TESTING

Software Testing

- Testingistheprocessofexercisingorevaluatingasystemorsystemcomponentbymanualora utomatedmeanstoverifythatitsatisfiesspecifiedrequirements.

- Testingisaprocessofexecutingaprogramwiththeintentoffindinganerror.

- Software testing is a critical element of software quality assurance and represents the ultimate review of system specification, design and coding.

- Testing is the last chance to uncover the errors / defects in the software and facilitates delivery of quality system.

Role of Testing

- Primary Role of Testing
- Determine whether system meets specifications
- Determine whether system meets needs
- Secondary Role of Testing
- Instil confidence
- Provide insight into the software development process
- Continuously improve the testing process
- Why Test
- Developer not fallible
- Bugs in compilers, languages, DBs , Operating Systems
- Certain bugs easier to find in testing
- Don't want customers to find bugs
- Post release debugging is expensive
- Good test designing is challenging & rewarding

Testing Strategies, Levels/Phases

- **Testing Strategies**
  - Testingbeginsattheunitlevelandworks“outward”towardtheintegrationoftheentiresystem
  - DifferenttestingtechniquesareappropriateatdifferentpointsofS/Wdevelopmentcycle.
- **Testing Levels/Phases**

1. Unit / Component Testing

- Focuses on individual software units (programs) and group of related units

2. Integration Testing

- Focuses on combining units to evaluate the interaction among them

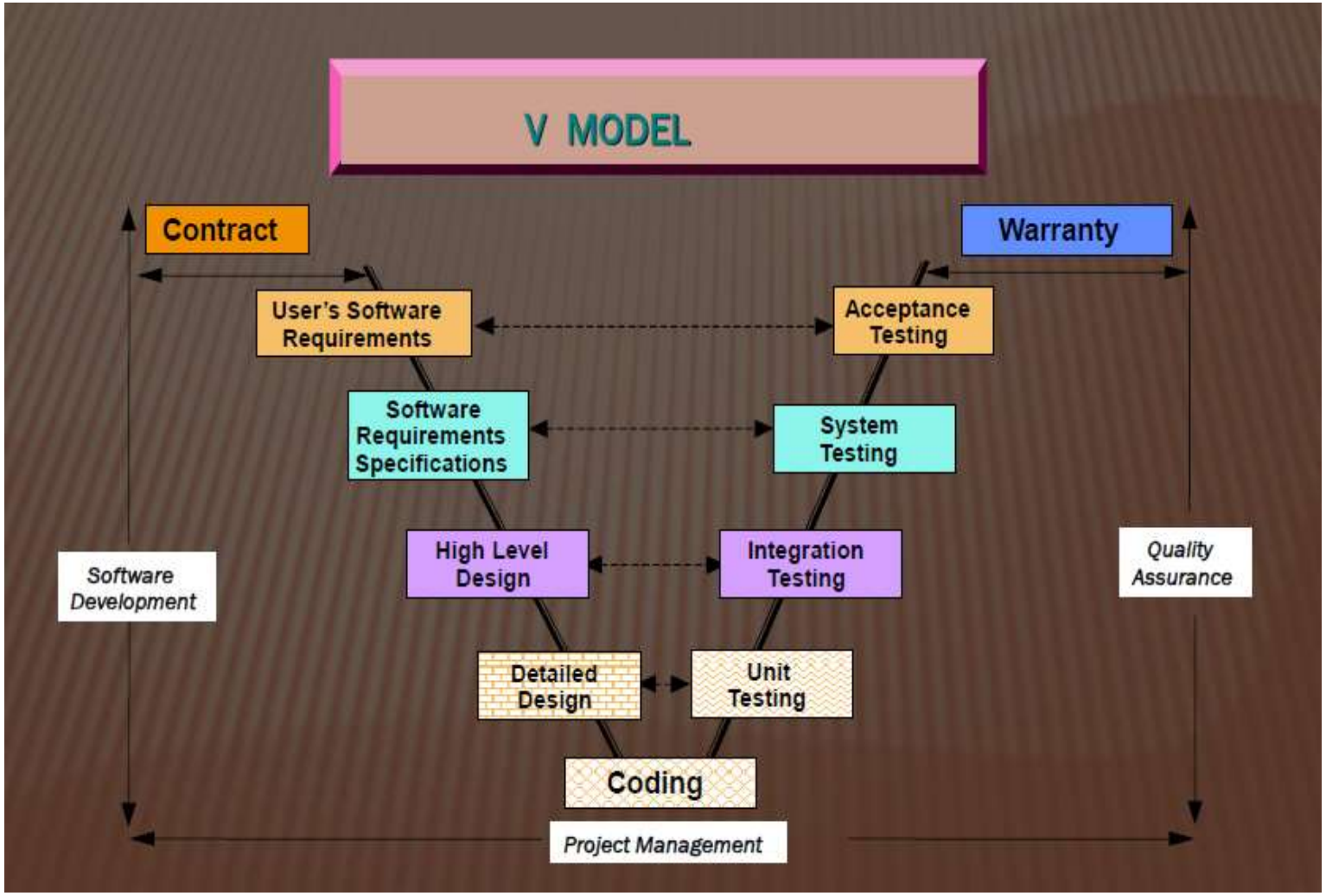
3. System Testing

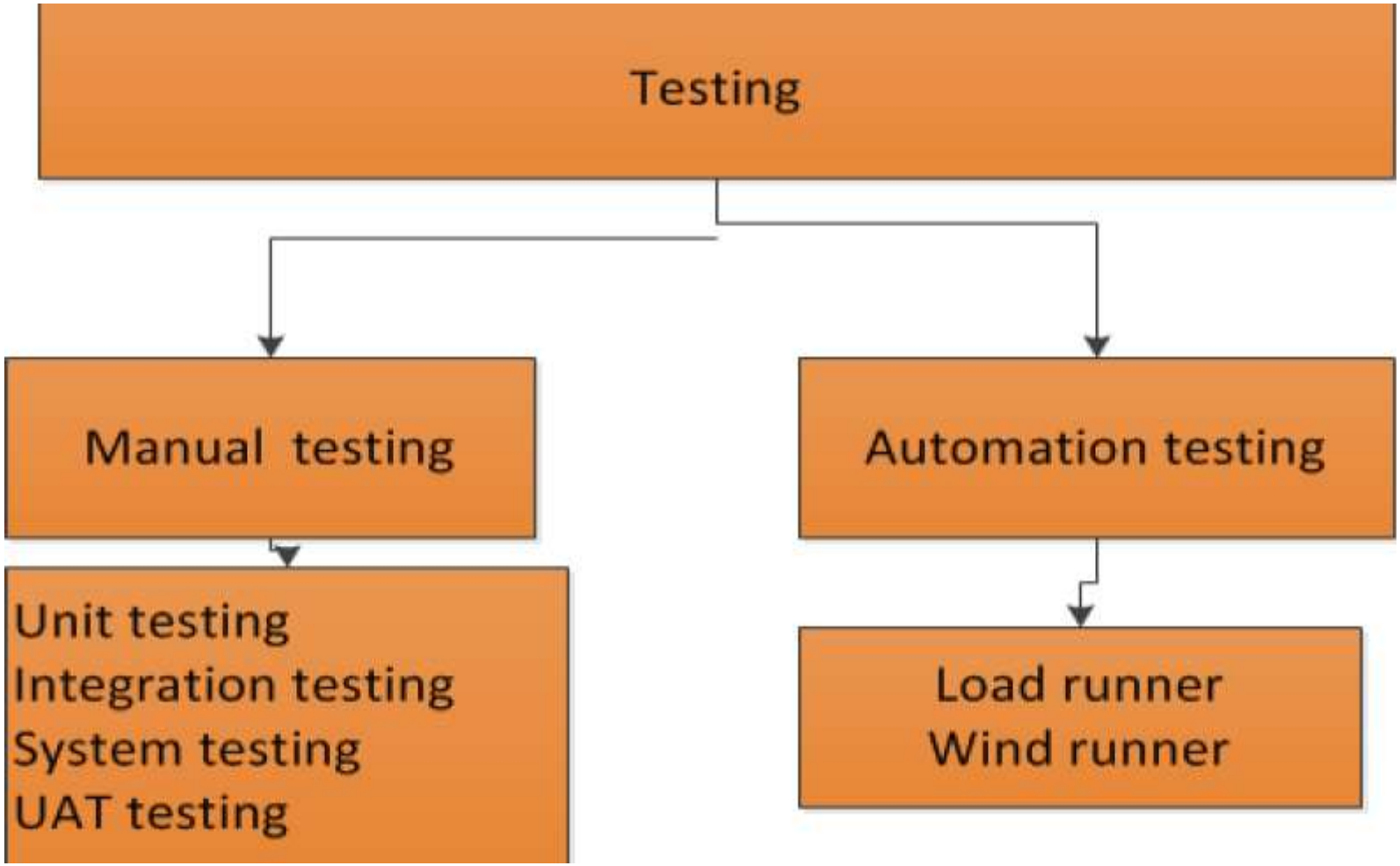
- Focuses on complete integrated system to evaluate compliance with specificities requirements (test characteristics that are present only when entire system is run)

