UNIT-4 Software - Testing

Software Testing:

* It is a process of evaluation to check the functionality of the s/w application and to identify the defects.

→ It can also be defined as the process of checking that the developed s/w meet the specified requirement or not. i.e. if the developed s/w covers all the requirements mentioned in SRS or not.

As per IEEE —

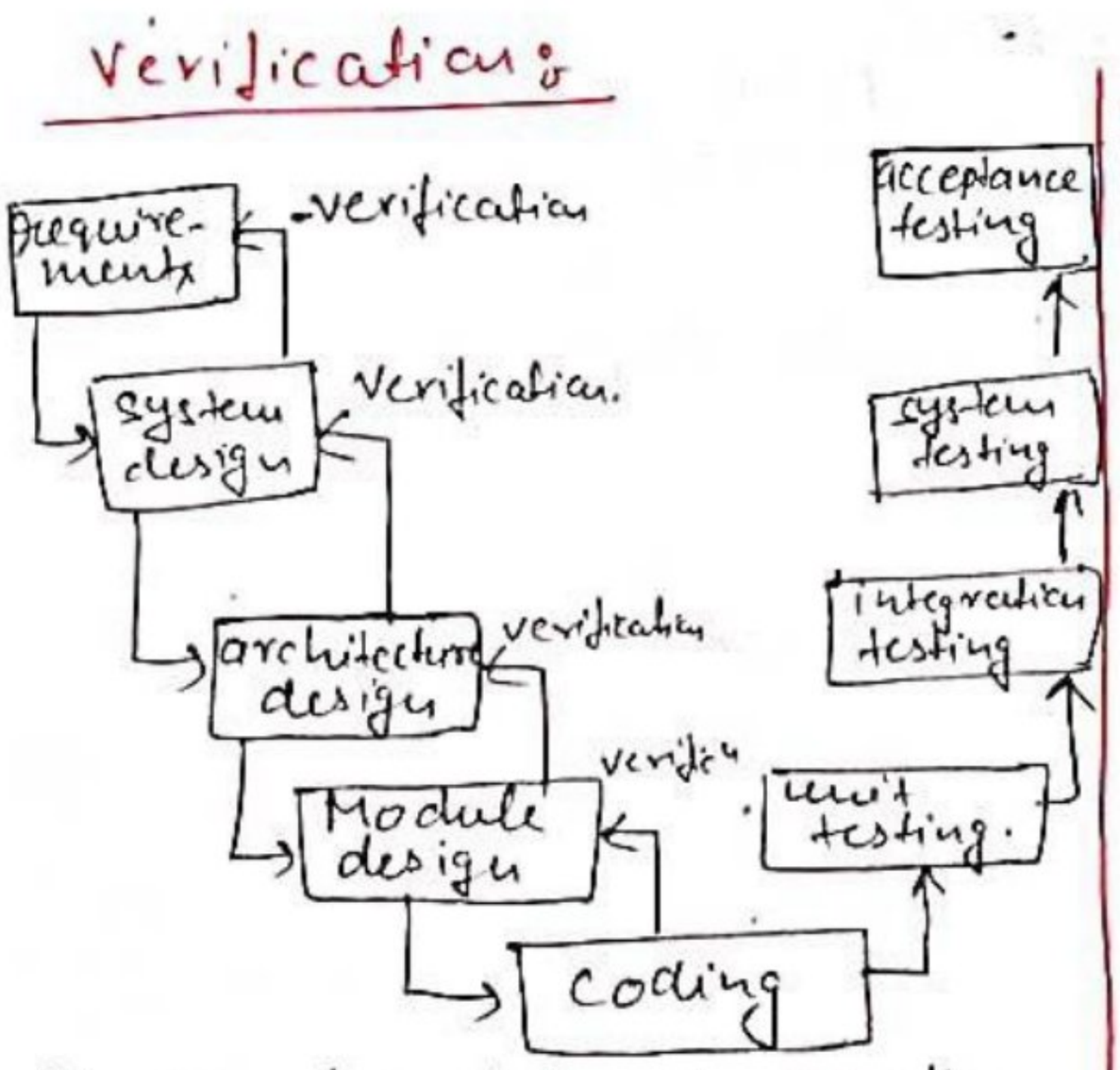
s/w testing can be defined as "A process of analysing a s/w item to detect the differences b/w existing and required conditions (i.e. defects) and to evaluate the features of s/w items".

s/w testing is a two step process which can be divided as — By Barry Boehm —

⇒ Verification → "Are we building the ~~right~~ product right"

