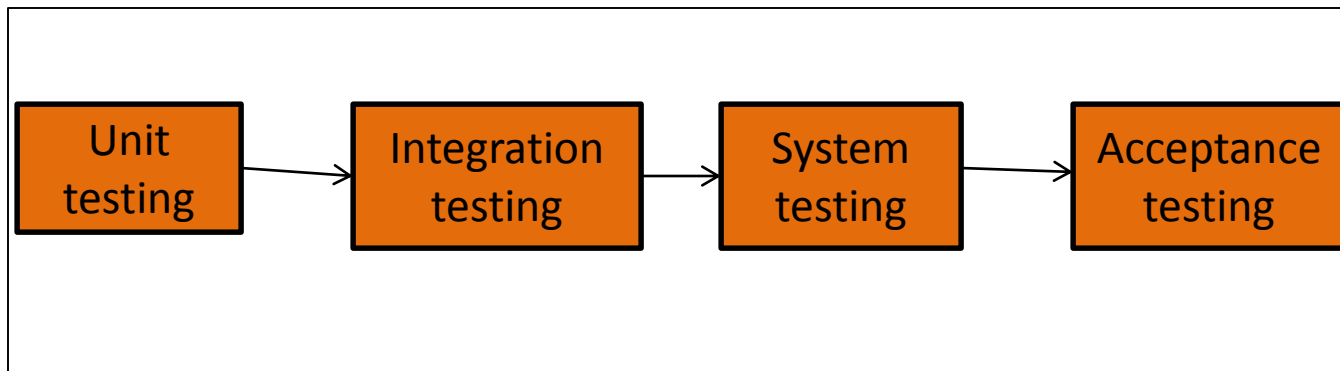


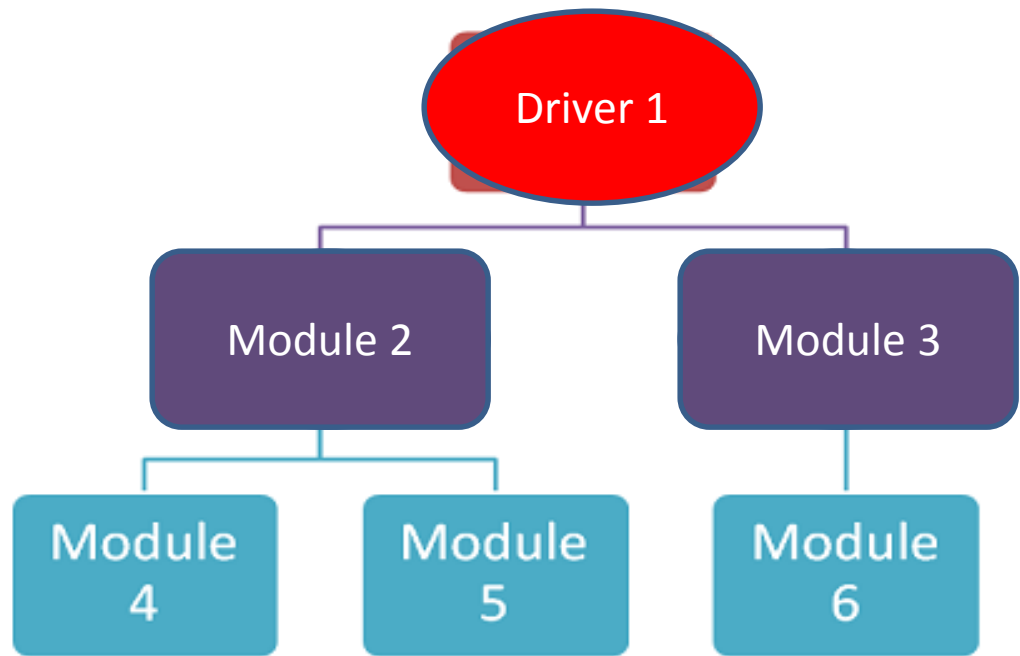
Unit II: Levels of testing & special test