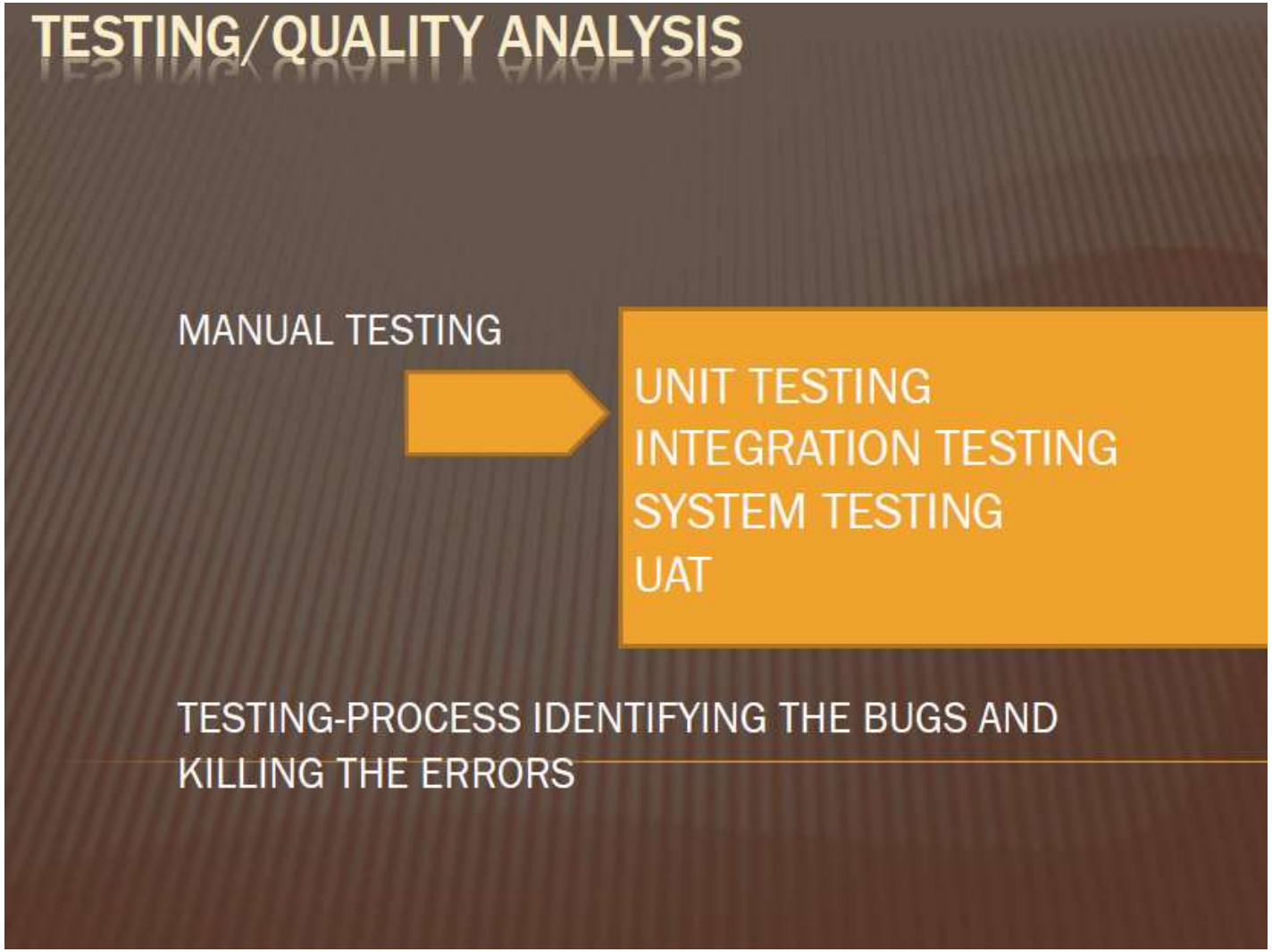
4. Acceptance Testing

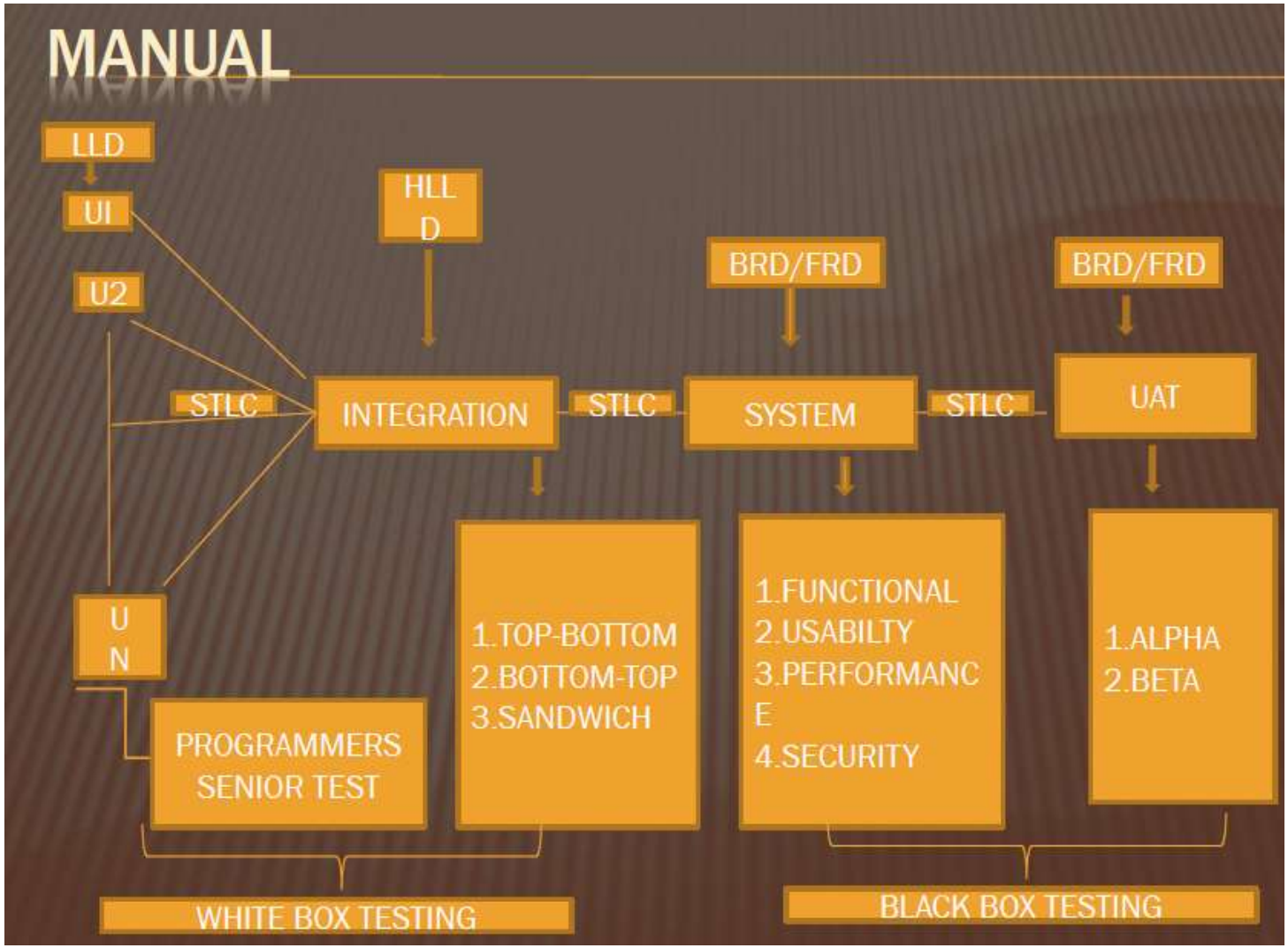
- Done from users perspective to evaluate fitness of use”

Relation of Development and Testing Phases




Testing
Manual testing
Automation testing
Unit testing
Integration testing
System testing
UAT testing
Load runner
Wind runner


TESTING/QUALITY ANALYSIS
MANUAL TESTING
UNIT TESTING
INTEGRATION TESTING
SYSTEM TESTING
UAT
TESTING-PROCESS IDENTIFYING THE BUGS AND KILLING THE ERRORS


MANUAL
LLD
UI
U2
U
N
HLL
D
STLC
INTEGRATION
STLC
SYSTEM
STLC
BRD/FRD
BRD/FRD
UAT
1.TOP-BOTTOM
2.BOTTOM-TOP
3.SANDWICH
1.FUNCTIONAL
2.USABILTY
3.PERFORMANC
E
4.SECURITY
1.ALPHA
2.BETA
PROGRAMMERS
SENIOR TEST
WHITE BOX TESTING
BLACK BOX TESTING

Software Test Life Cycle

1. TEST PLAN
2. TEST DESIGN
3. TEST EXECUTION
4. BUG REPORT
5. RESULT ANALYSIS
6. TEST SUMMARY

Test Plan

- TEST PLAN IS PREPARED BY SENIOR TEST ENGINEER

- NOT BY BA

1. INTRODUCTION
2. TEST OBJECTIVE
3. TESTING STRATEGY
4. ITEMS TO BE TESTED

EXAMPLE: FOR REQUIREMENTS