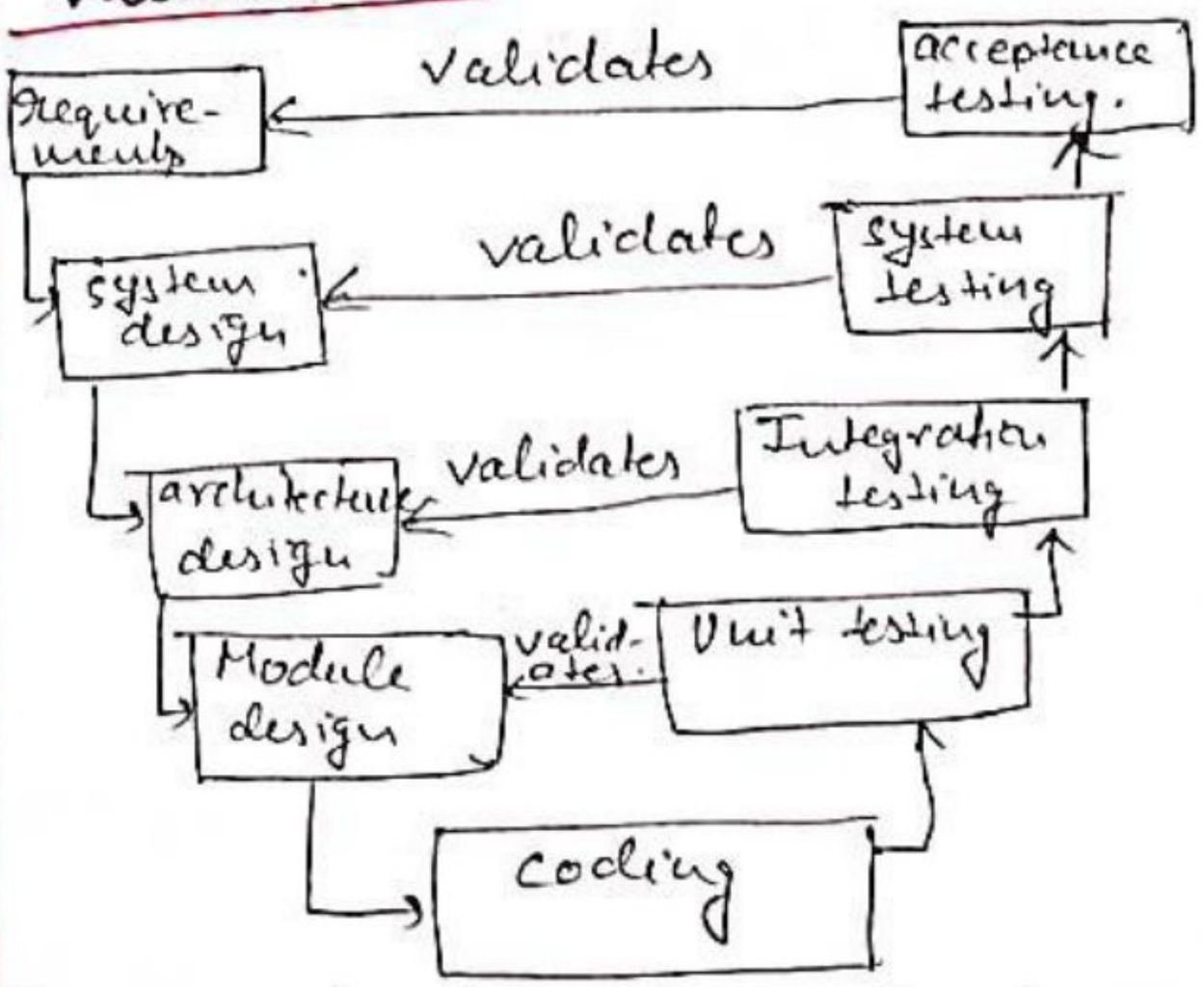
⇒ Validation → Are we building the right product."

Verification:



It can be defined as the process of determining whether or not the products of a given phase of the s/w development cycle fulfill the

Validation



It can be defined as the process of ~~determining~~ evaluating s/w at the end of the s/w development process to ensure compliance with s/w requirements.

requirements established during the previous phase.

Methods used in verification are reviews, walkthroughs, inspection & desk-checking.

It consists of checking of document/files and is performed by humans.

Quality assurance team does verification.

It checks whether the s/w conforms to specification or not.

