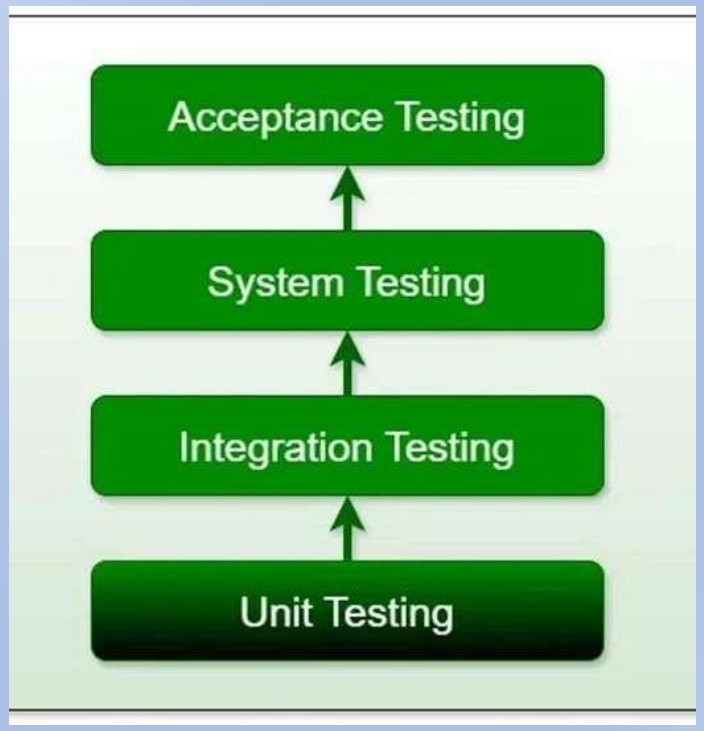
UNIT 2: Types and Levels of Testing

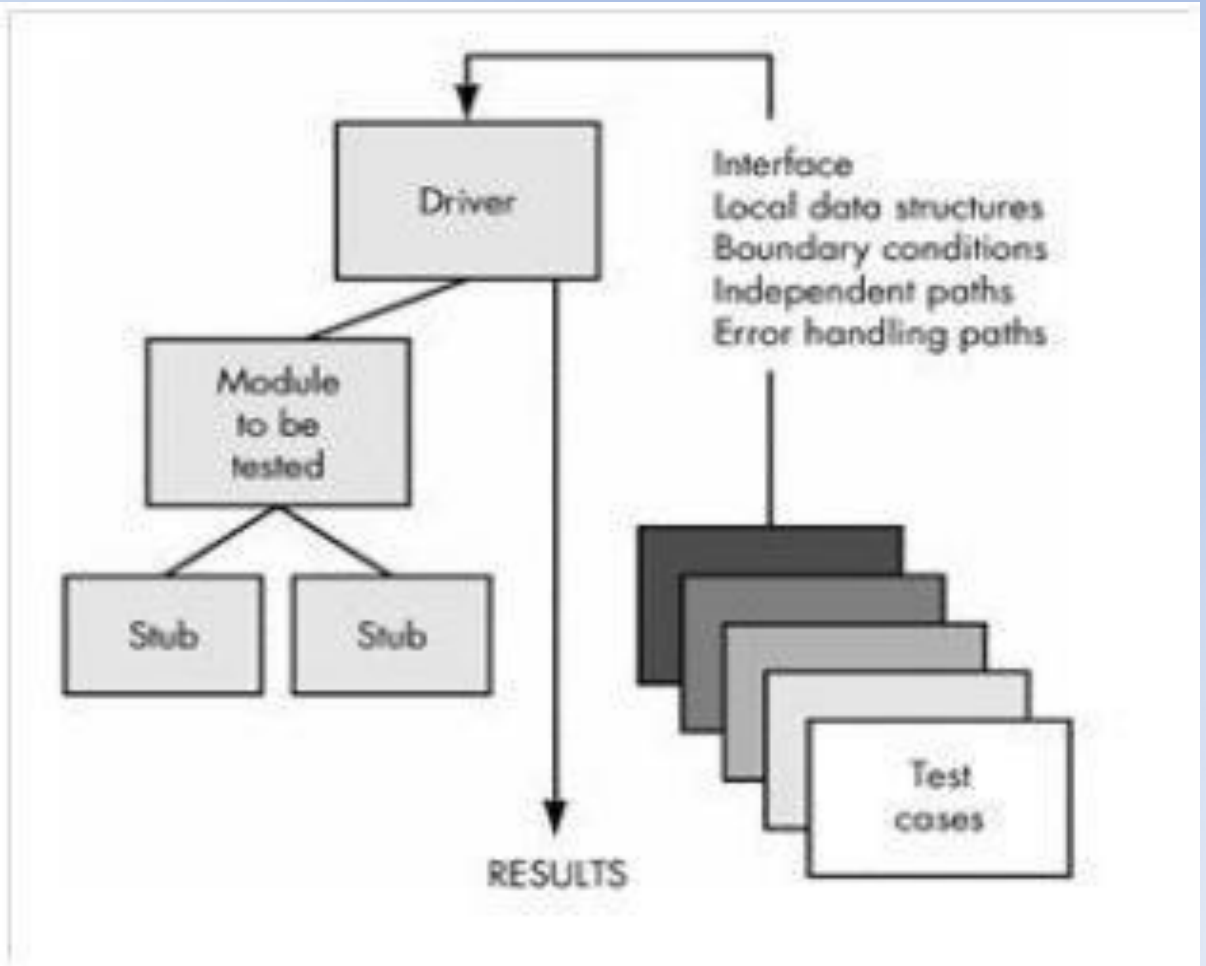
- Levels of Testing
- Software testing levels are different stages of SDLC where testing is conducted.
- There are 4 levels of testing:
  - Unit testing
  - Integration testing
  - System testing
  - Acceptance testing



Unit Testing

- Individual units or components of software are tested.
- Purpose is to validate each unit of software performs as designed.
- Unit: smallest testable part of any software .
- Unit has one or more inputs and a single output.
- Unit can be – individual program, function, procedure etc.
- In object oriented programming -smallest unit is a method , belonging to base or super class.

- Unit Testing Method:
  - It is performed by using White Box Testing Method.
- When is it performed?
  - Performed at first level of software testing and prior to Integration Testing.
- Who performs it?
  - software developers.
  - sometimes software testers.



- Drivers :
  - Drivers are used in bottom-up integration testing approach.
  - It can simulate the behaviour of upper-level module that is not integrated yet.
  - Drivers modules act as the temporary replacement of module and act as the actual products.
  - Drivers are also used for interact with external system and usually complex than stubs.
  - Driver: Calls the Module to be tested.

- Assume you have 3 modules, Module A, Module B and module C.
- Now suppose you have modules B and C ready.
- but module A which calls functions from module B and C is not ready.
- So developer will write a dummy piece of code for module A which will return values to module B and C.
- This dummy piece of code is known as driver.

- Stubs:
  - Stubs are used in top down integration testing.
  - It can simulate the behaviour of lower-level module that are not integrated.
  - They are act as a temporary replacement of module and provide same output as actual product.
  - When needs to intact with external system then also stubs are used.
  - Stub: Is called by the Module under Test.

- Assume you have 3 modules, Module A, Module B and module C.
- Module A is ready and we need to test it, but module A calls functions from Module B and C which are not ready,
- so developer will write a dummy module which simulates B and C and returns values to module A.
- This dummy module code is known as stub.

- Importance of Stubs and Drivers:

1. Stubs and Drivers works as a substitute for the missing or unavailable module.
2. They are specifically developed, for each module, having different functionalities.
3. Generally, developers and unit testers are involved in the development of stubs and drivers.

4. Their most common use may be seen in the integration

incremental testing, where stubs are used in top bottom approach and drivers in a bottom up approach.

	Stub	Driver
Type	Dummy codes	Dummy codes
Description	Routines that don't actually do anything except declare themselves and the parameters they accept. The rest of the code can then take these parameters and use them as inputs	Routines that don't actually do anything except declare themselves and the parameters they accept. The rest of the code can then take these parameters and use them as inputs
Used in	Top Down Integration	Bottom Up Integration
Purpose	To allow testing of the upper levels of the code, when the lower levels of the code are not yet developed.	To allow testing of the lower levels of the code, when the upper levels of the code are not yet developed.

Benefits of Unit Testing:

- Unit testing increases confidence in changing/ maintaining

code. If good unit tests are written and if they are run every time any code is changed, we will be able to promptly catch any defects introduced due to the change.

- Codes are more reusable.
- Development is faster.
- The cost of fixing a defect detected during unit testing is lesser in comparison to that of defects detected at higher levels.
- Debugging is easy.

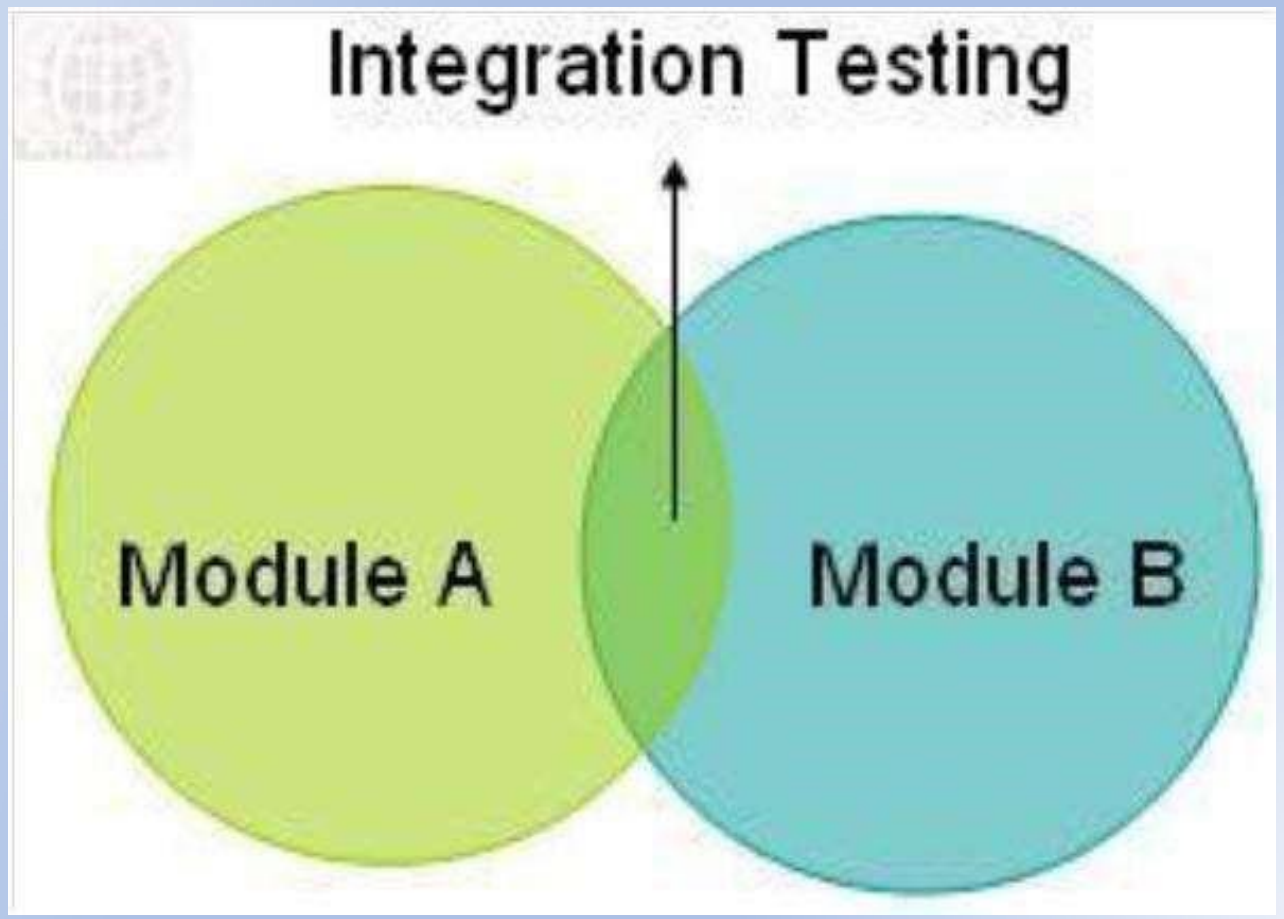
- Errors Detected By Unit Testing
  - local data structure.
  - boundary condition.
  - independent path.
  - error handling path.
  - local and global variable.
  - incorrect initialization.
  - incorrect symbolic representation of an expression.
  - incorrect arithmetic precedence.
  - global variable naming convention.
  - mapping error.
  - log and exception handling.