5. ITEM TO BE NOT TESTED

6. Resource Allocation Template

S/N	Item Name	Resource Collection
1	**TEST CASES OF XYZ MODULE**	**2**
2	**TEST CASES FOR LOGIN PAGE**	**4**
3	**TEST CASES FOR HOME PAGE**	**3**

7. Resource Scheduling

S/N	Task Name	Resource Name	Resource Scheduling
1	**LMS**	**Calendar**	**3**

8. Hardware/Software Requirements

8.1 Software Requirements

S/N	Software Number	Number of License	Reason
1	**QTP**	**33333**	**FOR FUNCTIONAL TESTING**

8.2 Hardware Requirements

S/N	Hardware Number	Number of License	Reason
1	**SOURCE CODE**	**2**	**FOR FUNCTIONAL TESTING**

Exit or Resume Criteria

- SITUATION IT IS USED

- IT WILL BE EXISTED FOR TESTING

Test Design

- TEST SCENARIO
- TEST CASE
- 1.TEST SCENARIO

Scenario Number	Scenario Name	Steps	Alternative Steps	Expected Result
1	**Cash with**	**Visit ATM Insert Card**		

Test Case

- Test case document is prepared

Test Execution

Requirement Number	Requirement Name	Steps	Expected Result	Actual Result	Severity	Priority
Req. 001	Login	Useyk	as	Pass	Fatal	P1