It does not include execution of the code.

Methods used in validation are black-box testing, white box testing, non functional testing etc.

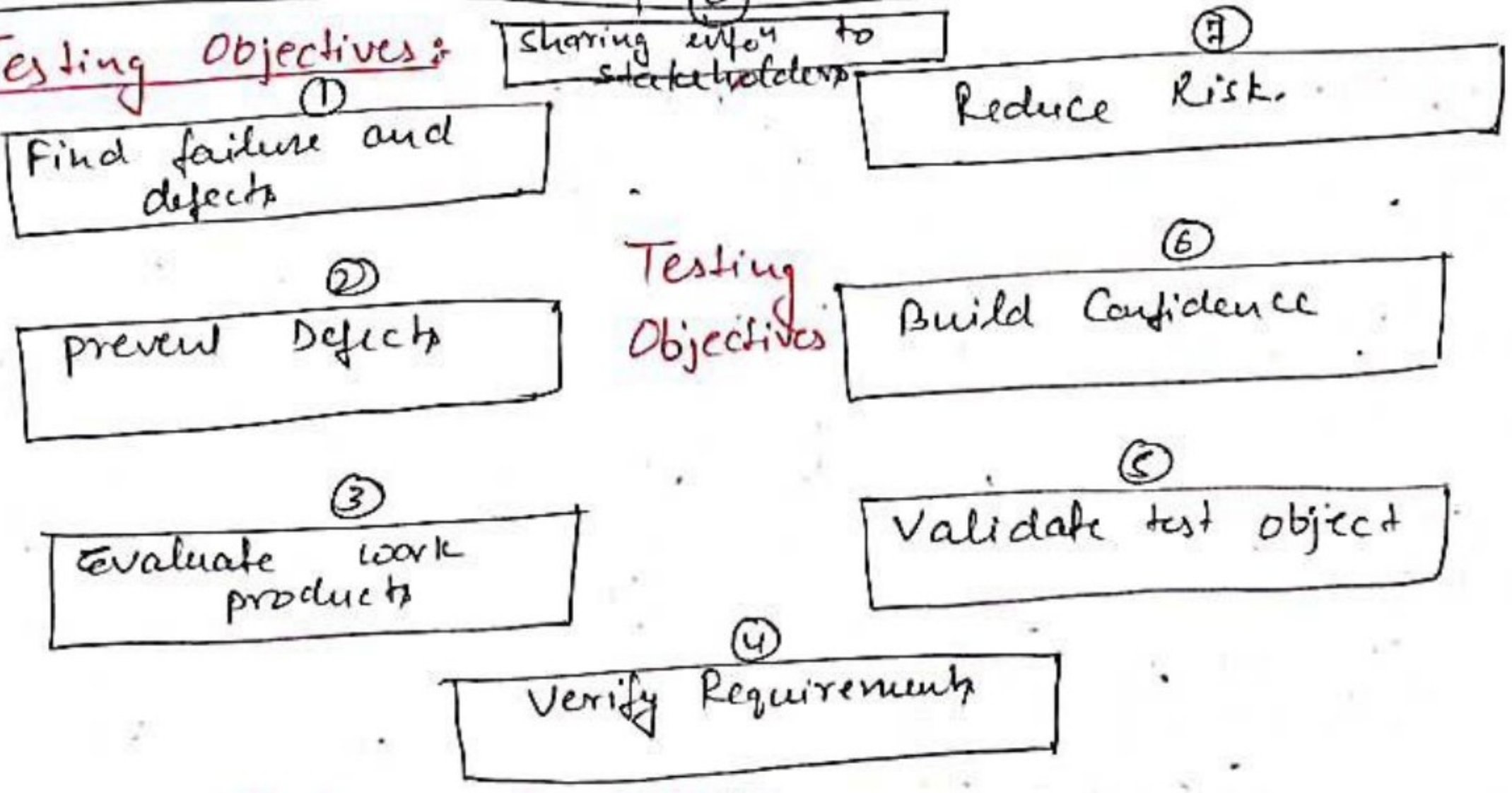
It consists of execution of program and is performed by computer.

Validation is executed on s/w code with the help of testing team.

It checks whether the s/w meets the requirements and expectations of a customer or not.

It includes the execution of the code.

Testing Objectives:



1) Find failure and defects:

Defects should be identified as early in the test cycle as possible.

2- **Prevent Defects:**

If the test procedures are efficient it helps in providing an error-free application.