- Integration Testing:
- Integration Testing is a level of software testing where individual units are combined and tested as a group.
- In integration Testing, individual software modules are integrated logically and tested as a group.

- Integration testing tests integration or interfaces between

components, interactions to different parts of the system such as an operating system, file system and hardware or interfaces between systems.

- As displayed in the image below when two different modules 'Module A' and 'Module B' are integrated then the integration testing is done.



- Reasons for performing Integration Testing:
  - Compatibility between different sub units.
  - Single model is risky.
  - Easy to find errors in sub modules.

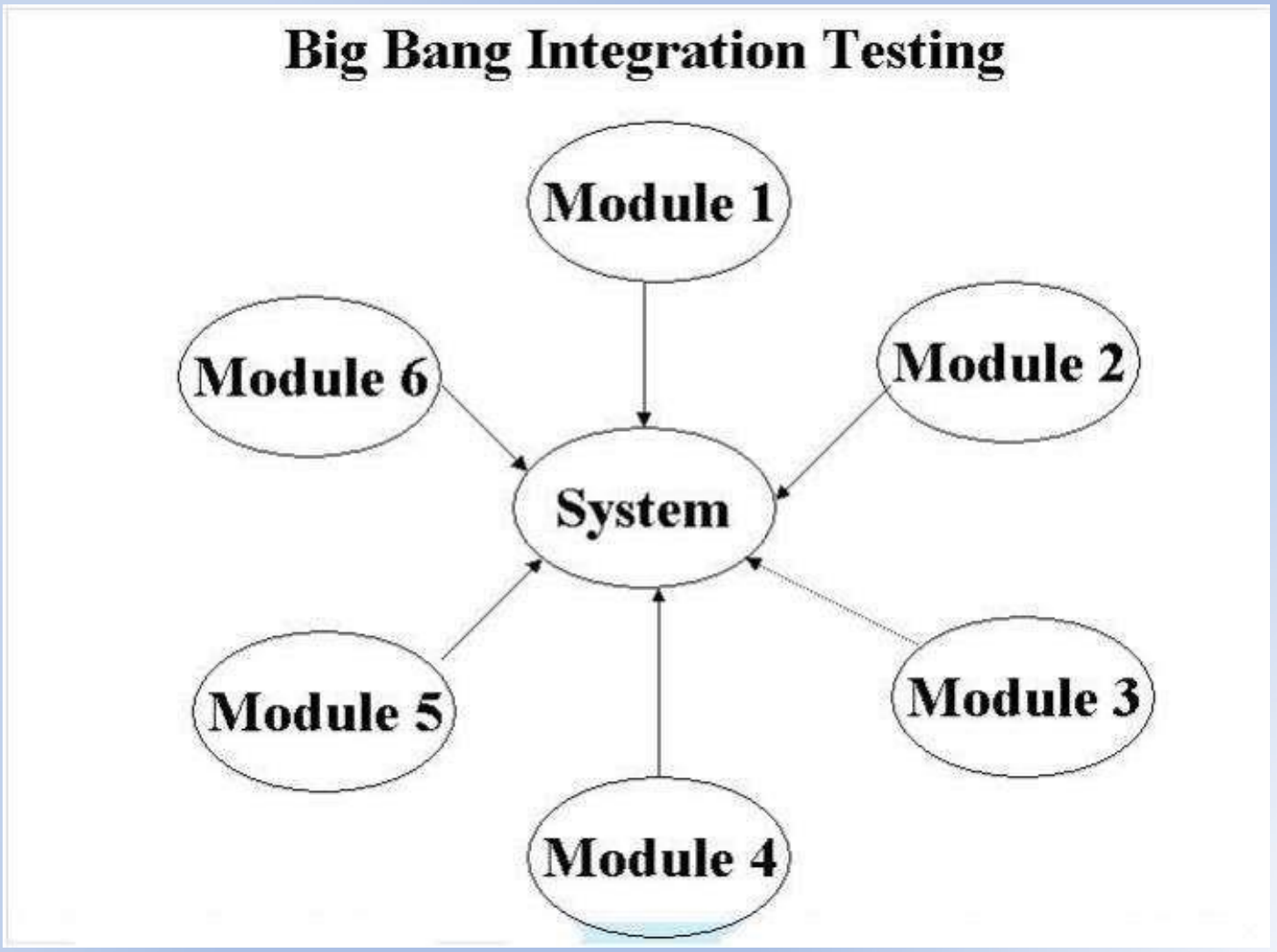
- Broadly there are 2 approaches for integration testing:
  - Incremental testing approach&
  - Non incremental testing approach

1. Non- Incremental Integration :

- The non-incremental approach is also known as —Big-Bang Approach.
- Big Bang Integration Testing is an integration testing strategy wherein all units are linked at once, resulting in a complete system.
- When this type of testing strategy is adopted, it is difficult to isolate any errors found, because attention is not paid to verifying the interfaces across individual units.

- In Big Bang integration testing all components or modules are integrated

simultaneously, after which everything is tested as a whole. As per the below image all the modules from ‘Module 1’ to ‘Module 6’ are integrated simultaneously then the testing is carried out.



- 2. Incremental Approach:

- In this approach, testing is done by joining two or more modules that are logically

related. Then the other related modules are added and tested for the proper functioning. Process continues until all of the modules are joined and tested successfully.

- This process is carried out by using dummy programs called Stubs and Drivers. Stubs and Drivers do not implement the entire programming logic of the software module but just simulate data communication with the calling module.
- Stub: Is called by the Module under Test.
- Driver: Calls the Module to be tested.

- There are 3 approaches:

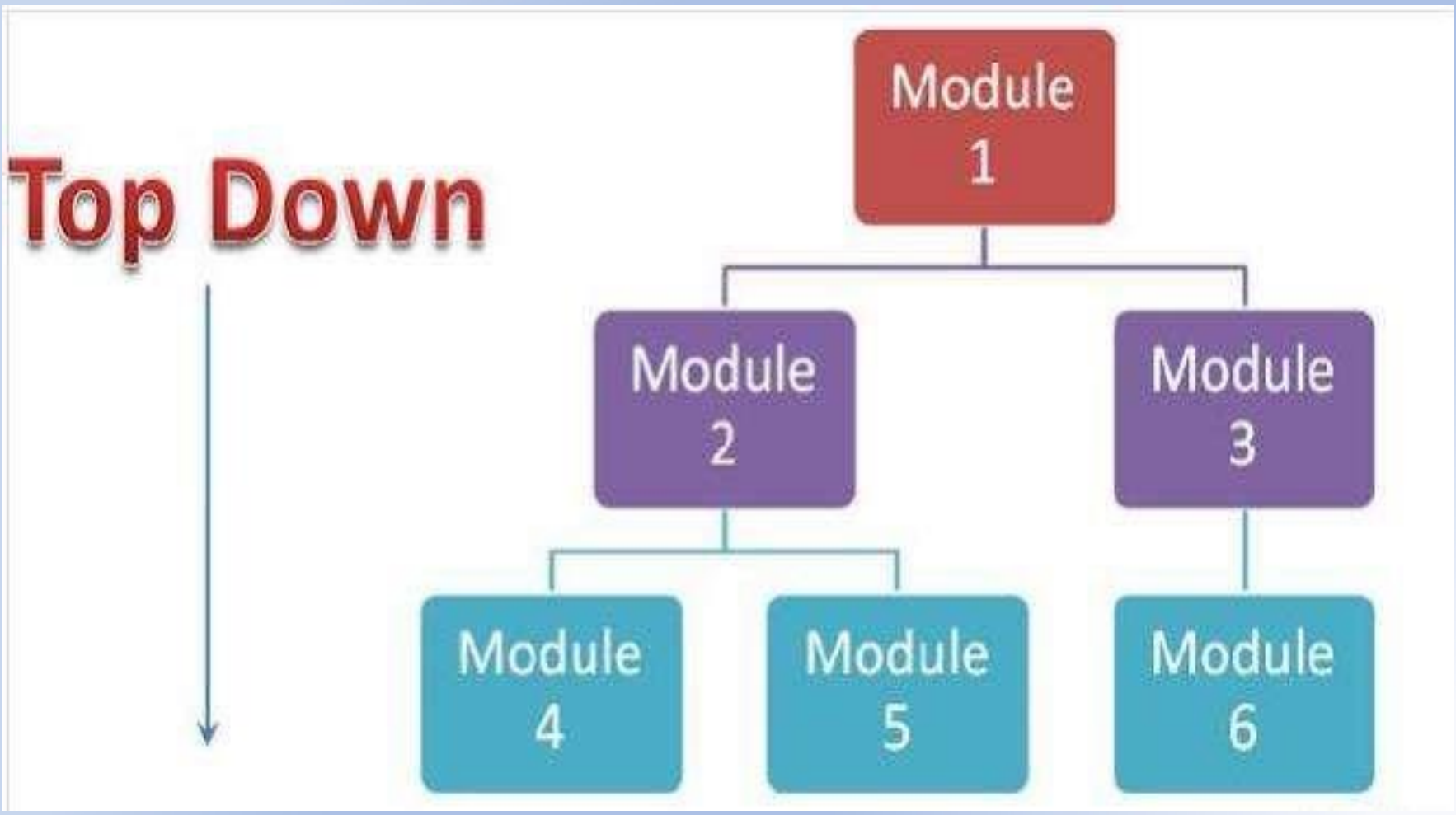
  a) Top down approach,

  b) Bottom up approach, &

  c) Bidirectional or hybrid approach

1. Top down Integration : (6 slides)

- The strategy in top-down integration is look at the design hierarchy from top to bottom.
- Start with the high-level modules and move downward through the design hierarchy. In this approach testing is conducted from main module to sub module.
- If the sub module is not developed a temporary program called STUB is used for simulate the sub module.
- Modules subordinate to the top modules are integrated in the following two ways:



1. Depth first Integration: In this type, all modules on major control path of the design hierarchy are integrated first.