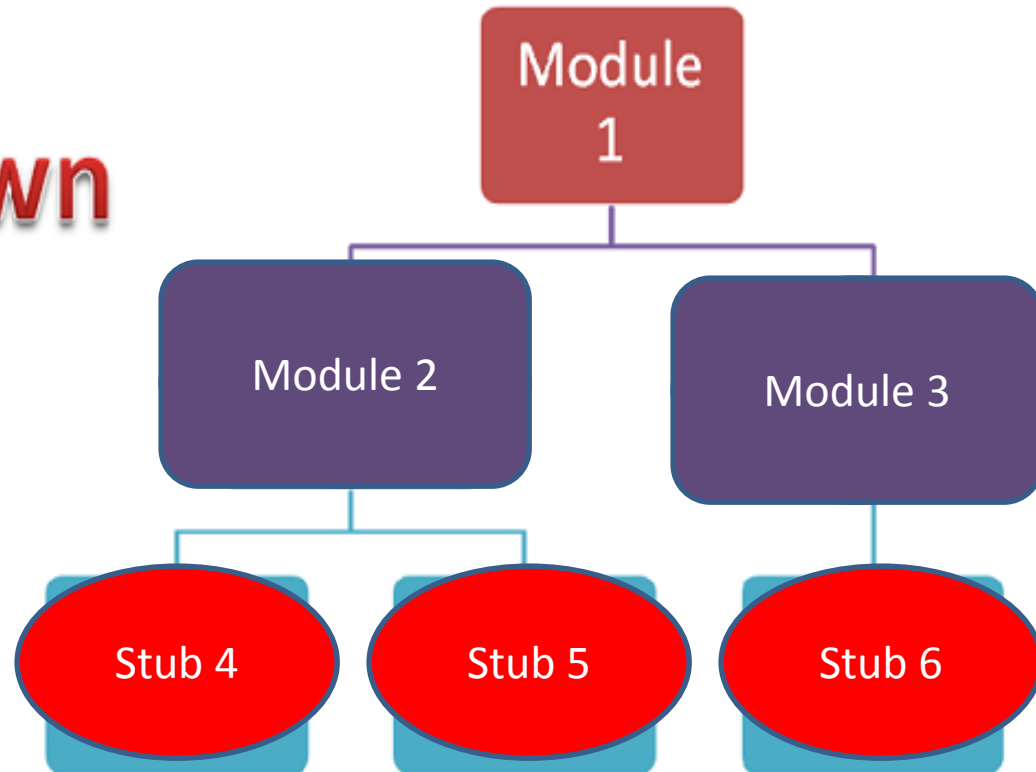
Unit testing

- Testing that occurs at lowest level
- Unit testing may include code file , classes , methods which can be tested individually for correctness.
- Drivers and stubs are written.
- Perform following functions
 - 1. tests all control paths.
 - 2. ensure all statements executed at least once.
 - 3. Test data structure.
 - 4. Check range of i/ps