Result Analysis

Test Name	Test Case Number	Severity	Priority	Pass	Fail
Req. 001	Login	Fatal	P1	Pass	
		High	P2		
		Medium	P3		
		Low			

TEST CASE SUMMARY

- Number of test cases tested----100
- Number of test cases executed---20
- Number of test cases passed---50
- Number of test cases failed---29
- Number of test cases not tested ----29
- % of test cases tested/pass
- %of test cases failed/

BUG LIFE CYCLE

- **Description of Various Stages:**

1. New: When the bug is posted for the first time, its state will be "NEW". This means that the bug is not yet approved.

2. Open: After a tester has posted a bug, the lead of the tester approves that the bug is genuine and he changes the state as "OPEN".

3. Assign: Once the lead changes the state as "OPEN", he assigns the bug to corresponding developer or developer team. The state of the bug now is changed to "ASSIGN".

BUG LIFE CYCLE

4. Test: Once the developer fixes the bug, he has to assign the bug to the testing team for next round of testing. Before he releases the software with bug fixed, he changes the state of bug to "TEST". It specifies that the bug has been fixed and is released to testing team.

5. Deferred: The bug, changed to deferred state means the bug is expected to be fixed in next releases. The reasons for changing the bug to this state have many factors. Some of them are priority of the bug may be low, lack of time for the release or the bug may not have major effect on the software.

BUG Report

- Rejected: If the developer feels that the bug is not genuine, he rejects the bug. Then the state of the bug is changed to "REJECTED".

- Duplicate: If the bug is repeated twice or the two bugs mention the same concept of the bug, then one bug status is changed to "DUPLICATE".

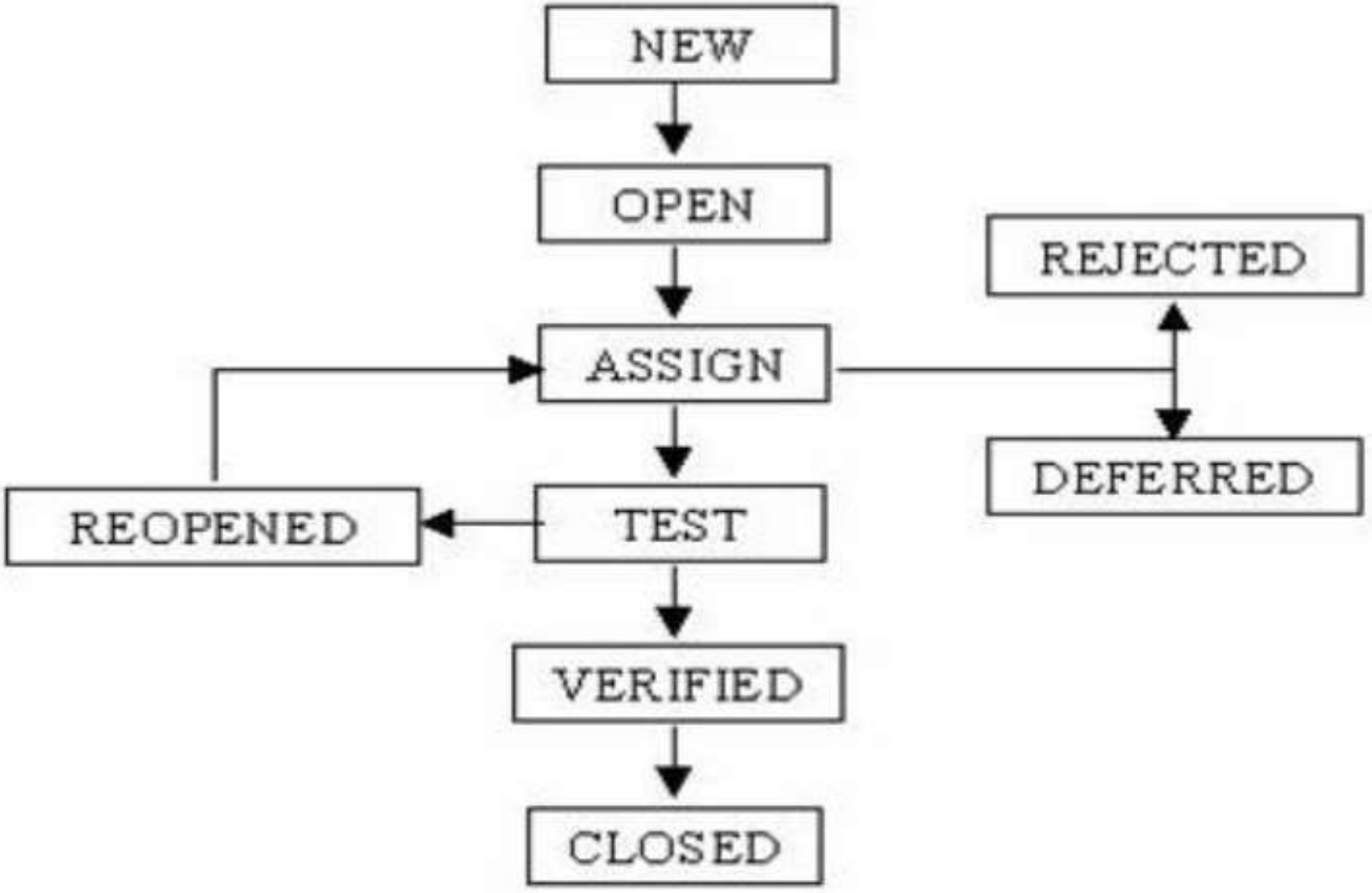
- Verified: Once the bug is fixed and the status is changed to "TEST", the tester tests the bug. If the bug is not present in the software, he approves that the bug is fixed and changes the status to "VERIFIED".