- In this example shown in fig. modules 1, 2, 4/5 will be integrated first. Next, modules 1, 3, 6 will be integrated.

2. Breadth first Integration: In this type, all modules directly subordinate at each

level, moving across the design hierarchy horizontally, are integrated first.

- In the example shown in figure modules 2 and 3 will be integrated first. Next, modules 4,5 and 6 will be integrated .

Procedure:

- The procedure for Top-Down integration process is discussed in the

following steps:

1. Start with the top or initial module in the software. Substitute the stubs for all the subordinate of the top module. Test the top module.
2. After testing the top module, stubs are replaced one at a time with the actual modules for integration.
3. Perform testing on this recent integrated environment.
4. Regression testing may be conducted to ensure that new errors have not appeared.
5. Repeat steps 2-4 for whole design hierarchy.

- Advantages of top-down integration:
  - Advantageous if major flaws occur toward the top of the program.
  - Once the I/O functions are added, representation of test cases is easier.
  - Early skeletal Program allows demonstrations and boosts moral.

- Disadvantages of Top-Down integration :

- Stub modules must be produced.
- Stub Modules are often more complicated than they first appear to be.

- Before the I/O functions are added, representation of test cases in stubs can

be difficult.

- Test conditions may be impossible, or very difficult, to create.
- Observation of test output is more difficult.
- Allows one to think that design and testing can be overlapped.
- Induces one to defer completion of the testing of certain modules.

2. Bottom up Integration: (3 slides)

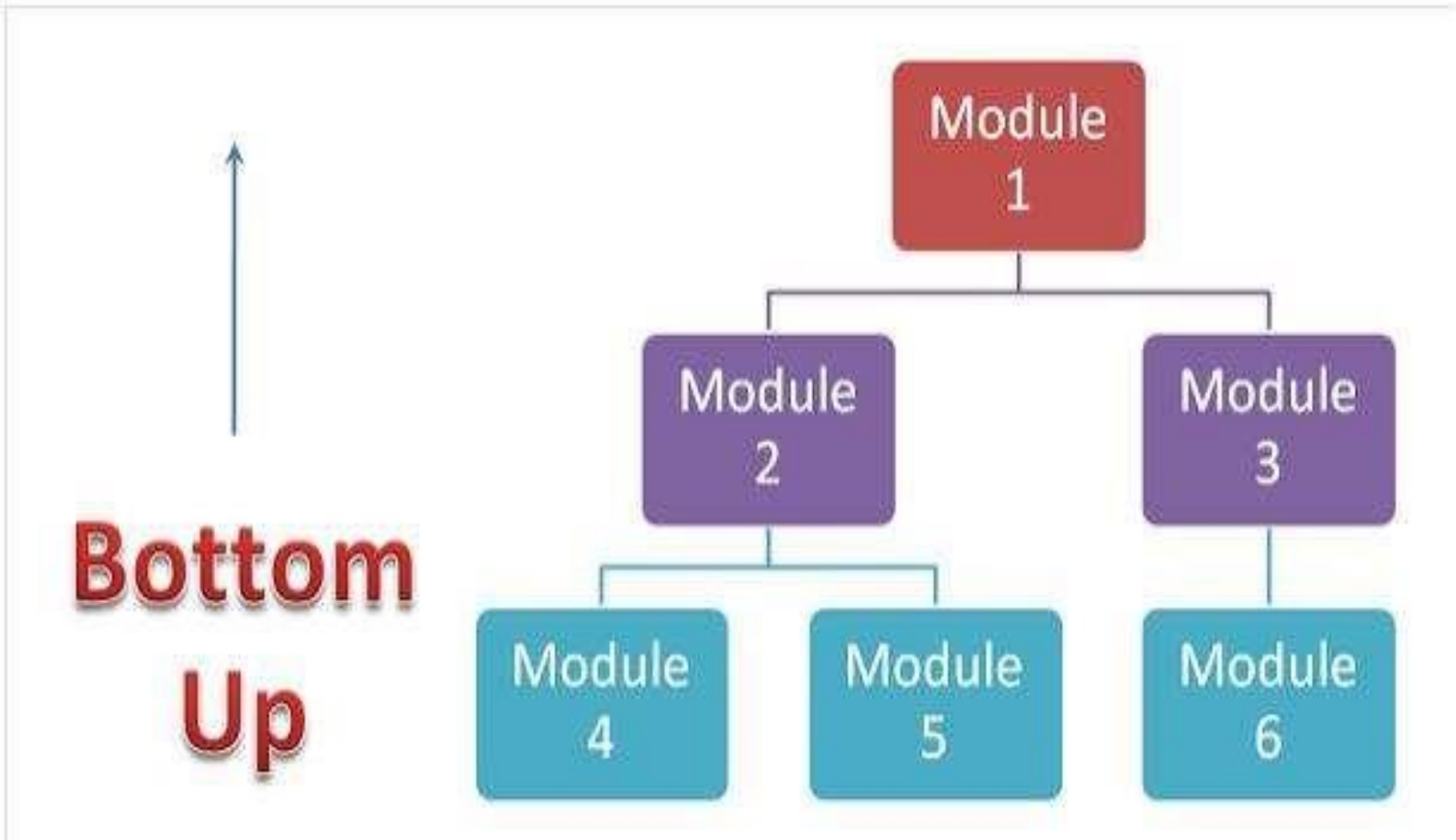
- In this approach testing is conducted from sub module to main module , if the main module is not developed a temporary program called DRIVERS is used to simulate the main module.

Advantages:

- Advantageous if major flaws occur toward the bottom of the program.
- Test conditions are easier to create.
- Observation of test results is easier.
- Driver Modules must be produced.
- The program as an entity does not exist until the last module is added .

Disadvantages:

- Critical modules (at the top level of software architecture) which control the flow of application are tested last and may be prone to defects.
- Early prototype is not possible .



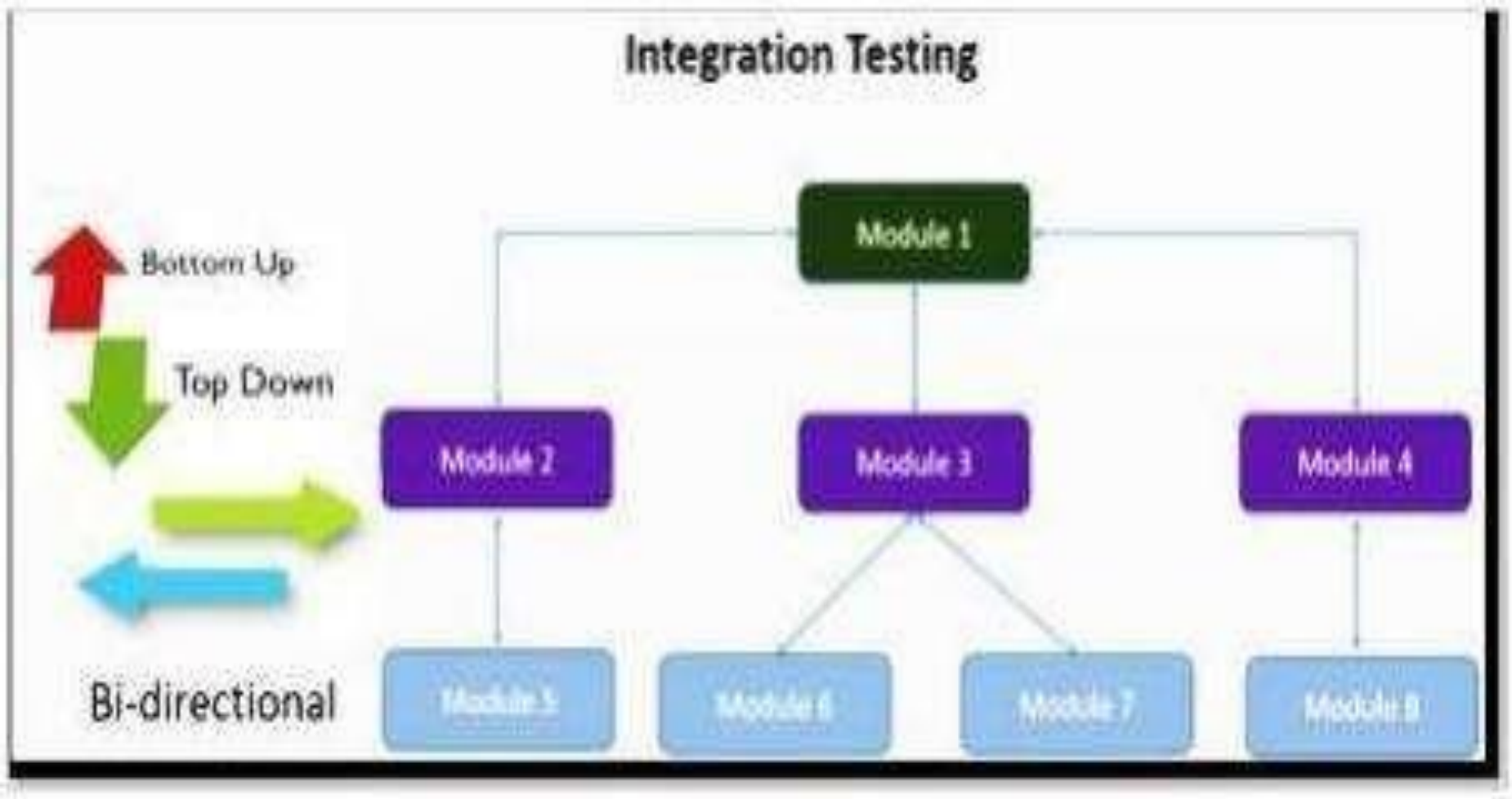
- Bi Directional /Hybrid/ Sandwich Integration Testing : (4 slides)

1. Bi-directional Integration is a kind of integration testing process that combines top-down and bottom-up testing.

$ Top modules are tested with lower modules at the same time lower modules are integrated with top modules and tested

2. With an experience in delivering Bi-directional testing projects custom software development services provide the best quality of the deliverables right from the development of software process.
3. Bi-directional Integration testing is a vertical incremental testing strategy that tests the bottom layers and top layers and tests the integrated system in the computer software development process.
4. Using stubs, it tests the user interface in isolation as well as tests the very lowest level functions using drivers.
5. Bi-directional Integration testing combines bottom-up and top-down testing.

