w system that can lead to system error.

(3) failure – It is an event that occur at some point in time when the system does not deliver a service as per user expectation.

Defect :-

Any error that remain uncovered and are found in later task, are causes of defect. Error removes is s/w developer activity and defect remove is a s/w quality assurance activity

(5) Debugging :- Debugging is a process of removes of defect. It occur as a consequence of successful testing.

TESTING Vs DEBUGGING :-

Testing	Debugging
(1) It is a process in which the bugs is identified	(1) It is a process in which the bugs or error is corrected by the programmer.
(2) In testing, process it is identify where the bug occur.	(2) debugging is a deductive process
(3) Testing start with the execution result from the test cases.	(3) Debugging start after the testing process.

⇒ Find the boundary value test cases for the following :-

(i) if x is less than level 1 go to 100 else 200.

(ii) if y is greater than level 2 go to 300 else 400.

S.No.	Test case name	Test data	Expected Results
1.)	Testing lower boundary of x.	if x = level 0 if x = level 1	Go to 100 Go to 200
2.)	Testing upper boundary of x.	if x = level 2 or more.	Go to 200.
3.)	Test lower boundary of y	if y = level 1 or level 0 if y = level 2	Go to 400 Go to 400.
4.)	Test upper boundary of y.	if y = level 3 or more.	Go to 300.