- Reopened: If the bug still exists even after the bug is fixed by the developer, the tester changes the status to "REOPENED". The bug traverses the life cycle once again.

BUG Report

- Closed: Once the bug is fixed, it is tested by the tester. If the tester feels that the bug no longer exists in the software, he changes the status of the bug to "CLOSED". This state means that the bug is fixed, tested and approved.

BUG Life Cycle

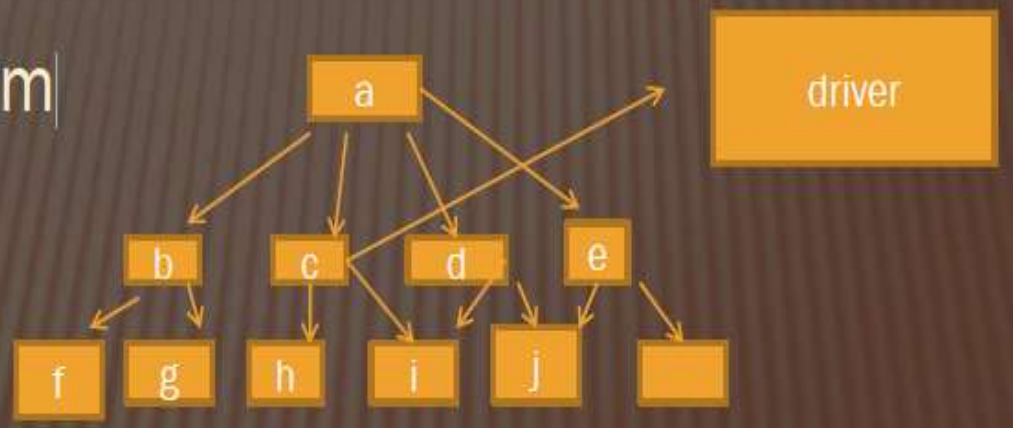


UNIT TESTING

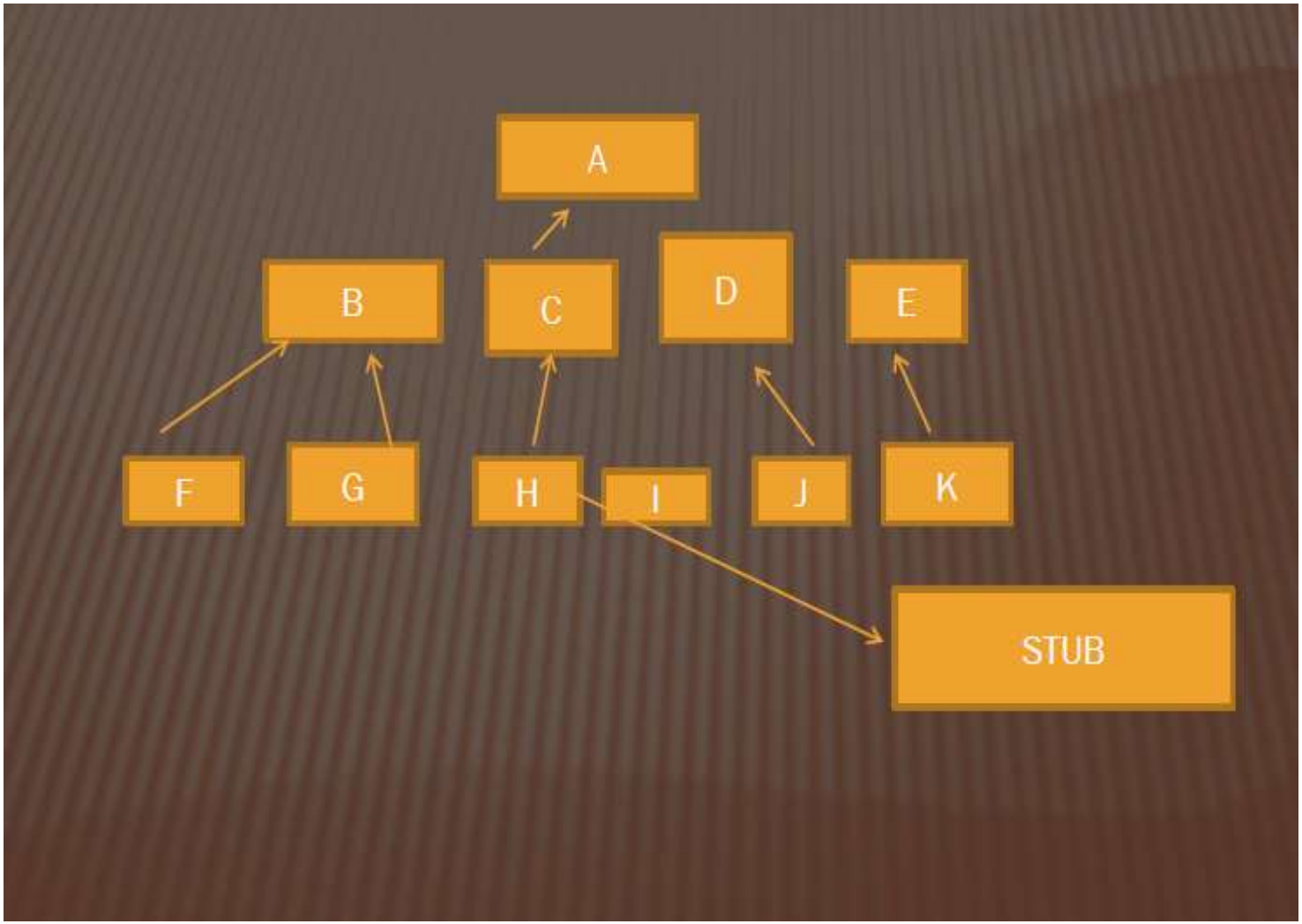
- Unit testing is done unit wise
- Code base testing