6. Bottom-up (middle to top) testing is a process where lower level modules are integrated and then tested.
7. This process is repeated until the component of the top of the hierarchy is analyzed. It helps custom software development services find bugs easily without any problems.
8. Top-down(middle to down) testing is a process where the top integrated modules are tested and the procedure is continued till the end of the related module.
9. Top down testing helps developers find the missing branch link easily
10. This technique is called as Sandwich Integration.
11. Stubs and drivers are not necessary for the topmost and bottom most level



- Advantages:

1. Sandwich approach is useful for very large projects having several subprojects.
2. Both Top-down and Bottom-up approach starts at a time as per development schedule.
3. Units are tested and brought together to make a system .Integration is done downwards.

- Disadvantages:

1. It require very high cost for testing because one part has Top-down approach while another part has bottom-up approach.

2. It cannot be used for smaller system with huge interdependence between different

modules. It makes sense when the individual subsystem is as good as complete system.

Unit test	Integration test
· The idea behind Unit Testing is to test each part of the program and show that the individual parts are correct.	· The idea behind Integration Testing is to combine modules in the application and test as a group to see that they are working fine
· It is kind of White Box Testing	· It is kind of Black Box Testing
· It can be performed at any time	· It usually carried out after Unit Testing and before System Testing

· Unit Testing tests only the functionality of the units themselves and may not catch integration errors, or other system-wide issues	· Integrating testing may detect errors when modules are integrated to build the overall system

· It starts with the module specification	· It starts with the interface specification
· It pays attention to the behavior of single modules	· It pays attention to integration among modules
· Unit test does not verify whether your code works with external dependencies	· Integration tests verify that your code works with external dependencies

- Testing on Web Applications:

  - Web application is accessed over the network.
  - Web testing is the name given to software testing that focuses on web application.
  - Complete testing of web-based application is done before going live.

- Important points to be considered before Testing Web-based applications:
  - Compatibility with different browsers
    - Tester should test that behaviour of application is consistent on all the browsers.
  - Functional testing
    - Functionality of the application should be proper.
  - Usability
    - Tester should focus on ease of use of the web application.
    - Look and feel should also be good.

- Integration
  - Integration between- browsers and server, hardware and software, application and data should be validated by tester.
- Security
  - Security of web-based application is most important.
  - Authorization or access control should be properly tested.
- Performance
  - Performance should be load tested and stress tested.

- Usability testing: (3 slides)
- This testing is done by testing application with real users who are part of the target audience.
  - Need of Usability testing
    - To understand how real world users interact with your website
    - To ensure app is easy to use and navigate.
    - Usability testing helps in gathering data needed to improve web application.
    - To identify trouble spots for users where they are getting stuck.

- Components of Usability Test
  - There are 2 groups – users and observers .
  - Observers gather more objective data .
  - Users accomplish set of tasks under controlled conditions.
  - Observers can take notes while observing.

- Methods of Usability testing
  - Card Sorting
    - Used for testing navigation structure.
    - This testing determines what to call various screens, pages or functions and how to group them.
  - In-Person Testing
    - Run by one or more observers in fixed environment.
    - Users are asked to accomplish certain tasks and observers can interact with them and ask questions.
  - Remote Testing
    - Users perform various tasks in their own environment.
    - Their attempt to accomplish a task is recorded via browser webcam.

- Compatibility testing: (2 slides)
  - This testing is done to ensure a competitive edge in terms of quality, compatibility, cost and delivery .
  - To ensure customer satisfaction across multiple platforms.
  - This is a non-functional testing ensures compatibility with browsers, hardware, users, OS, mobile devices, networks.
  - Can be performed through automation or manually .

- Need of Compatibility testing: compatibility testing improves reputation of the company, improves sales, brings delight to customers, also ensures stability of software.
- Types of Compatibility testing:
  - There are two types of Compatibility testing

  - Forward Compatibility testing: verifies that the software is compatible with newer and upcoming versions.

  - Backward Compatibility testing: this checks weather current version seamlessly works with older versions of hardware or software.

- Performance Testing: (3 slides)
- Testing that ensures that software product will perform good under expected workload.

- It identifies how system performs in terms of responsiveness and

stability under certain conditions.

- Objective is to remove performance bottlenecks.
- Performance depends upon
- Speed : how quickly software responds.
- Scalability : maximum number of users that can access the software at the same time.
- Stability : is software product stable under changing load.
- Performance testing is also called as “Perf Testing”.
- It is subset of performance engineering.

- Test objectives frequently include the following:
  - Response time. For example, the product catalogue must be displayed in less than 3 seconds.
  - Throughput. For example, the system must support 100 transactions per second.
  - Resource utilization. A frequently overlooked aspect is the amount of resources your application is consuming, in terms of processor, memory, disk input output (I/O), and network I/O.

- Types of performance testing:
  - Load testing.
  - Stress testing.
  - Endurance testing.
  - Spike testing.
- Performance testing tools- WEBLOAD, Load View, Open STA etc.

- Load testing (4 slides)
- This testing specifies performance of system under real-life load conditions.
- It measures websites quality of service performance based on actual customer behaviour.
- Load Testing is type of performance testing to check system with constantly increasing the load on the system until the time load is reaches to its threshold value.
- Here Increasing load means increasing number of concurrent users, transactions & check the behaviour of application under test.
- It is normally carried out underneath controlled environment in order to distinguish between two different systems.

- The main purpose of load testing is to monitor the response time and staying

power of application when system is performing well under heavy load.

- The successfully executed load testing is only if the specified test cases are executed without any error in allocated time.
- Load testing is testing the software under customer expected load.

- In order to perform load testing on the software you feed it all that

it can handle. Operate the software with largest possible data files.

- If the software operates on peripherals such as printer, or communication ports, connect as many as you can.
- If you are testing an internet server that can handle thousands of simultaneous connections, do it. With most software it is important for it to run over long periods.
- Some software‘s should be able to run forever without being restarted. So Time acts as a important variable.
- Load testing can be best applied with the help of automation tools.

- Simple examples of load testing:
  - Testing printer by sending large job.
  - Editing a very large document for testing of word processor.

- Continuously reading and writing data into hard disk.
- Running multiple applications simultaneously on server.
- Testing of mail server by accessing thousands of mailboxes.

•In case of zero-volume testing & system fed with

zero load.

- Advantages of Load Testing:
  - Discovery of bottleneck before deployment.
  - Enhance the scalability of a system.
  - Reduced risk for system down time.
  - Improved customer satisfaction.
  - Reduced failure cost.

- Stress testing : (3slides)
- Stress Testing is performance testing type to check the stability of software when hardware resources are not sufficient like CPU, memory, disk space etc.
- It is performed to find the upper limit capacity of the system and also to determine how the system performs if the current load goes well above the expected maximum.
- Main parameters to focus during Stress testing are “Response Time” and “Throughput”.

- Stress testing is Negative testing where we load the software with

large number of concurrent users/processes which cannot be handled by the systems hardware resources.

- This testing is also known as Fatigue testing
- Stress testing is testing the software under less than ideal conditions.
- So subject your software to low memory, low disk space, slow cps, and slow modems and so on. Look at your software and determine what external resources and dependencies it has.
- Stress testing is simply limiting them to bare minimum. With stress testing you starve the software.

- For e.g. Word processor software running on your computer with all

available memory and disk space, it works fine. But if the system runs low on resources you had a greater potential to expect a bug. Setting the values to zero or near zero will make the software execute different path as it attempt to handle the tight constraint.

Ideally the software would run without crashing or losing data.

- Advantages of Stress testing
- It helps developer to know system functionalities and capabilities to handle multiple tasks at a time.
- Helps to control and manage bugs
- Allows establishing application- monitoring triggers to warn about incoming failures
- Determines side effects of common h/w or supporting application failure.

- Security testing: (6 slides)
- Security testing is a testing technique to determine if an information system protects data and maintains functionality as intended.

  - Aims at
    - Authenticity
    - Authorization
    - Availability
    - Confidentiality
    - Integrity
    - Non-repudiation (avoid refusing)

- Confidentiality : A security measure which protects against the disclosure of information to parties other than the intended recipient is by no means the only way of ensuring the security.
- Integrity : Integrity of information refers to protecting information from being modified by unauthorized parties.

- Authentication : This might involve confirming the identity

of a person, tracing the origins of an artifact, ensuring that a product is what its packaging and labelling claims to be, or assuring that a computer program is a trusted one.

- Availability : Assuring information and communications services will be ready for use when expected. Information must be kept available to authorized persons when they need it.

- Non-repudiation (acknowledgment) : In reference to digital

security, non-repudiation means to ensure that a transferred message has been sent and received by the parties claiming to have sent and received the message. Non-repudiation is a way to guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message.

- Example :
- A Student Management System is insecure if 'Admission' branch can edit the data of 'Exam' branch.
- An ERP system is not secure if DEO (data entry operator) can generate 'Reports'.
- An online Shopping Mall has no security if customer's Credit Card Detail is not encrypted.
- A custom software possess inadequate security if an SQL query retrieves actual passwords of its users .

Why Security testing is required ?

- It is important in each and every stages of SLDC (Software Development Life Cycle)
- Requirement Phase: Security analysis of all requirements
- Design Phase: Implementation of Test plan including Security Tests.
- Code and Unit testing: Security White box testing
- Integration Testing: Black box testing
- System Testing: Black box testing and Vulnerability Scanning.

- Types of Security testing:
  - Vulnerability Scanning.
  - Security Scanning.
  - Penetration Testing.
  - Risk Assessment.
  - Security Auditing.
  - Ethical Hacking.