**Bottom
Up**

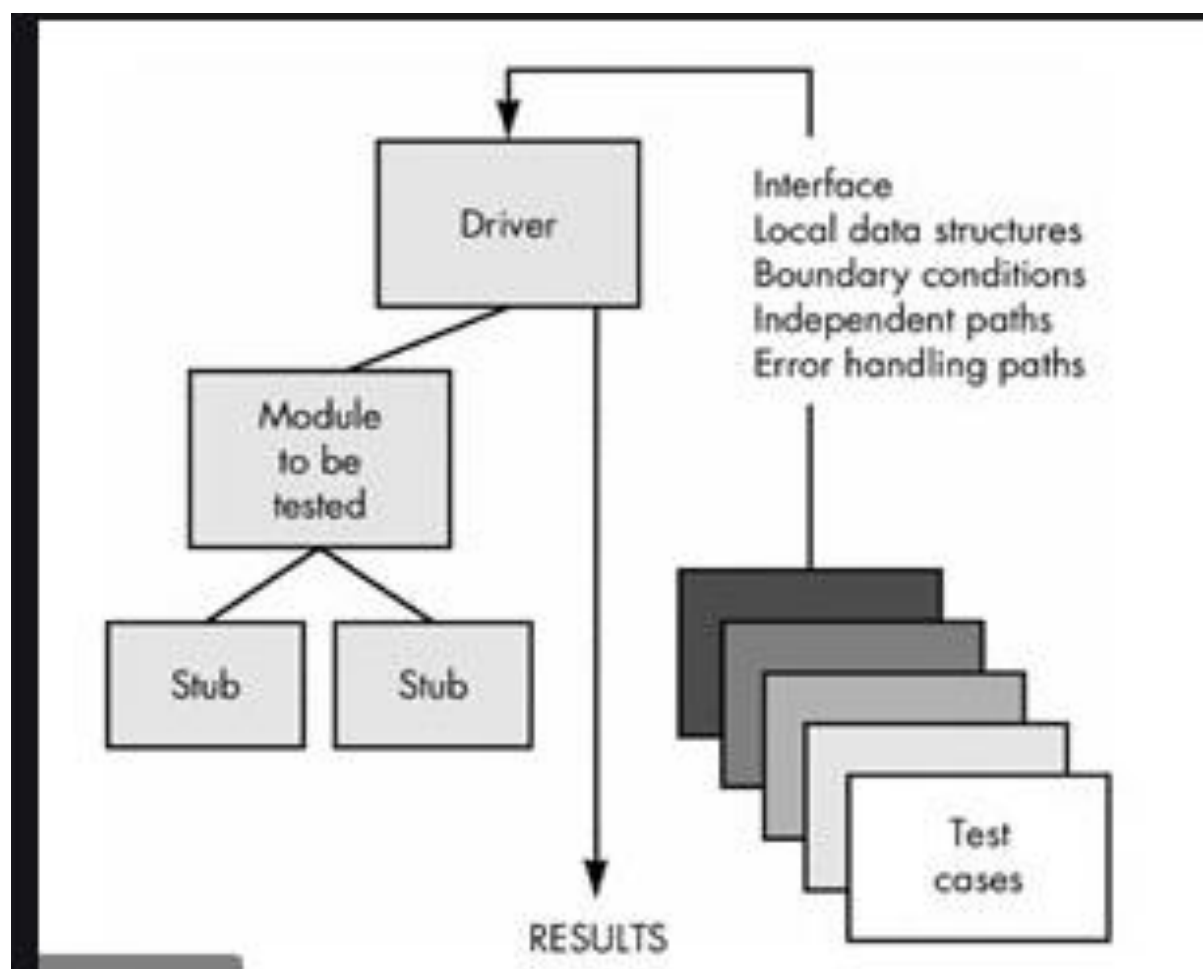


Top Down

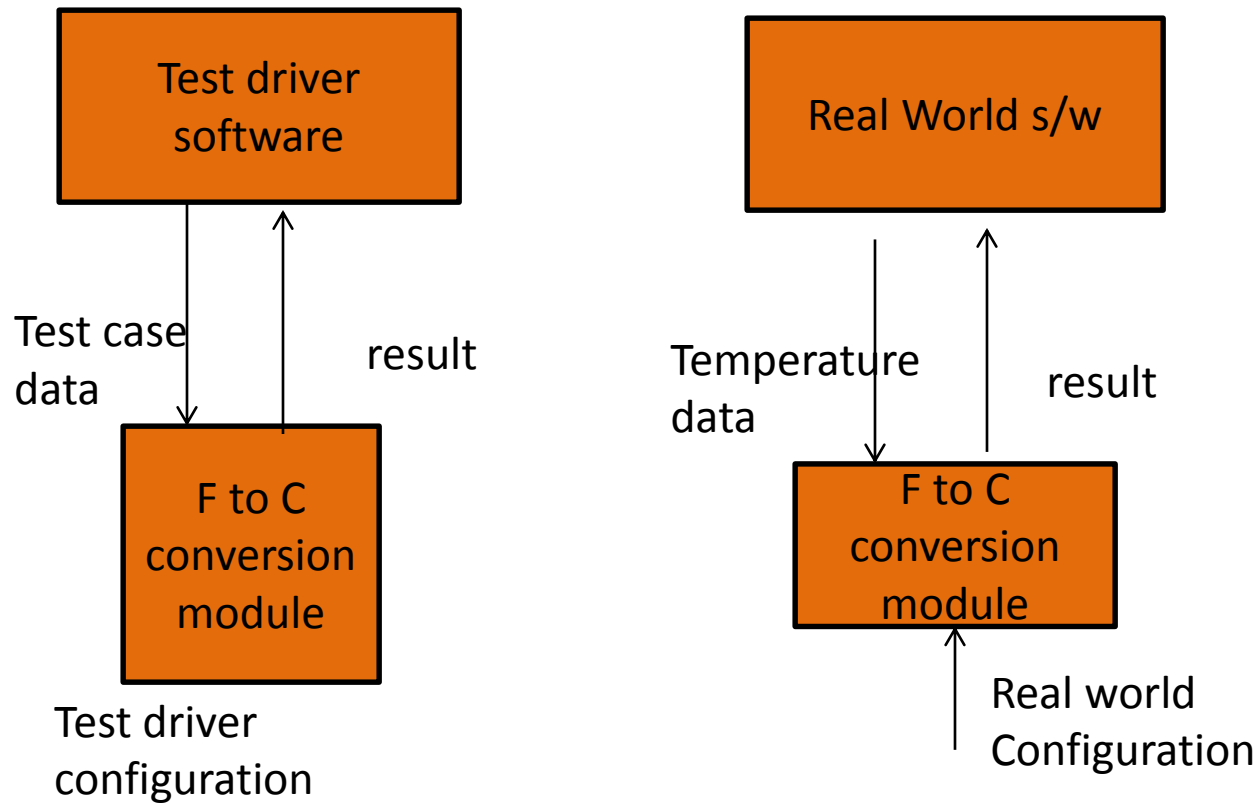


- Driver:
- simulate calling unit
- Act as main program that accept test cases data , passes such data to the components and print result.
- **Driver:** Calls the Module to be tested.
- Used in bottom up approach