3) **Evaluate work products:**

The work products refers 'automated tests, test cases, test strategy document, use case diagrams, class models, SRS etc.

So these work products are evaluated or checked for the betterment of the product.

4. **Verify Requirements:**

To check if all the requirements mentioned in the SRS have been fulfilled or not.

5. **Validate test objects**

For validation it is required to check if there is no illegal or incomplete requirements mentioned in the SRS.

After this to ensure everything is complete and and works as per the users requirement and other stakeholders.

6- **Build Confidence:**

To build confidence in the level of quality of the test object.

7 **Reduce Risk:**

Reducing the level of risk of inadequet s/w quality eg. previously undetected failures occuring in operation.

8) **Sharing info to stakeholders**

It is required to provide enough info to stakeholders to allow them to make informed decision, especially regarding the level of quality of the test object.

Levels of s/w testing:

1- UNIT Testing
2- Integration Testing
3- System Testing
4- Acceptance Testing.

1) Unit Testing:

It is a type of s/w testing in which the smallest testable parts of an application called units are individually tested.

Process of Unit Testing:

a- Create the test cases
b- Review or rework on test cases
c- Baseline. (The known measures based on which testing should be done.).
d- Execute test cases.

Advantages of Unit Testing:

1- Makes the process Agile
2. Quality of code.
3- Finds s/w bugs early.
4- Facilitates changes and simplifies iteration.
5- Provides Documentation.
6- simplifies the debugging process.
7- Reduce Costs.

Disadvantages:

1- It takes time to write test cases.
2- Tests requires lot of time PB for maintenance.
3- Testing GUI is not easy using Unit testing.
4- Unit testing can't catch all errors.

Q.

Integration Testing:

Integration testing is done to test the modules/components when integrated to verify that they work as expected. i.e. to test the modules which are working fine individually also works well after integration.

Advantages:

1- It makes sure that integrated modules work properly as intended.
2- It can detect the errors related to interfaces b/w the modules
3- It can identify the errors at early stages, so that it can be cured early.
4- Covers large volume of the system, so more efficient.
5- More reliable.

System Testing:

System testing means testing the system as a whole.

All the modules and components are integrated in order to verify if the system work as expected or not.

Advantages:

1- verifies the system against the business, functional and technical requirements of the end users.
2- Helps in getting maximum bugs.
3- It increases the confidence level of the team.
4- This testing phase uses the test environment similar to real business environment.
5- It is a black box testing hence testers do not need programming knowledge to perform it.

Acceptance Testing :

* An acceptance testing is the form of S/W testing, performed by customer or end-user.
* It is the final stage of S/W evaluation, conducted once the team is confident that any bugs or errors have been resolved.
* It acts as "go" or "no-go" from customers and stops the release of any below standard S/W from being launched.

Advantages :

* provides an end users vision.
* keeps ongoing maintenance cost as low as possible.

Regression Testing:

* It is a type of s/w testing that ensures an application still functions as expected after any code changes, updates or improvements. is done
* It is responsible for overall stability and functionality of the existing features.

When to apply Regression Testing:

Typically regression testing is applied under these circumstances.

⇒ A new requirement is added to existing feature.
⇒ A new feature or functionality is added.
⇒ The sourcecode is optimized to improve performance.
⇒ Patch fixes are added
⇒ Changes in configuration.

Approaches to regression Testing:

1- Retest Everything:

This implies that all of the tests of the system should be re-executed.

2- Regression Test selection:

we can minimize the operating cost by selecting a subset of existing test cases.

3 Prioritization of test cases:

* on this only the most frequent used functionalities ~~and test~~ are being tested first. All other things all tested later.
* By prioritizing test cases, we can cut the size of the testing suite tremendously and have more time to thoroughly assess the performance of the crucial part of the system.