Integration Testing

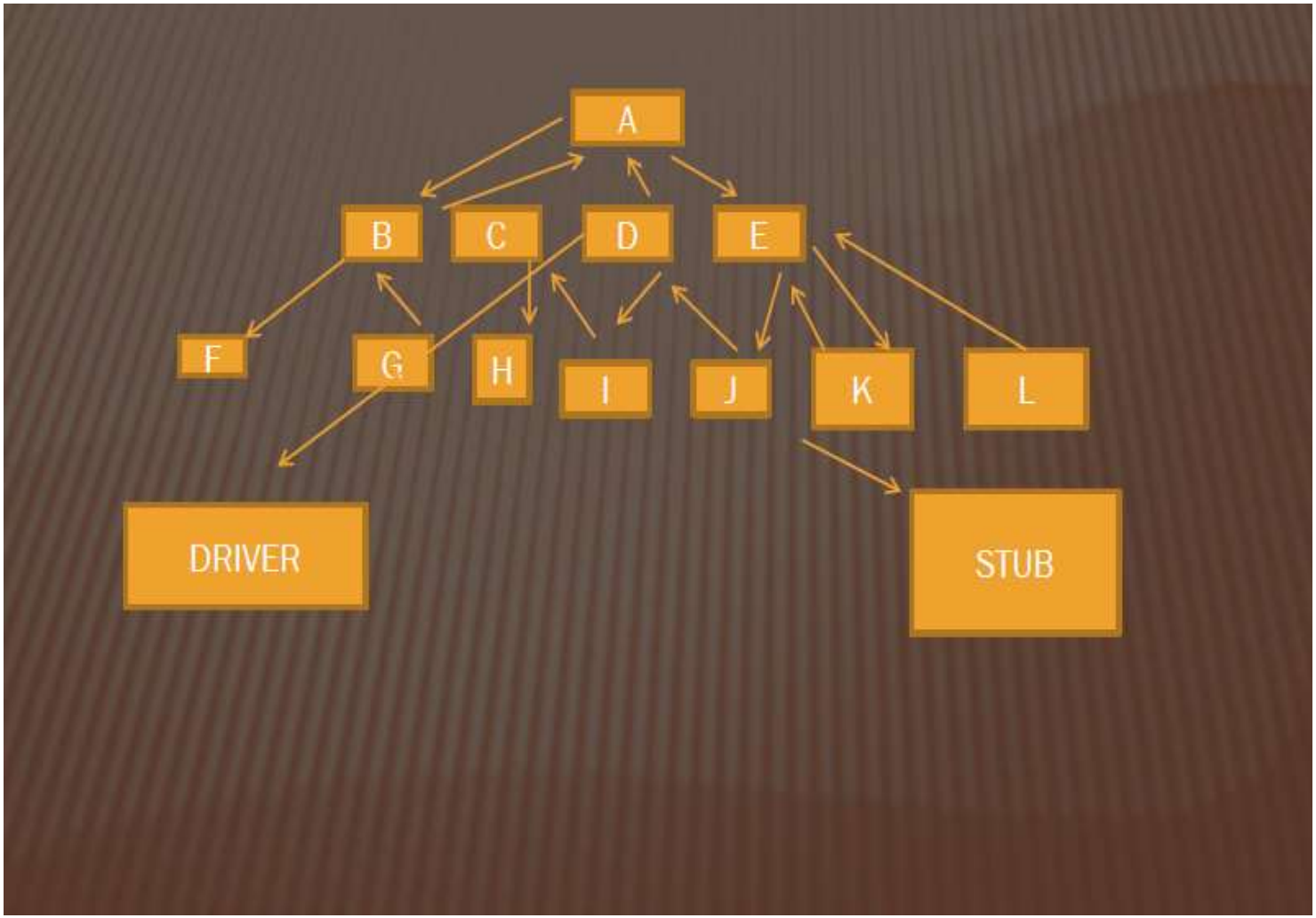
- Top-bottom
- Bottom-top
- Sandwich
- Top-bottom