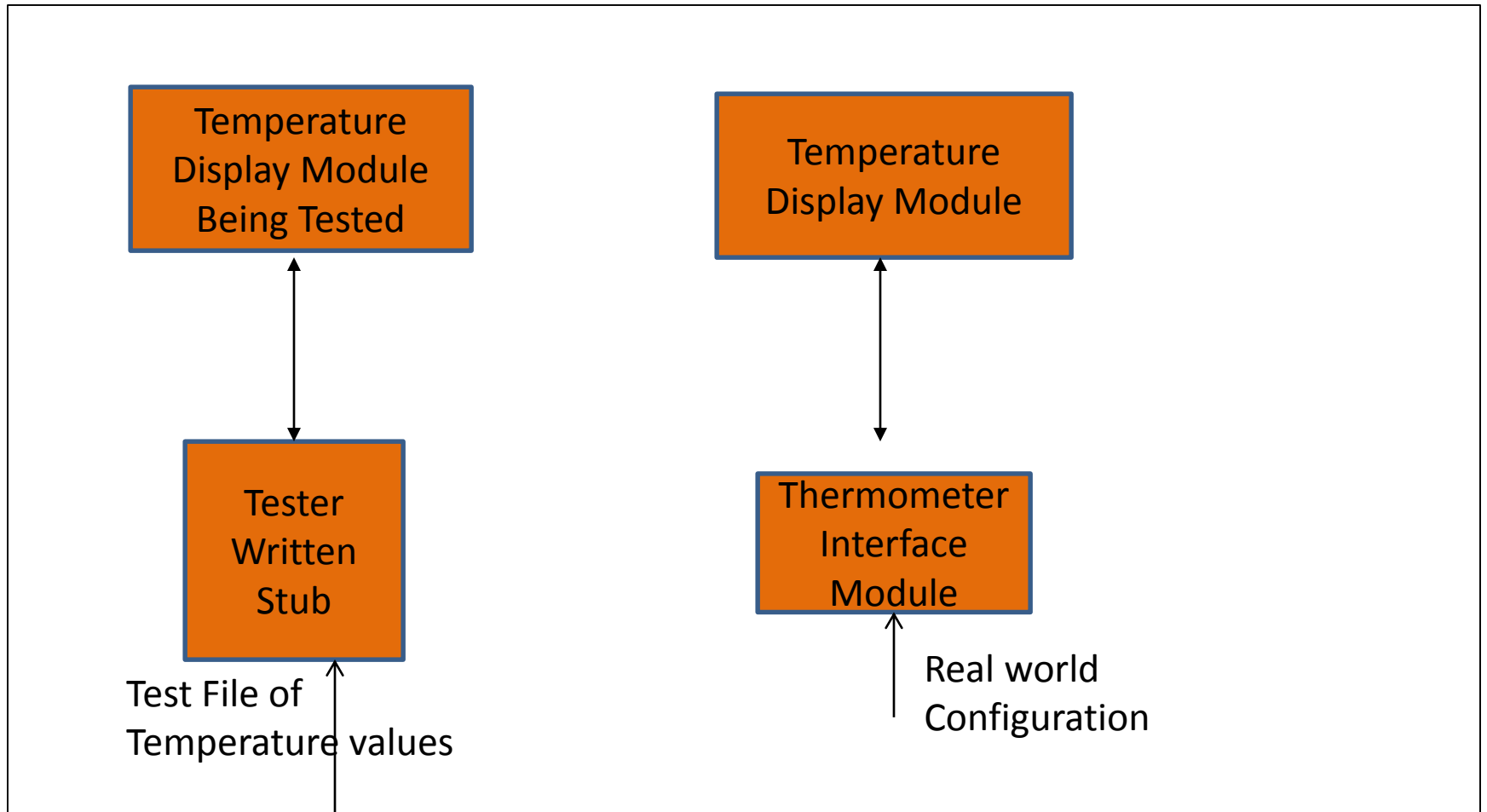
- Stubs :
- Simulate called unit
- Act as sub module that are subordinate the components to be tested.
- Dummy subprogram uses subordinate module interface , may do minimal data manipulation , print verification and return control to the tested module
- **Stub:** Is called by the Module under Test.
- Used in top down approach



Example of Driver



Example of Stub....