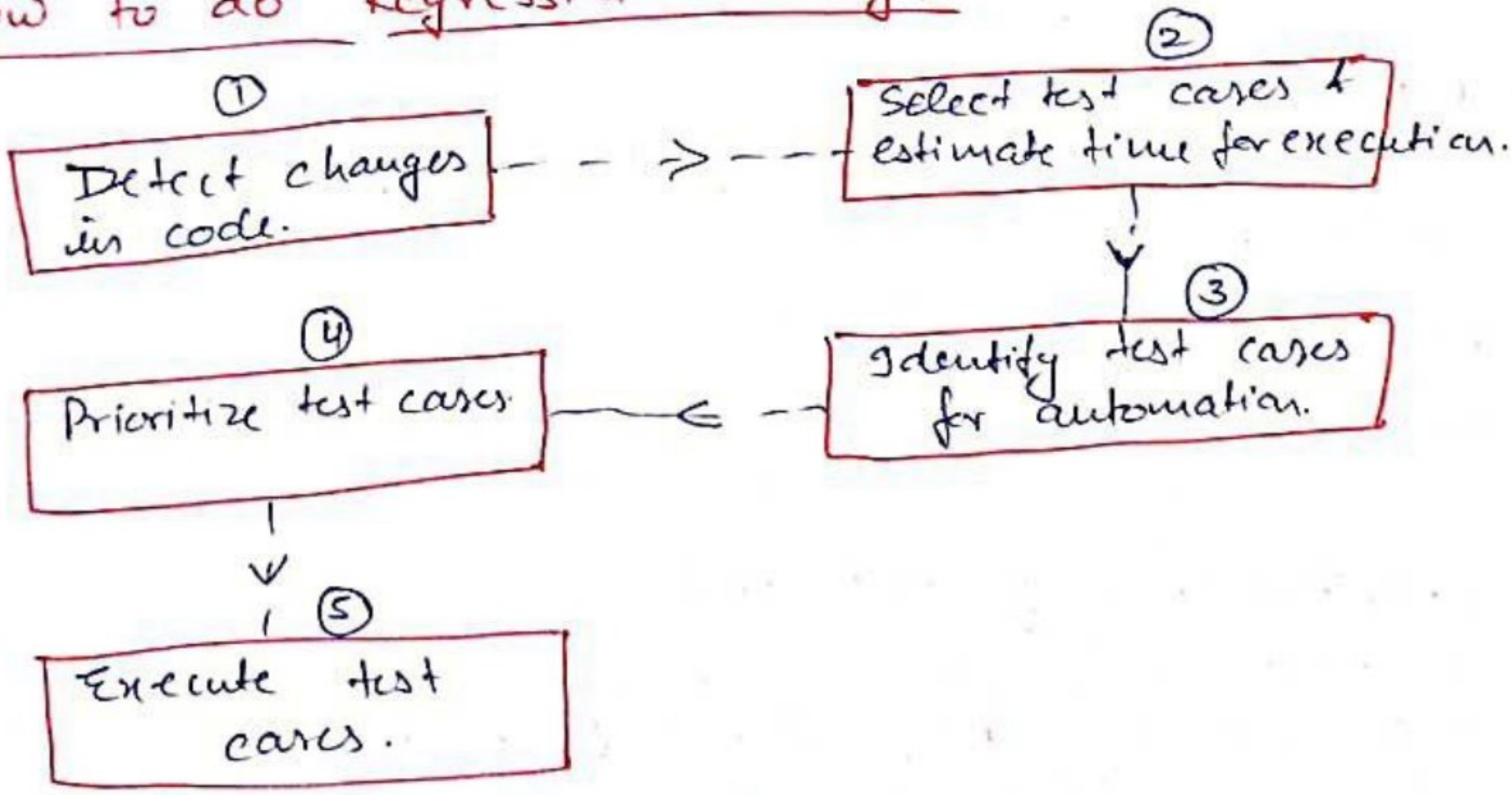
Advantages of Regression testing:

1- Ensures that any changes in the code doesn't adversly affect other functionality.
2- Makes sure that already fixed issues don't re-occur.
3- Serves as a risk mitigation strategy during testing.
4- Easy to learn understand and analyze.

Disadvantages:

1- Without automation regression testing is time consuming.
2- Need to perform for every small change of code.
3- Requires to create complex test cases.

How to do Regression Testing?



How to priorities test cases?

1- Select test cases with frequent defects.
2- Choose test cases with critical functionalities.
3- Select test cases with frequent code changes.
4- Cover end-to-end test flow.

Qu

Testing for functionality (Functional Testing) and Testing for Performance (Performance Testing):-

Functional Testing:

* Functional Testing is performed to confirm that the functionality of an application or system is behaving as expected.
* It is done to verify all the functional requirements of the system.

Functional Testing Types:

(i) Unit Testing
(ii) Component testing
(iii) Smoke "
(iv) Sanity "
(v) Regression "
(vi) Integration "
(vii) API testing
(viii) UI testing
(ix) System testing
(x) white box testing
(xi) Black box testing.
(xii) Acceptance testing
(xiii) Alpha testing
(xiv) Beta testing.

Performance Testing:

It is a non functional S/w testing technique that determines how the stability, speed, scalability and responsiveness of an application holds up under a given workload.