- Vulnerability Scanning:
- Also known as vulnerability assessment
- Identify and report the vulnerability using vulnerability scanning tools
- This is first step to improve security of the system
- Reports should contain: title, description and severity of vulnerability
- Security Scanning:
- Done to find weak points in the security of network and system
- Also provides solutions to reduce these risks
- Penetration Testing:
- Also called as pen test
- Identify vulnerability and attempt to exploit them using penetration tools
- 3 techniques:
  - Manual
  - Automated
  - Combination of both

- Risk Assessment:
- Retrieving and analysing security risk
- Later prioritized as low, medium, high
- Also recommends possible ways to prevent risks
- Security Auditing:
- Procedure of defining security flaws
- It is an internal inspection of the system
- Done by line by line inspection of the code
- Ethical Hacking:
- Done on the system with intent to find and expose issues in the system
- Done by White Hat Hacker
- Who uses their skills to reveal defects of the system

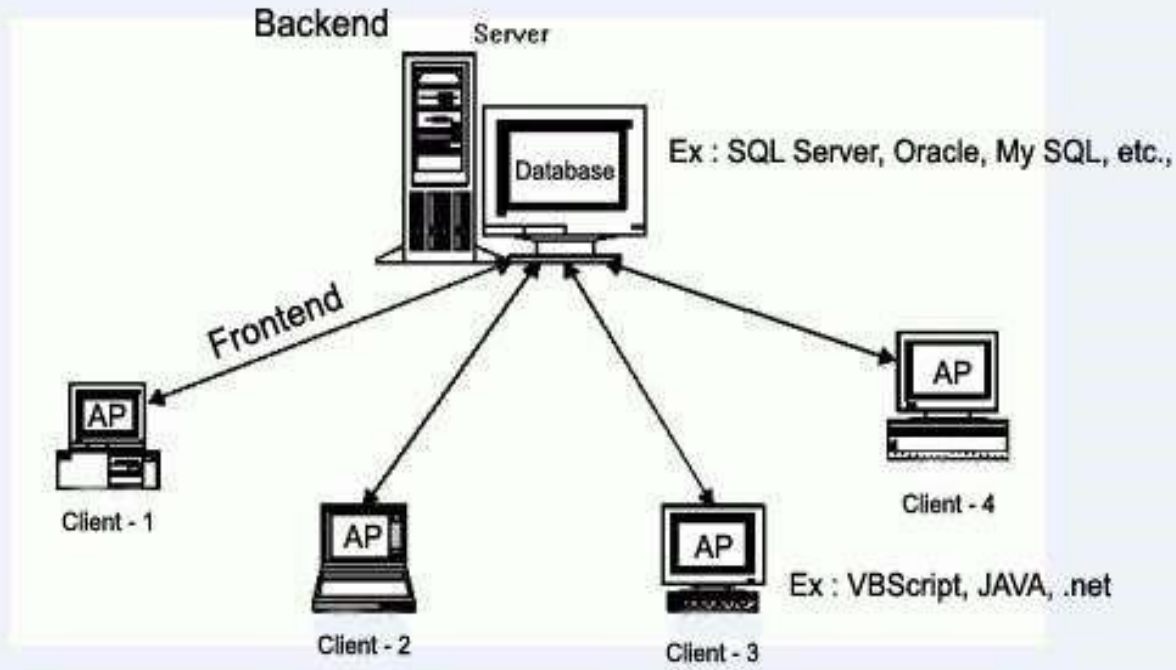
What are white, gray, and black hat hackers?

White hat hackers: Hack into a company's systems with permission to do so in order to enhance its security.

Gray hat hackers: Hack into a company's systems without permission but don't have malicious intents and may disclose their findings to the affected business.

Black hat hackers: Hack into a company's systems without permission and with malicious intent. Exploits security vulnerabilities.

Client - Server Architecture



- Client Server Testing: (6 slides)

i) This type of testing usually done for 2 tier applications (usually developed for LAN)

Here we will be having front-end and backend.

ii) The application launched on front-end will be having forms and reports which will be monitoring and manipulating data.

E.g: applications developed in VB, VC++, Core Java, C, C++, D2K, Power Builder etc.

iii) The backend for these applications would be

MS Access, SQL Server, Oracle, Sybase, MySQL, Quadbase.

iv) The tests performed on these types of applications would be

- User interface testing.
- Manual support testing.
- Functionality testing.
- Compatibility testing & configuration testing.
- Intersystem testing.

- The approaches used for client server testing are :

- 1. User interface testing: User interface testing, a testing

technique used to identify the presence of defects is a product/software under test by using Graphical user interface [GUI].GUI Testing - Characteristics:

i) GUI is a hierarchical, graphical front end to the application, contains graphical objects with a set of properties.

ii) During execution, the values of the properties of each objects of a GUI define the GUI state.

iii)It has capabilities to exercise GUI events like key press/mouse click.

iv) Able to provide inputs to the GUI Objects.

v) To check the GUI representations to see if they are consistent with the expected ones .

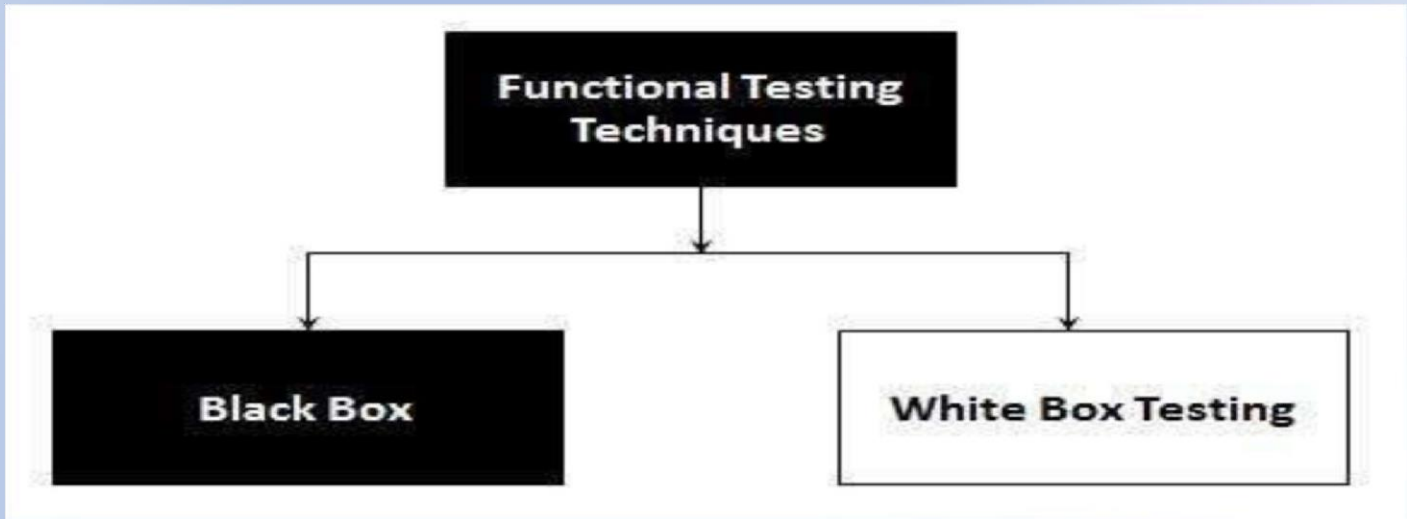
vi) It strongly depends on the used technology.

- 2. Manual testing :

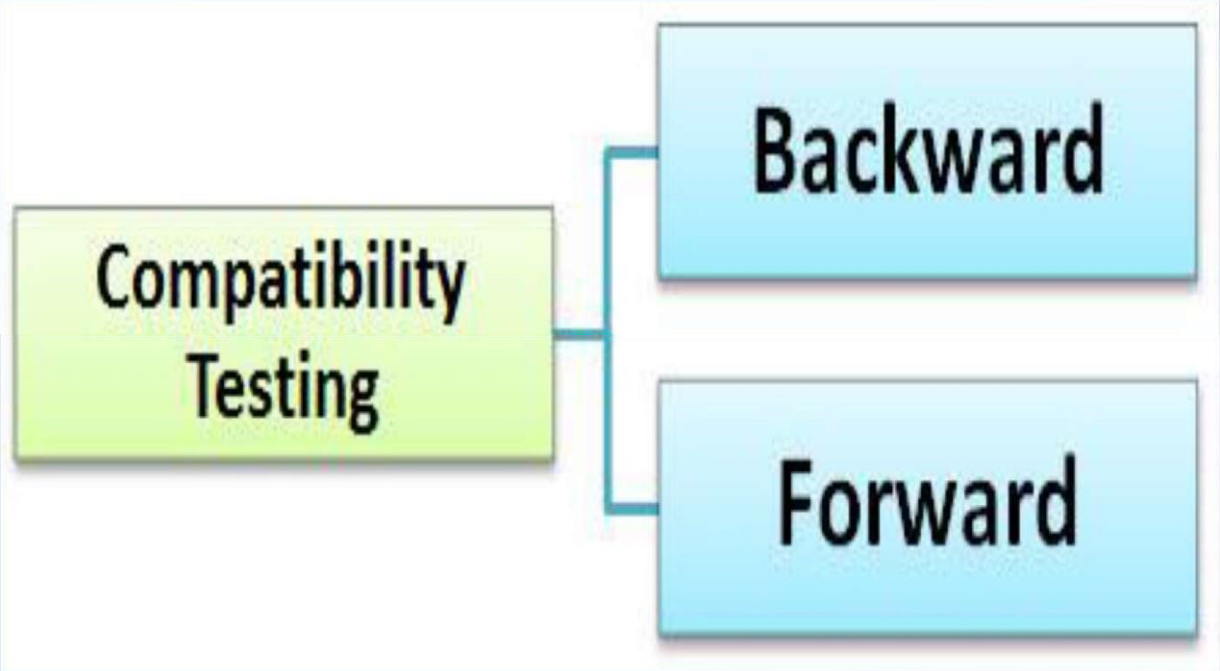
  Manual testing is a testing process that is carried out manually to find defects without the usage of tools or automation scripting. A test plan document is prepared that acts as a guide to the testing process to have the complete test coverage.

- Following are the testing techniques that are performed manually during the test life cycle are Acceptance Testing, White Box Testing, Black Box Testing, Unit Testing, System Testing, Integration Testing.

- 3. Functional testing: Functional Testing is a testing technique that is used to test the features/functionality of the system or Software, should cover all the scenarios including failure paths and boundary cases.
- There are two major Functional Testing techniques as shown below:



- 4. Compatibility testing: Compatibility testing is a non-functional testing conducted on the application to evaluate the application's compatibility within different environments. It can be of two types - forward compatibility testing and backward compatibility testing.



- 1. Forward Compatibility Testing: This type of testing verifies that the software is compatible with the newer or upcoming versions, and is thus named as forward compatible.

- 2. Backward Compatibility Testing: This type of testing helps to

check whether the application designed using the latest version of an environment also works seamlessly in an older version. It is the testing performed to check the behavior of the hardware/software with the older versions of the hardware/software.

- Operating system Compatibility Testing - Linux , Mac OS, Windows
- Database Compatibility Testing - Oracle SQL Server
- Browser Compatibility Testing - IE , Chrome, Firefox
- Other System Software - Web server, networking/ messaging tool, etc.
- Acceptance Testing (6 slides)
- Acceptance Testing is a level of the software testing where a system is tested for acceptability.
- The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

- Usually, Black Box Testing method is used in Acceptance Testing.
- Acceptance Testing is performed after System Testing and before making the system available for actual use.

- The acceptance test cases are executed against the test data or using an

acceptance test script and then the results are compared with the expected ones.

- The goal of acceptance testing is to establish confidence in the system.
- Acceptance testing is most often focused on a validation type testing.