- Advantage of unit test:
 - Find problem early in the development cycle
 - Basic understanding of units.
-
- Disadvantage of unit test:
 - Initial time required for development
 - It will not catch integration errors

Driver

- Emulating a calling function
- Accept test case data and passes that data to the module that is being tested.
- Use in bottom up approach

Stub

- Emulating called function
- Are used to replace modules that are subordinate to the module to be tested.
- Use in top down approach

Integration testing

- Performed against group of modules
- Objective : take all tested individual modules, integrated them , test them again and develop s/w
- Checks parameters passing between units.
- Ensure that all modules work properly as per user requirements, when they are integrated
- When multiple modules are integrated , problem must be in module interaction.
- Advantage :
 1. easy to fix the error compared to system testing

Integration testing



Top down

Bottom up

Bi directional

1. Top down integration

- Testing begins from top level , progressively adds in lower level module one by one.
- As low level code added , stub will replace with actual components.
- It needs design and implementation of stubs, drivers may not be required ??

- **Advantage :**
- 1. feasibility of program can be determine easily at very early stage
- 2. doesn't require drivers.
- **Disadvantage:**
- 1. unit are rarely tested alone before their integration ??
- 2. It become difficult to exercise the top level routines in the desired manner.?? Bcoz , low level modules are absent
- 3. stubs are required ??
- Stubs are to be written and tested before they can be used in integrated testing.
- 4.It is difficult for other parties to perform this testing. Mostly developers will have to do it.

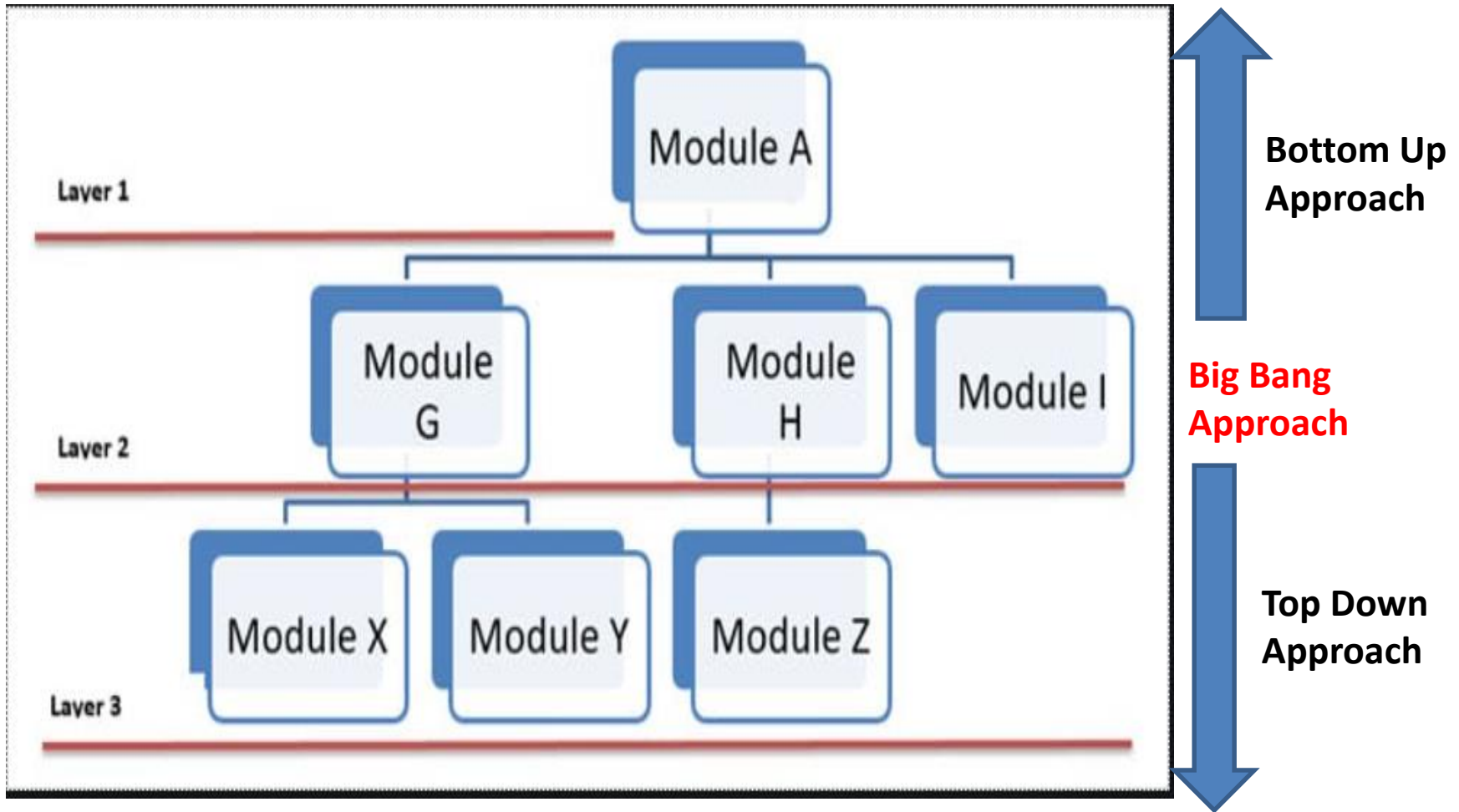
2. Bottom up integration

- Testing starts at the atomic modules level
- Each subsystem is tested separately and then the full system is tested.
- Steps :
 - 1. low level modules are combined into clusters (builds) that perform a specific s/w function.
 - 2. drivers are written
 - 3. build is tested
 - 4. drivers are removed clusters are combined moving upward .