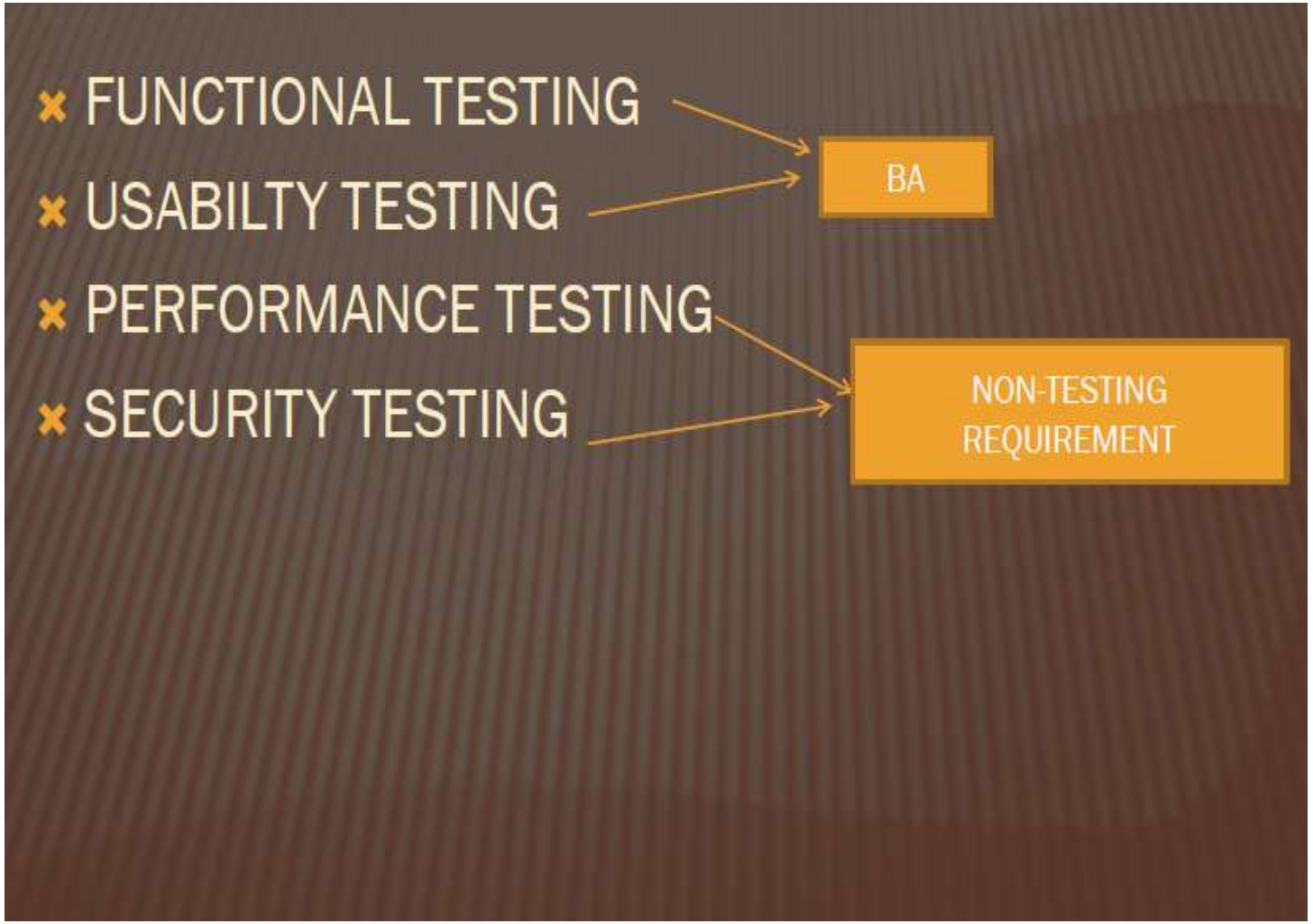
Bottom-Top



Sandwich



System Testing



Usability Testing

- GUI TESTING
- LOOK & FEEL
- COLOR
- FRONT SIZE
- SIZE

User Acceptance Testing (UAT)

ALPHA TESTING

BETA TESTING

Alpha Testing

Testing is done in our company in front of client for internal feedback

Beta Testing

- Testing is don't by client at their company for final feedback from client
- As a business analyst we are giving training to end user.
- Finally user manuals to clients