- Acceptance Criteria :

 Acceptance criteria are defined on the basis of the following attributes:

  - Functional Correctness and Completeness.
  - Data Integrity.
  - Data Conversion.
  - Usability.
  - Performance.
  - Timeliness.
  - Confidentiality and Availability.
  - Install ability and Upgradability.
  - Scalability.
  - Documentation.

- Types Of Acceptance Testing :
- User Acceptance test
- Operational Acceptance test
- Contract Acceptance testing
- Compliance acceptance testing

- User Acceptance test :
- It focuses mainly on the functionality thereby validating the fitness-for-use of the system by the business user. The user acceptance test is performed by the users and application managers.
- Operational Acceptance test :

- Also known as Production acceptance test validates whether the

system meets the requirements for operation. In most of the organization the operational acceptance test is performed by the system administration before the system is released. The operational acceptance test may include testing of backup/restore, disaster recovery, maintenance tasks and periodic check of security vulnerabilities.

- Contract Acceptance testing :
- It is performed against the contract's acceptance criteria for producing custom developed software. Acceptance should be formally defined when the contract is agreed .

- Compliance acceptance testing :
- It is also known as regulation acceptance testing is performed against the regulations which must be adhered to, such as governmental, legal or safety regulations.

- Advantages Of Acceptance Testing :
- The functions and features to be tested are known.
- The details of the tests are known and can be measured.
- The tests can be automated, which permits regression testing.
- The progress of the tests can be measured and monitored.
- The acceptability criteria are known.

- Disadvantages Of Acceptance Testing :
- Requires significant resources and planning.
- The tests may be a re-implementation of system tests.

- It may not uncover subjective defects in the software, since you are

only looking for defects you expect to find.

- Alpha Testing (2 slides)

- Alpha Testing is a type of testing conducted by a team of highly

skilled testers at development site. Minor design changes can still be made as a result of alpha testing.

- For Alpha Testing there is a dedicated test team.
- Alpha testing is final testing before the software is released to the general public. It has two phases:
- In the first phase of alpha testing, the software is tested by inhouse developers. They use either debugger software, or hardware-assisted debuggers.
- The goal is to catch bugs quickly.
- In the second phase of alpha testing, the software is handed over to the software QA staff, for additional testing in an environment that is similar to the intended use.
- Pros Of Alpha Testing :
- Helps to uncover bugs that were not found during previous testing activities
- Better view of product usage and reliability

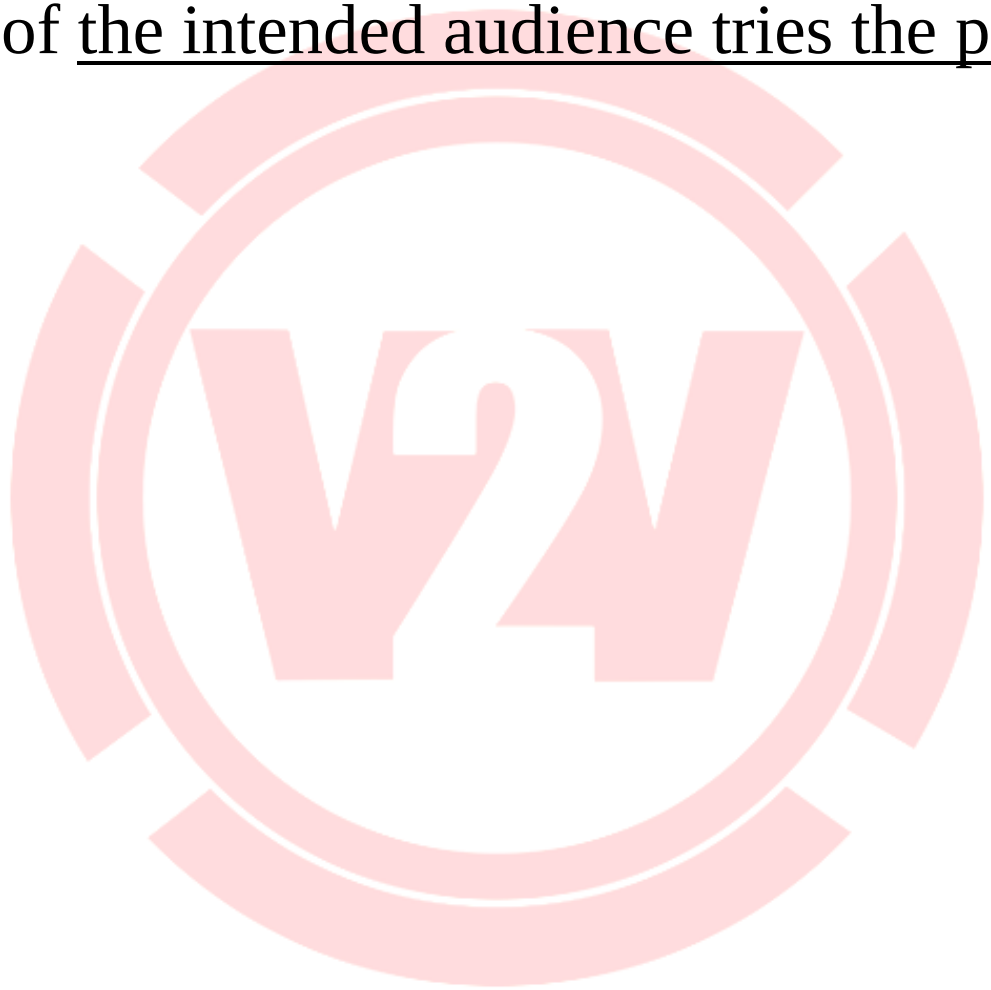
- Analyze possible risks during and after launch of the product
- Helps to be prepared for future customer support
- Helps to build customer faith on the product

- Maintenance Cost reduction as the bugs are identified and fixed before Beta /

Production launch

- Easy Test Management.
- Reduces delivery time
- Cons Of Alpha Testing :
- Not all the functionality of the product is expected to be tested.
- Only Business requirements are scoped.
- Cannot furnish requirements of the real world
- Done by developers perspective not clients
- Beta Testing (2 slides)
- Beta Testing is also known as field testing. It takes place at customer's site.
- It sends the system/software to users who install it and use it under real-world working conditions.

- A beta test is the second phase of software testing in which a sampling of the intended audience tries the product out



- The goal of beta testing is to place your application in the

hands of real users outside of your own engineering team to discover any flaws or issues from the user's perspective that you would not want to have in your final, released version of the application.

- Beta testing can be considered "pre-release testing".
- Advantages of beta testing
- You have the opportunity to get your application into the hands of users prior to releasing it to the general public.
- Users can install, test your application, and send feedback to you during this beta testing period.
- Your beta testers can discover issues with your application that you may have not noticed, such as confusing application flow, and even crashes.

- Using the feedback you get from these users, you can fix problems before it is released to the general public.

- The more issues you fix that solve real user problems, the higher the

quality of your application when you release it to the general public.

- Having a higher-quality application when you release to the general public will increase customer satisfaction.
- These users, who are early adopters of your application, will generate excitement about your application.

- Disadvantages of Beta Testing:
- Software engineer has no control over the process of testing , as it is performed by users in the real world environment
- It can be difficult to find end users wiling to test the software
- It can be time consuming process and can delay the final release of the product

- Types of Beta Testing:
  - Traditional Beta Testing
  - Public Beta Testing
  - Technical Beta Testing
  - Focused Beta
  - Post Release Beta

- Traditional Beta Testing: s/w product is provided to targeted users and associated data is collected. This data is useful for the improvement of the product.

- Public Beta Testing: product is released publicly in real world using

online channels and data is collected .using feedback from users improvements in product are done.

- Technical Beta Testing: s/w product is released in internal group of an organization and data is collected from employees of the organization

- Focused Beta: product is released in the market to collect feedback about specific features of the program

- Post Release Beta: product is released in the market and data is collected to improve the s/w product for the next release

Sr.No	Alpha testing .	Beta testing .
1	Performed at developer's site	Performed at end user's site

2	Performed in controlled environment in developers presence	Performed in uncontrolled environment in developers absence
3	Less probability of finding errors as it is driven by developer	High probability of finding errors as it is used by end user.
4	It is done during implementation phase of software	It is done at the pre-release of the software
5	It is not considered as live application of software	It is considered as a live application of the software.
6	Less time consuming as developer can make necessary changes in given time	More time consuming as user has to report the bugs if any via appropriate channels.
7	Alpha testing involves both white box and black box testing	Beta testing typically uses black box testing only
8	Long execution cycles may be required for alpha testing	Only a few weeks of execution are required for beta testing

- Special tests (6 slides) • 1. Regression Testing :
- Regression Testing is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features.
- Regression Testing is nothing but full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine.

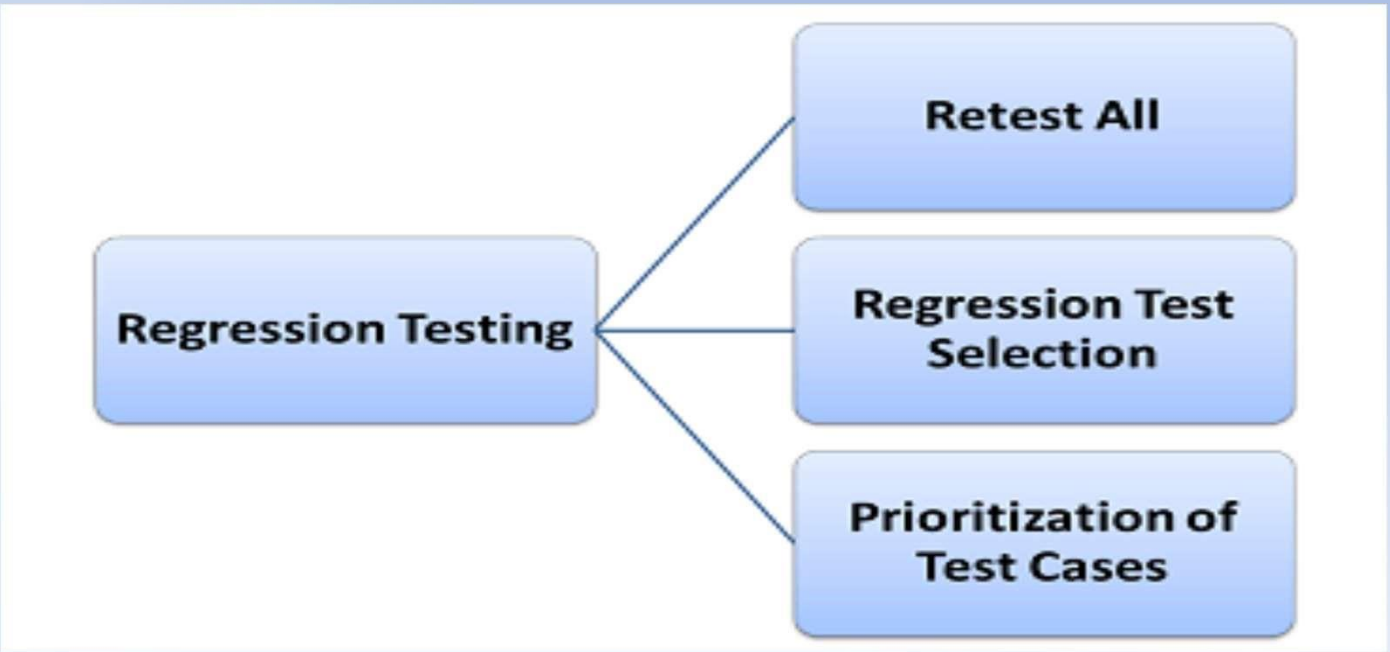
- This testing is done to make sure that new code changes should not have

side effects on the existing functionalities. It ensures that old code still works once the new code changes are done.