Types of Performance Testing:

1) **Load Testing:**
It measures the system performance as the workload increases. It can be either no. of users or transactions.

2) **Stress testing:**
This testing is used to measure system performance outside of the parameters of normal working conditions.

3) **Spike Testing:**
It is a type of stress testing that evaluates S/w performance when workload are substantially increased quickly & repeatedly. The workload is beyond normal expectations.

Difference b/w Functional Testing and Performance Testing :-

Functional Testing	Performance Testing.
1. To verify the accuracy of the s/w with definite inputs against expected outputs.	To verify the behaviour of the system at various load conditions.
2- It can be manual or automated.	It can be performed effectively if automated.
3- One user performing all the operations.	Several user performing desired operations.
4. Involvement required from customer, developer and tester.	involvement required from customer, tester, developer, DBA and N/w management team.
5- Production size test environment is not mandatory. H/w requirements are minimal.	Requires close to production test environment & several h/w facilities to populate the load.

Top down and Bottom-up Testing Strategies

Top-down testing :-

* In this approach testing is conducted from main module to sub modules.
* In some circumstances if the sub module is not developed a temporary program called "STUB" is used to stimulate the sub module.
* A stub has all the capabilities of the unavailable module.

Bottom-up testing :-

* In this approach testing is conducted from sub module to main module.
* If the main module is not developed, a temporary program called "DRIVER" is used to stimulate the main module.
* A DRIVER has all the capabilities of the unavailable main module.

Difference b/w Top-down and bottom up testing:

Top - down	Bottom- up
1- It is an integration testing used in order to stimulate the behaviour of lower level modules that are not yet integrated.	The lower level modules are tested with higher level modules until all the modules have been tested successfully
2- Advantageous if major flaws occurs at the top of the program.	Advantageous if major flaws occur at the bottom of the program.
3- Less complex.	More complex and highly data intensive.
4- performed from main module to sub module.	performed from sub module to the main module.
5 The observation of test o/p is more complex.	The observation of the test o/p is accessible.

Difference b/w STUB and Drivers

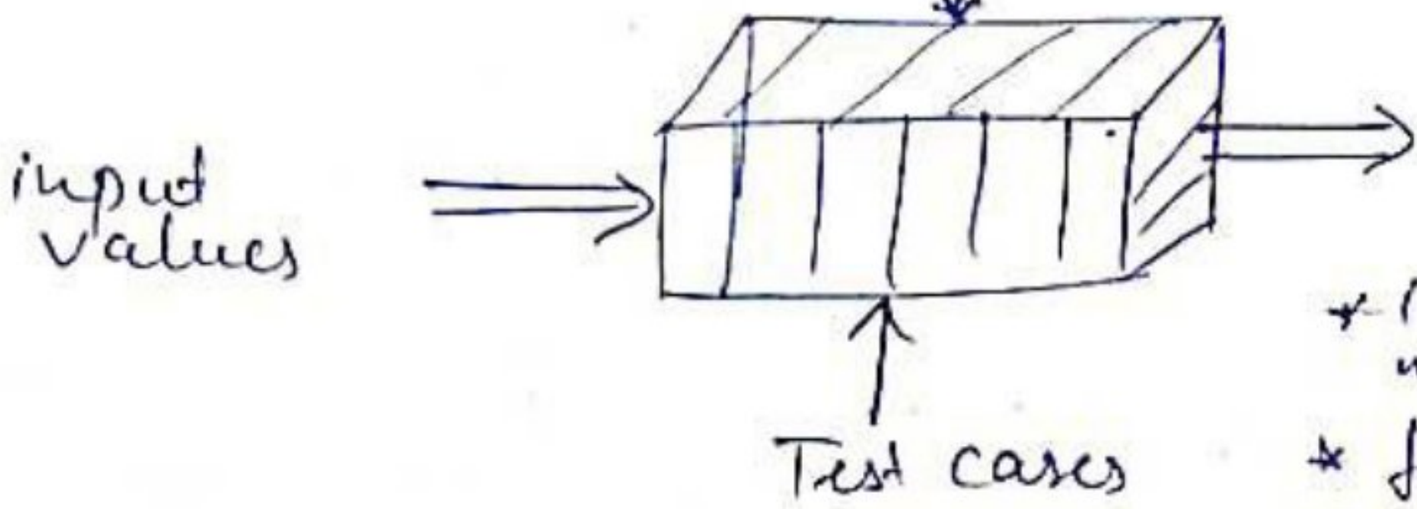
1- Stubs are used in top-down integration.	Drivers are used in bottom-up integration testing.
2- Stubs are generally known as "called programs"	Drivers are the "calling programs"
3- stubs are basically used in a unavailability of low-level - modules.	Drivers are used in place of high level modules
4- Stubs are taken into use to test the feature and functionality of the modules	Drivers are used if the main module of the S/W is not developed for testing

Black box testing and white box testing.