- Advantage:
- 1. makes system more robust since each unit is tested first for its correctness.
- 2. doesn't require stubs.
- Disadvantage:
- 1. top level units are most important but tested last , pressure of delivery may cause problem of not completing testing

Bi directional integration

- Follows top down , bottom up approach either simultaneously or one after another.
- Sandwich overcome the shortcoming of top down , bottom up approaches.??
- Steps :
 - 1. bottom up testing start from middle layers and goes upward to the top layer.
 - 2. top down testing starts from middle layers and goes downward.
 - 3. big bang approach is followed for the middle layers . From this layers , bottom up approach goes upwards and top down approach goes downwards
- **Big Bang** : where all the modules integrated together at once and tested.



- Advantage :
- 1. suitable for very large project
- 2. overcome the short coming of T/D B/U approach.
- Disadvantage:
- 1. cant use for project which has huge interdependence between different modules.
- High cost.

- SWB : Inspection , Walkthrough , Peer review
- DWB : code coverage , code complex(flow graph) – consider one form
- SBB : rqmts analysis :
- DBB : positive / negative , boundary , equivalence – consider 5 forms

Testing on Web Applications

1. **Functionality Testing :**

- Checking the behavior of the s/w application against specification.
- a. Link testing : check different links on web page.
 - Visited link color , un visited link color , anchor link.
- b. Web form testing :
 - validation of form field , mandatory fields ,
 - check all possible operations , tab orders , default values field

2. Usability testing

- Content testing:
 - No spelling or grammatical errors.
 - Avoid Dark color theme.
 - No broken images.
- Navigation testing :
 - Menus , links , button
 - Navigation should be easy to use.
 - All options on header , footer left/right navigation should be consistent throughout the pages.

3. Compatibility testing

- -non functional testing
- -Browser compatibility test : web application should display properly on different browsers.
- - OS compatibility : web application on different OS.
- - Mobile browsing

- **4. Database Testing :**

- - check for data integrity.
- -check for data consistent.
- - correctness of data.

- **5. Interface Testing :**

- - should cover web server , application server , database server.
- -web server : all web requests are accepting
- -application server : request is sending correctly.
- -database server : correct result return on request.

- **6. Performance testing :**

- -check response time under different speeds.
- -check if site handles many simultaneous user request at same time.
- - how web application sustain under peak load

- **7. Security testing :**

- - check if unauthorized access to secure page.
- - check if accessing internal pages directly , required login then user should redirected to login page
- - check if sessions are got expired after predefined amount of time.

System Testing

Recovery
Testing

Load Testing

Security
Testing

Stress Testing

Performance
Testing

Usability
Testing

Compatibility
Testing

System testing

- The system is tested against functional / nonfunctional requirements such as accuracy , reliability , speed defined by the s/w system specification.
- IEEE defined system testing as ‘ testing conducted on complete integrated system to evaluate the system’s compliance with its specified requirement
- Final testing done on a s/w system before it is delivered to the customer.

- Advantage :
- 1. clearly specify how the application should behave.
- Test system from user point of view.

Recovery testing

- System is forced to fail in different ways to check whether the s/w recovers from failure without any data loss
- Check how fast and better the application can recover against any type of crash
- Example :
 1. while application is running , suddenly restart the computer and check for data integrity.
 2. while receiving data from n/w , unplug the connecting cable and plug back and analyze whether s/w continue receiving data from break point.
- Advantage :
 - Less complaint and reduce maintenance cost.

Security testing

- Attempt to verify that protection mechanisms built into a system will protect it from improper penetration.
- Tester play the role of the individuals who desire to penetrate system.
- It also aim at verifying below principle:
 - 1. confidentiality 2. integrity
 - 3. authentication 4. authorization
 - 5. availability 6. non repudiation

- Security testing techniques to protect s/w from following :
- 1.Spoofing identity:
 - stealing someone identity. Force application to use no authentication.
- 2.Tampering with data.
- 3.information disclosure.
- 4.DoS: s/w stop responding to valid request.