- Regression Testing is required when there is a Change in requirements and code is modified according to the requirement

  .

- New feature is added to the software :
- Defect fixing .
- Performance issue fix.



Retest All

- This is one of the methods for Regression Testing in which all the tests in the existing test bucket or suite should be re-executed This is very expensive as it requires huge time and resources.

Regression Test Selection

- Instead of re-executing the entire test suite, it is better to select part of test suite to be run .
- Test cases selected can be categorized as 1) Reusable Test Cases 2) Obsolete Test Cases.
- Re-usable Test cases can be used in succeeding regression cycles.
- Obsolete Test Cases can't be used in succeeding cycles.

- Prioritization of Test Cases
  - Prioritize the test cases depending on business impact, critical & frequently used functionalities .

    Selection of test cases based on priority will greatly reduce the regression test suite.

- Selecting test cases for regression testing
  - It was found from industry data that good number of the defects reported by customers were due to last minute bug fixes creating side effects and hence selecting the Test Case for regression testing is an art and not that easy.

- Effective Regression Tests can be done by selecting following test cases –
- Test cases which have frequent defects
- Functionalities which are more visible to the users
- Test cases which verify core features of the product

- Test cases of Functionalities which has undergone more

and recent changes

- All Integration Test Cases
- All Complex Test Cases
- Boundary value test cases
- Sample of Successful test cases
- Sample of Failure test cases
- Regression Testing Tools
- If your software undergoes frequent changes, regression testing costs will escalate.
- In such cases, Manual execution of test cases increases test execution time as well as costs.
- Following are most important tools used for both functional and regression testing:

- Selenium: This is an open source tool used for automating web applications. Selenium can be used for browser based regression testing.
- Quick Test Professional (QTP): HP Quick Test Professional is automated software designed to automate functional and regression test cases. It uses VBScript language for automation. It is a Data driven, Keyword based tool.

- Rational Functional Tester (RFT): IBM's rational

functional tester is a Java tool used to automate the test cases of software applications. This is primarily used for automating regression test cases and it also integrates with Rational Test Manager.

- Advantages of Regression Testing
  - Promotes improvement of product quality
  - Ensures issues created are fixed
  - Ensures modification does not impact already tested code
  - Can be done during integration testing

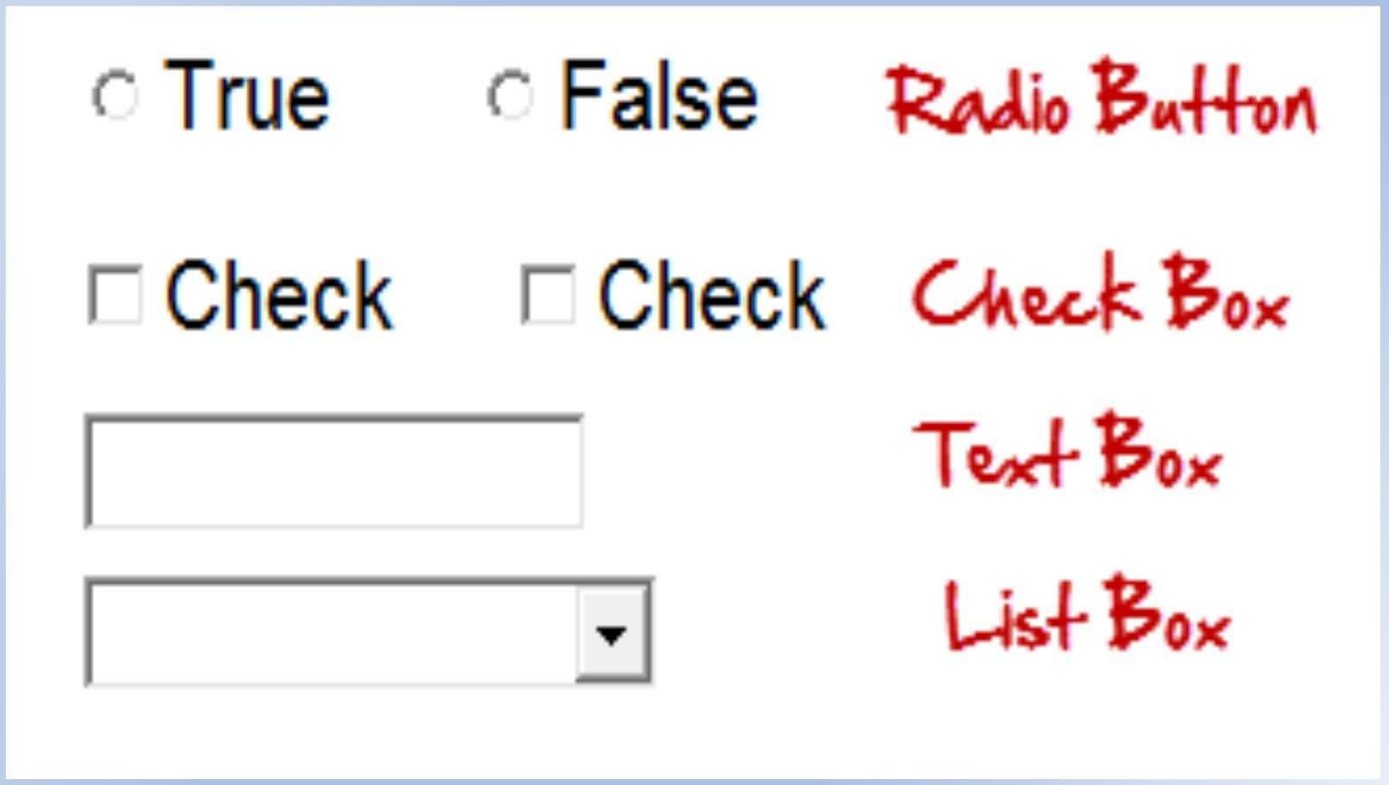
- Disadvantages of Regression Testing
  - Without automation it is hard to manage cost of regression testing
  - Automating regression requires skilled software engineers
  - Testing new features requires more test cases which will increase maintenance cost

- 2. GUI Testing (6 slides)
- There are two types of interfaces for a computer application.
- Command Line Interface is where you type text and computer responds to that command.
- GUI stands for Graphical User Interface where you interact with the computer using images rather than text.

- GUI testing is the process of testing the system's

Graphical User Interface of the Application Under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars - toolbar, menu bar, dialog boxes and windows, etc.

- GUI is what user sees. A user does not see the source code. The interface is visible to the user. Especially the focus is on the design structure, images that they are working properly or not.
- GUI Testing Guidelines
  1. Check Screen Validations.
  2. Verify All Navigations .
  3. Check usability Conditions.
  4. Verify Data Integrity.
  5. Verify the object states .
  6. Verify the date Field and Numeric Field Formats .
  7. Ensure messages displayed accurately
  8. Validate font size
  9. Check alignment
  10. Check quality and clarity of image
  11. Check GUI for different screen resolutions


True
False
Radio Button
Check
Check
Check Box
Text Box
List Box

- In addition to functionality, GUI testing evaluates design elements such as

layout, colors, fonts, font sizes, labels, text boxes, text formatting, captions, buttons, lists, icons, links and content. GUI testing processes can be either manual or automatic, and are often performed by third party companies, rather than developers or end users.

- Example: Consider any website like MSBTE, Google, yahoo or any login form or GUI of any application to be tested. It includes following:
- All colors used for background, control colors, and font color have a major impact on users. Wrong color combinations and bright colors may increase fatigue of users.
- Error messages and information given to users must be usable to the user. Reports and outputs produced either on screen or printed should be readable. Also paper size on printer, font, size of screen should be consider.
- Screen layout in terms of number of instructions to users, number of controls and number of pages are defined in low level design. More controls on single page and more pages reduce usability

- Types of control on a single page are very useful considering usability.

- Number of images on page or moving parts on screen may affect

performance. These are high-priority defects. It has direct relationships with usability testing, look, and feels of an application. It affects emotions of users and can improve acceptability of an application. .

- Advantages of GUI Testing:
- Good GUI improves feel and look of the application; it psychologically accepts the application by the user.

- GUI represents a presentation layer of an application. Good GUI helps an

application due to better experience of the users.

- Consistency of the screen layouts and designs improves usability of an application.
- Disadvantages of GUI Testing:
- When number of pages is large and number of controls in a single page is huge requires more memory resources.
- Special application testing like those made for blind people or kids below age of five may need special training for testers.
- Limited or no access to source code –makes process of testing difficult

- Different Ways of GUI Testing

- 3 different ways of GUI testing
  - Manual Based Testing
  - Record And Replay
  - Model Based Testing

1. Manual Based Testing:

- Tester checks all graphics with pre-requisites in business document manually

2. Record And Replay:

- This is an automated GUI testing tool
- All tasks are recorded during test
- Recorded tasks are executed with expected behaviour
- This can be repeated several times with various data sets

3. Model Based Testing:

- This acts as graphics description
- It predicts the system behaviour and technique generates the test cases efficiently
- Charts and decision tables are model based techniques

Client/Server Testing:

1	With help of diagram describe client-server testing.	S V 4 M

2

Explain the client-server application testing

3	Explain Client-Server testing.	S V 4 M

- Client server is software architecture consists of client and server systems which communicate to each other over the computer network or the same machine.

- In such architecture the client system sends the request to the server system and the server system sends the response to the client system.

- The application launched on front-end will be having forms and reports which will be monitoring and manipulating data of database.
- Such applications are developed in Visual Basic, VC++, C, C++, Core JAVA, etc.
- Backend database could be BM DB2, MS Access, Oracle, Sybase, SQL Server, Quad base, MySQL, etc.
- Figure

Client-server architecture.