4.12

Black box testing:

* It is a s/w testing used to examine the functionality of the s/w instead of bothering about the internal structure of the s/w.
* It is also Termed as Functional / behavioural testing as it examines the functionality / behaviour of the s/w.



* Internal structure is not known.
* functionality is known.
* Test cases are defined
* requirements are known.

Types of errors identified during black box testing:-

1- It will discover the functions that are missing. In the s/w and also the functions that are implemented incorrectly.

2- It also uncovers the errors that came across while using the interfaces.

3- It uncovers the errors faced in accessing the database.

4- It discovers the error that occurs while initiating any function or while terminating any function.

5- It also uncovers the errors in the performance or behaviour of the s/w.

How to perform Black-box testing by testers

1- Analyze the requirements and specification of S/w.
2- Select the set of valid input values and invalid input values (detected as an error)
3- Declaration of desired output.
4- Design test cases, execute them with set of valid and invalid inputs and collect the generated output.
5- Now compare the generated output with the desired output.
6. In case any error is detected, it is fixed and testing is performed again.

Advantages of Black-box testing:

1- Efficient when used on large systems.
2- Tester and developer are independent so testing is balanced.
3- There is no need for the tester to have detailed functional knowledge of the system.
4- Testing is done from users point of view, so it becomes easy to identify missing functions and function errors.

Disadvantages:

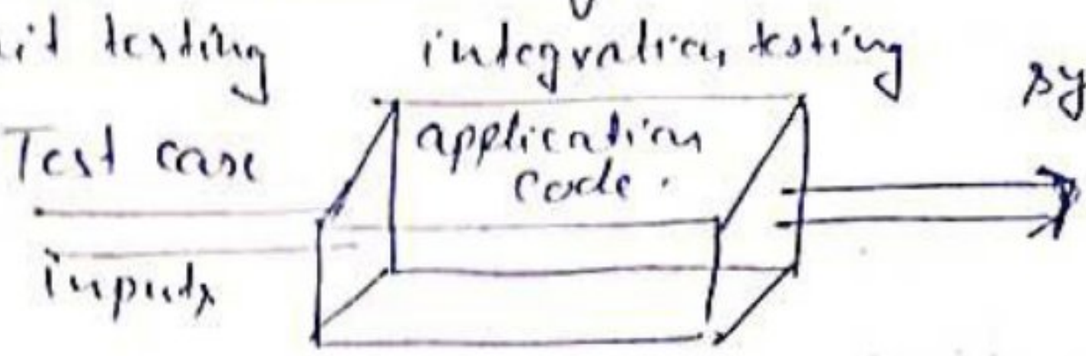
1- It is difficult to design test cases without having clear functional specifications.
2- It is difficult to identify all possible inputs in limited testing time.
3- There is chances of having undefined paths during the testing process.
4- There is high probability of repeating tests, already performed by the programmer.

White box testing:

It is a type of s/w testing which analyzes the internal structures, internal design, code structure and the working of the s/w also.

It do not just focuses on the functionality as in black box testing.

It is also called glass box testing or structural testing.



What should be verified in white box testing:

1- Testing of any broken or incomplete path.
2- To verify the flow of structure as mentioned in requirement document.
3- To check the conditionality of all loops and the overall functionality of the s/w.
4- To check if all the expected outcomes are met.
5- Line by line verific^{n} of code.

How to perform white box testing:

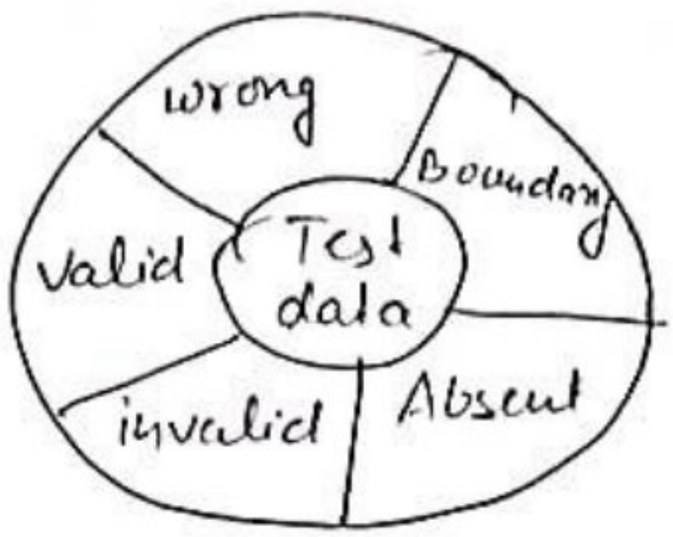
1) Understand the code.
2) Creating and executing test cases:-

The test cases will be written by ~~developer~~ tester by dividing the applic^{n} into categories as- statement / Decision / condition / branch.

- Statement coverage will include those statements that are executed at least once during the execution of the program.
- Branch coverage will include the outcome for every code module (statement or loop).
- A decision coverage will include reports for each boolean expression present in the source code.
- Conditional coverage is used to test the variables used in different types of conditional statements (like IF/ELSE, SWITCH etc.

Test data Suit Preparation:

* Test data in S/W Testing is the input given to a S/W program during test execution. It represents data that affects or is affected by S/W execution while testing.



* Test data is used to varify the functionalities of any S/W as well as it is used to test S/W ability to handle unusuals, unexpected or exceptional inputs/behaviours.

* S/W System should be test against the following.
  - ⇒ Valid test data
  - ⇒ Invalid test data
  - ⇒ Boundary test data
  - ⇒ wrong data
  - ⇒ Absent data

Test Suite:

* It is a container that has a set of tests which helps testers in executing and reporting the test execution status.

* A test case can be added to multiple test suites and test plans.

* After creating a test plan, test suites are created which in turn can have any number of tests.

Test suite can also be defined as "validation suite" which is a collection of test cases that are used to test S/W programs.

Alpha Testing and Beta Testing:

* Alpha testing is the final internal acceptance testing for the s/w. to ensure that the s/w is ready to release.
* It is a white box testing in which the QA team know exactly how the s/w will behave.
* The aim of this is to test every single flow end to end. and ensure that the s/w is bug free.

Beta Testing:

* Beta testing is the final stage of acceptance testing where the s/w is released to a limited no. of real users.
* This testing is unstructured and users are encouraged to give feedback about how the application performs.
* This testing is more focused on performance and stability.

Difference b/w Alpha and Beta testing:

	Alpha	Beta.
Main Purpose	To test every single flow and ensure that the applicn is working as expected.	To Understand how real users interact with the applicn and test how it works in real-world condition.
Done by	Internal team.	General public (selective users) either via invite or by control release.
Visibility	White box, the testers know what is happening & what they are testing.	Black box, the testers just see the applicn and test them by using functionalities.
structure.	Rigorously structured. every flow is tested and all results are recorded and carefully analyzed.	Completely unstructured. Users are free to do what they want. Feedback is requested, but not required.
performance.	Not interested in how the applicn or backend performs.	Reliability and performance are key aspects of beta testing.

Static Testing VS Dynamic Testing:

Static testing is a s/w testing method that involves the examination of a program, along with any associated document but does not require the program to be executed.

Dynamic Testing on the other hand involves interaction with the program while it runs.

What exactly is tested in static testing:

Requirement Specifications

Design Documents.

User documents.

Test cases, test data

source code. (code inspection)

Benifits of static testing:

Easy detection and correction of any coding errors.

Reduces cost

Reduces timescale for development.

Quality issue identification.

Static Testing Strategies:

1- Formal Technical Reviews /Peer review (Page -2.21).

2- walkthroughs

3- Code inspection.

1) FTR(Formal Technical reviews):

Technical specifications are reviewed by peers (groups) in order to detect any errors.

2) walkthrough:

The authour of the whichever document is being reviewed will explain the document to their team.

participants will ask questions and any notes are written down. 4.18

3) Inspection

A designated moderator will conduct a strict review as a process to find defects.

Design and Coding Standards:

1- Limited used of global.
2- Standard headers for different modules.
3- Naming Conventions should be followed for local, global and constants & functions.
4- Indentations should be properly done.
5- Error handling and exception handling should be follower properly.
6- Avoid difficult coding style.
7- Code should be well documented.
8- Design should be clear.
9- Flow should be mentioned properly.
10 Cardinalities should be mentioned properly.
11 Low level and high level design specifications
. should be followed.
92-