Performance Testing



Load Testing



Stress Testing



Soak Testing



Spike Testing

Performance testing

- Non functional testing technique performed to determine the system parameters in terms of responsiveness and stability under various workload
- Measure quality attribute such as reliability availability , resource usage.
- Factors that governs performance testing:
- **1. throughput** : capability of a product to handle multiple transaction in given period.
- **2. response time** : delay between the point of request and first response
- **3. tuning** : setting different values to the parameters of the product

Performance testing techniques

- **1.load testing:** determine whether the system is capable of handling anticipated number of users or not.
- **2.Stress testing :** determine how the system performs if current load goes beyond limits of its specified rqmts.
- **3. Soak testing :** determine system parameters under continuous expected load.
- test the system with expected or little more than the expected load for a long amount of time.
- **4.Spike testing :** to determine whether the system will be able to sustain the workload.
- testing is performed by increasing / decreasing the number of users suddenly and measuring performance.

Performance test process

- **1. Identify testing environment:** details of h/w , s/w n/w configurations..
- **2. Identify the performance acceptance criteria:**
 - Set performance bench mark which may not be present in specification.
 - goals and constraints for throughput , response time set.
- **3. Plan and design performance tests :** simulate variety of end users.
- **4. Configuring test environment:** prepare tools , resources.
- **5. Implement test design :** create performance test cases
- **6. Run the tests:** execute and monitor
- **7. Analyze , tune and retest**

Load testing

- Check whether system can perform well for specified load
- You feed it all that it can handle
- It involves simulating real life user load. It help in Identify maximum operating capacity of an application.
- If number of users are increased then how much CPU , memory will be consumed , what is n/w and bandwidth response time.
- **Example:**
 - 1. downloading a series of large files from the internet.
 - 2. server that can handle thousand of simultaneous connections , max out the capability.
 - 3. running multiple applications on a computer or server simultaneously.
 - 4. reading writing data to and from a hard disk continuously.

- **Advantages of load Testing:**
- 1.Expose memory overflow bug.
- 2. prevent s/w failures bcoz it can predict how system will react when it is given large load .
- 3. measure performance of internet infrastructure.

Stress testing

- IEEE , testing conducted to evaluate system beyond the limits of its specified rqmts.
- Test cases are designed to execute the system in such a way that abnormal condition arise.
- Limiting s/w to their bare minimum.
- Running s/w under less than ideal conditions—low memory , low disk space , slow CPU.
- Goal : ensure that s/w doesn't crash in condition of insufficient computational resources.
- **Example :**
 - 1. running resource intensive applications in a single computer at the same time
 - 2. infect system with viruses , trojans

- **Advantages of Stress Testing :**
- Indicate expected behavior of system when it reaches the extreme level.
- Help to determine expected operating conditions and the failure conditions.

Load Testing

- Increasing the load on the system
- Goal : expose defects related to buffer overflow ,
- Help to determine reliability
- Focus on response time
- Measure performance based on volume of users

Stress Testing

- Overload the existing resources with multiple program execution.
- Goal: analyze post crash report to define behavior of system after failure.
- Help to observe stability
- Focus on through put
- Measure breakpoint of system

Soak Testing

- **Why do Soak Testing?**
- A system may behave normally when used for 2 hours, but when the same system is used continuously for 10 hours or more than that then it may fail or behave abnormally/randomly/it may crash. To predict such failure Soak Testing is performed.
- memory utilization is monitored to detect **memory leak** (memory which is no longer needed is not released.)
- short period of test doesn't reveal Memory leak problems , but it will be reveal when test for a long amount of time.

Example :

- When a bank announces that it will be closing, its system is expected to handle a large number of transactions during the closing days of the bank. This event is rare and unexpected, but your system has to handle this unlikely situation anyway.

- **Advantages of Soak Testing:**
- It spots the deterioration in performance, which could happen under high continuous load
- It shows how sustainably the system runs over time.

Spike Testing

- Is a performance testing type used to test software applications with extreme increments and decrements in load and determine recovery time after a spike of user load.
- The goal of spike testing is to determine whether the system will fail or survive in case of dramatic changes in load.
- **Example Spike Testing Scenarios:**
- When an eCommerce store is launching special deals with great discounts such as on Black Friday.
- When the certain content of a site goes viral over the Internet.

- **Advantages of Spike Testing:**
- Spike testing helps in maintaining the system under the extreme load.
- Spike testing saves system or software application from crashing.
- It reduces the chances of failure for the system or software application.

Client server testing

- **1. components testing.**
- --test client and server individually.
- -- when server is tested , we may need a client simulator
- --while testing client , need a server simulator.
- -- test network by using client server simulator.
- **2. integration testing:**
- --after successful testing of server , clients , n/w , they are put together to form system and system test cases are executed.