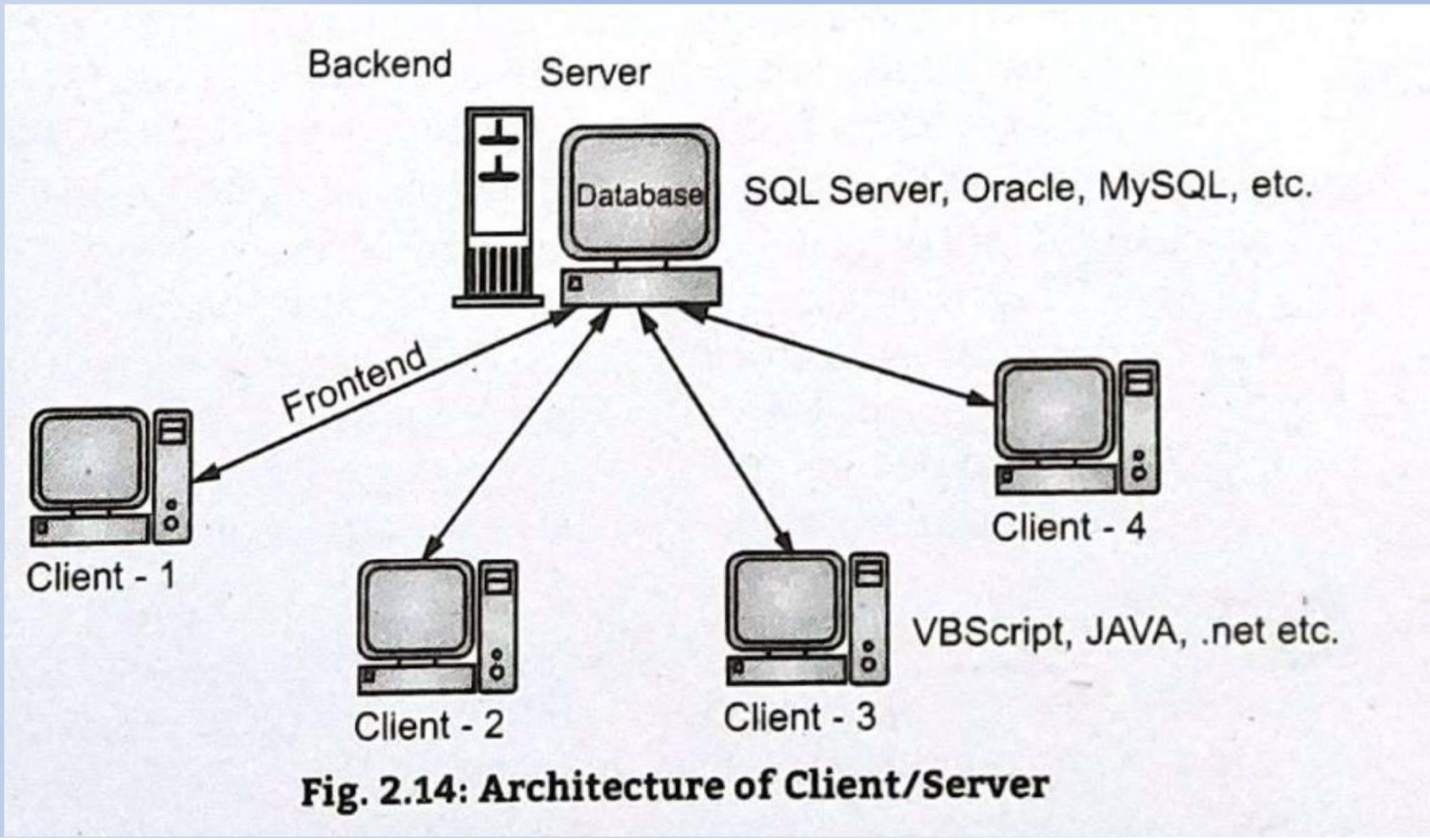


Fig. 2.14: Architecture of Client/Server

- The tests performed on these types of applications would be as follows:
  - User interface testing
  - Manual support testing
  - Functionality testing
  - Compatibility testing and configuration testing
  - Intersystem testing

- Client-server testing approaches:

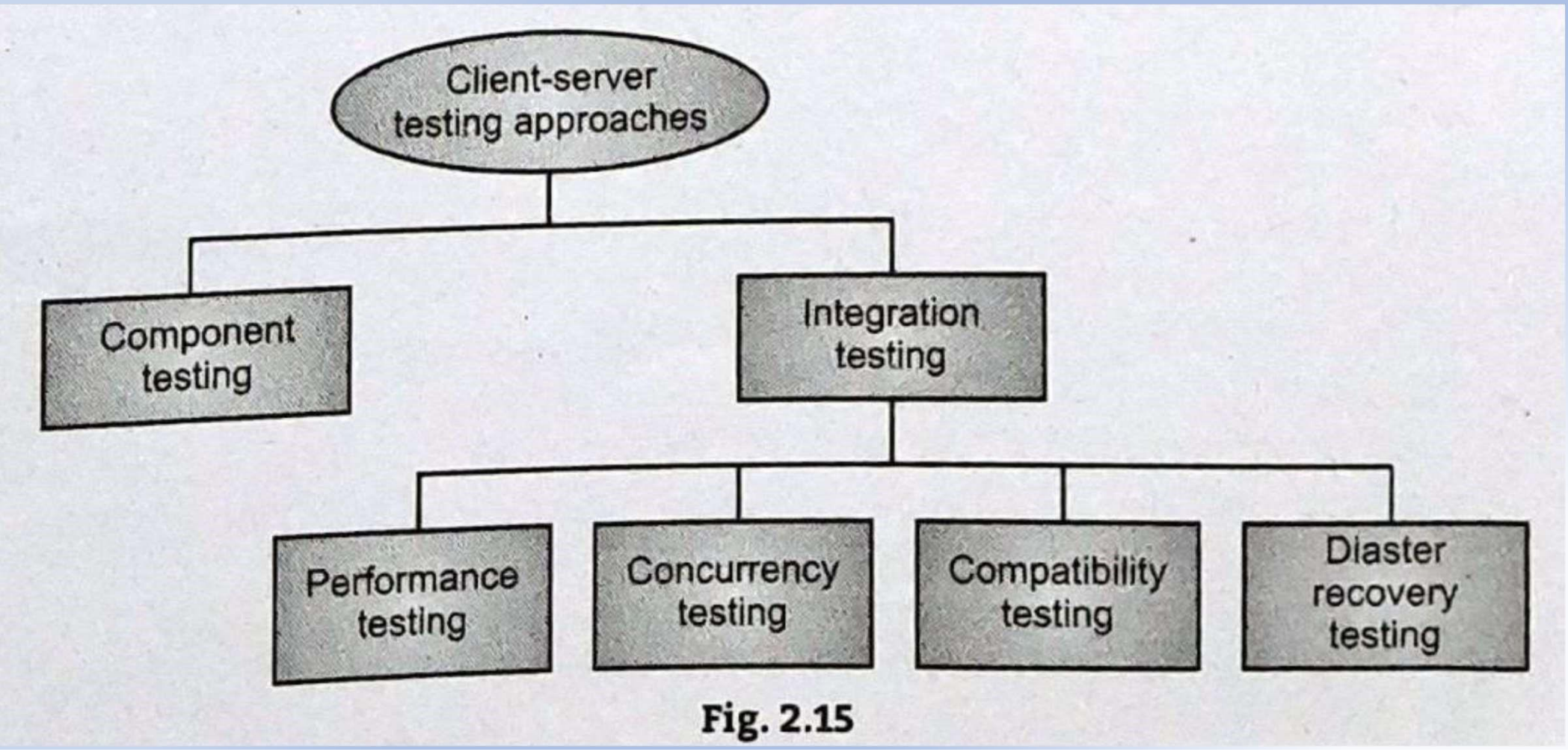


Fig. 2.15

- Above chart shows following testing for client/server system:

1.Component Testing:

- needs to define the approach and test plan for testing client and server individually
- It is related to simulator.
- We may have to test network by using client and server simulators at a time.

- 2. Integration Testing:

- After successful testing of server, clients and network, they are

brought together to form the system and system test cases are executed.

- Communication between client and server is tested in integration testing.

There are several special testing involved in client-server application.

Some of them are given below:

1	Performance Testing	▯System Performance is tested when numbers of clients are communicating with server at a time. ▯Similarly, volume testing and stress testing may be used for testing client-server applications.
2	Concurrency Testing	▯It may be possible that multiple users may be accessing same record at a time and Concurrency testing is required to understand the behavior of a system under such circumstances.

3	Disaster Recovery/Business Continuity Testing	There exists a possibility of breaking of the communication due to various reasons or failure of either client or server or link connecting them. Test for disaster recovery and business continuity may be involved to understand how system behaves in such case of disaster.

4	Testing for Extended Periods:	▯ Generally server is never shut down unless need for shut down for maintenance. It may be expected that server is running 24 X 7 for extended period. ▯ One needs to conduct testing over an extended period to understand if service level of network and server deteriorates over a time due to some reasons like memory leakage.

5	Compatibility Testing:	□When servers are placed in different environments and the users are using them in production. □Testing must ensure that Performance is maintained on the range of hardware and software configurations and users must be adequately protected in case of configuration mismatch.

- Skills Required for Tester
- • Communication skills
- • Domain knowledge
- • Desire to learn
- • Technical skills
- • Analytical skills
- • Planning
- • Integrity
- • Curiosity
- • Think from users perspective
- • Be a good judge of your product

1.3.2 Software Tester

(S-17, 22, 23, W-16, 18)

- Software tester is a expert who conducts prescribed tests on software programs and applications prior to their implementation to ensure quality, design integrity and proper functionality.
- The tester's job is to find out defects so that they will be eventually fixed by developers before the software product goes to the user/customer.
- We can define software tester as, "a technician (person) who conducts prescribed tests on software programs and applications prior to their implementation to ensure quality, design integrity and proper functionality".
- A software tester must have complete knowledge about testing as a discipline. The goal of a software tester is to find bugs, find them as early as possible, and make sure they get fixed.
- The qualities/skills of software testers are listed below: (S-22, 23)

1. **They are explorers:** Software testers aren't afraid to project into unknown situations. They like to get a new piece of software, install it on PC, and see what happens.
2. **They are troubleshooters:** Software testers are good at thinking out why something doesn't work.
3. **They are relentless:** Software testers keep trying. They may see a bug that quickly disappears or is difficult to re-create. They will try every way possible to find it.
4. **They are creative:** Their job is to think up creative and even unusual approaches to find bugs.
5. **They are perfectionists:** They strive for perfection, but they know when it becomes unattainable and they're okay with getting as close as they can.
6. **They exercise good judgment:** Software testers need to make decisions about what they will test, how long it will take, and if the problem they're looking at is really a bug.
7. **They are tactful and diplomatic:** Good software testers know how to do so tactfully and professionally and know how to work with programmers who aren't always tactful and diplomatic.
8. **They are persuasive:** Bugs that testers find won't always be viewed as severe enough to be fixed. Testers need to be good at making their points clear, demonstrating why the bug does indeed need to be fixed, and following through on making it happen.

1.4 VERIFICATION AND VALIDATION (V MODEL) (S-19, W-19)