Client server testing

- Testing on client side (GUI testing)
- -- **cross platform nature:** same GUI run transparently on different platform
- --**event driven nature:** user action are events that determine application behavior.
- Eg : application handles both mouse and keyboard as an alternate method of input

- Testing on server side (Application testing) :
- **Performance testing** : when number of client are communicating with a server at a time , system performance is tested for maximum throughput.
- **Concurrency testing** : multiple users may be accessing same records at a time , so concurrency test is required to understand response of a system.
- **Disaster recovery testing**: test for disaster recovery and business continuity.

Acceptance testing

- Formal testing conducted to determine whether s/w satisfies its acceptance criteria
- **Types of acceptance testing :**
- **1. user acceptance test :** validate by user
- **2. operational acceptance test:** ensure process/procedures are in place to allow system to be used. Eg: checking backup facilities , procedure for disaster recovery , security procedure.
- **3. contract acceptance testing:** system is tested against acceptance criteria as documented in a contract
- **4. compliance acceptance testing:** validate against regulation to ensure it meets governmental, legal and safety standards.

Acceptance criteria

- Definition : conditions that a s/w must satisfy to be accepted by a user.
- List of requirements that must be satisfied prior to the customer acceptance.
- Work at each stage of s/w development.
- Starting from proposal stage till the point where the system is formally accepted by user
- May be used as basis on which exit criteria for each phase and entry criteria of next phase may be defined.
- Example : Designs must fulfill acceptance criteria so coding phase can start.

Alpha testing

- Consider as internal acceptance testing in which End users test the s/w at developer's site.
- Assess the performance of s/w in the environment in which it is developed.
- Users report the errors to s/w developers.
- Advantage :
 - 1. Simulate real time user behavior and environment ?
 - 2. Early detection of errors ?
- Disadvantage :
 - 1. In depth functionality can not be tested
 - 2. sometimes developers and tester are dissatisfied with result of alpha testing

Beta testing

- Consider as external acceptance testing in which s/w sent out to a selected group of customers who use it in a real world environment.
- 'live testing' and conducted in an environment , which is not controlled by the developers.
- Testing is performed without any interference from the developers.
- Beta tests can be a good way to find compatibility and configuration bugs ??

- Advantage of beta testing :
 - 1. allow to test post launch infrastructure
 - 2. reduces s/w failure risk via validation.
 - 3. Improve product quality via customer validation
 - 4. Increase customer satisfaction
- Disadvantage of alpha testing:
 - 1. test management is an issue ?
 - 2. finding right beta users could be a challenge.

alpha testing

- Performed at developers site
- It involves white box black box testing
- Requires lab environment
- Ensure quality of product
- Test in controlled environment

Beta testing

- Performed at customer site
- It Involves black box testing
- Doesn't require lab environment
- Ensure product is ready for the real time users
- No controlled environment

Special Test



Regression Testing



GUI Testing

Regression testing

- Ensure that bug fixing doesn't introduce errors.
- Ensure that all code still work once code changes.
- Changes doesn't affect existing features.

Regression testing techniques

- Retest all
- Retest part of product
- Retest based on prioritized test cases.
- **--selecting test cases based on:**
 - - -frequent defect
 - - -verify core feature
 - - -function that has recent change
 - - -boundary value
 - - -failure test case(test to fail)
 - - -pass test case(test to pass)

GUI testing

- Important??
- How application handles keyboard , mouse events , and how menu bars , tool bars reacts to user input.
- 7 important traits common to good UI
- 1. follow standards and guidelines.
- -- when to use check box instead of option
- -- when to use warning , critical messages.

- **2. intuitive :**

- --functions be there when you expect them.
- --easily get from one function to other.
- --eg : move back button
- -- remove excessive functionality that make s/w complicate.

- **3. consistent:**

- --Users have a habit that if they do something a certain way in one program , another will do the same operation in same way.
- --Eg : in Notepad search → find or F3
- in word pad edit → find or Ctrl+F
- --Shortcut keys and menu selection. Eg : F1 for Help
- --Terminology and naming . Eg : find , search
- --Placement of buttons : position of OK Cancel button

- **4. Flexible :**

- -- users like to select what they want to do , and how they want to do.
- -- windows calc two views : 1. standard
2. scientific
- --state jumping : more ways to complete same task.
- -- data input / output : eg : to put text into a word pad → type , paste it , load it.

- **5. Comfortable :**

- ---appropriateness: look and feel. Eg: financial business application should not have loud color , sound.
- -- error handling : eg :warning before critical message
- -- performance : eg: show progress through status bar

- **6. correct :**
- --Test whether GUI does what its supposed to do..
- WYSIWYG :Eg: click on save button should saved document on disk.
- Language & spelling
- Bad media

- **7. Useful:**
- -- features actually valuable to s/w.