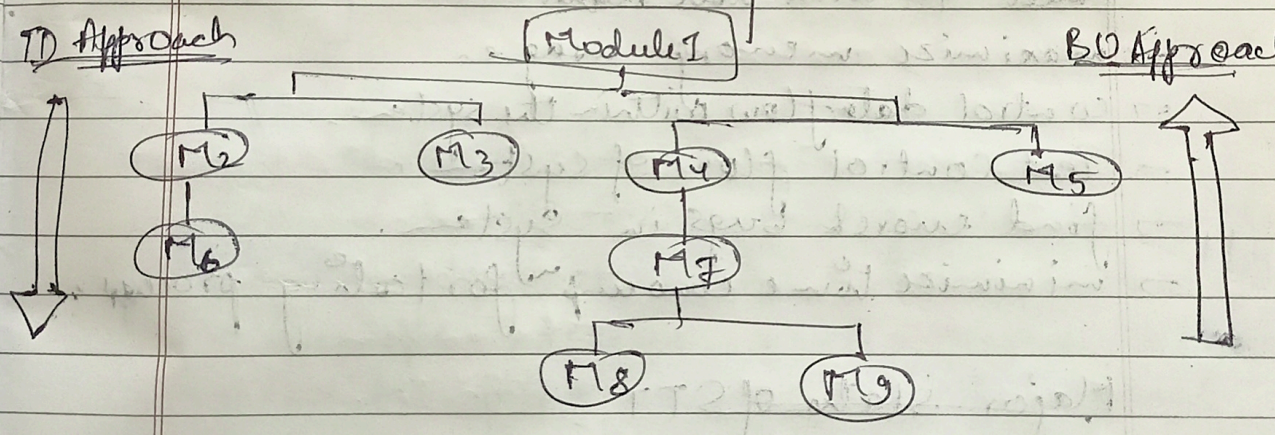
MID SEM-2

Unit-3

Date
Page

Integra"

Top-Down	Bottom-Up
→ system integra" begins with top level modules	→ ———
→ Higher level modules tested first, then lower level are tested, & then modules are integrated accingly.	→ lowest ——— lower ——— higher ———
→ Stub works as temporary replacement for submodule.	→ Driver ——— main module.
→ Main module is designed 1st, then submodules are called from it.	→ Diff modules are created 1st then these modules are integrated with the main".
→ Implemented on Structured Oriented lang.	→ ——— Object ———
→ Complexity Simple	→ high complexity.
→ Works on big to small components	→ ——— small to big ———



(2) Bi-direc"al Integra" Testing

→ also k/a Sandwich IT / Hybrid IT
→ Combina" of TD & BU approaches.
→ Stubs & Drivers are used.

Adv

Disadv

Date
Page

Adv	Disadv
→ useful for very large prg's.	→ requires very high cost & both TD & BU components are req
→ overcomes the shortcoming of TD & BU approaches that in TD, top-level module is to be coded & unit tested 1st	→ can't be used for smaller systems with huge interdependencies b/w diff. modules.
→ Parallel tests can be performed in top & bottom layers.	

(3) System Integration Testing

→ carried out to perform the overall testing of a complete system that consists of many integrated components.

→ The system on which SIT is performed may have diff. h/w parts, diff. s/w parts or both h/w & s/w.

Objectives of SIT:

→ meet s/w with user requirements.
→ maximize memory usage.
→ control data flow within the system.
→ Test control flow of system.
→ find errors & bugs in system.
→ minimize time consumption for testing process...

Major states of STT

→ Data State within Integration layer
→ _____ Database _____
→ _____ Appln _____.

④ Scenario Testing

→ uses scenarios i.e. speculative stories to test the system.
→ test cases are single steps whereas scenarios cover a no. of steps.
→ The tester assumes themselves to be the end users & find the real world scenarios which can be carried out of the s/w by the end user.

Methods in Scenario Testing

→ System Scenarios: that covers various components of syst.
→ Use-Case/Role Based Scenarios: focus on how the system is used by a user with diff. roles & env.

Risks:-

→ complex involving many features
→ not designed for coverage of the probs.
→ heavily documented & time consuming.

⑤ Defect Bash

→ is an ad-hoc testing.
→ performed by diff. roles in an organiza- to test the product together at the same time such as all the developers, testers, prog. managers, designers, documenta- folks & even marketing people.
→ Its not based on written test cases.
→ What reqd. to be tested is up to individual's decision & creativity
→ Main goal is to find out as many defects as possible so that we can fix all the issues with the business pt. of view.

(6)

Functional	Non-functional
→ verifies opera's & actions of an app".	→ Verifies behaviour of an app".
→ based on requirements of customer	→ based on expecta's of ———
→ enhance behaviours of sys	→ improve performance of sys
→ easy to execute manually	→ hard ———
→ describes what the product does	→ describes how ———
→ based on business requirem	→ ——— performance ———
Eg:- → Unit Testing	Eg:- → Performance Testing
→ Smoke "	→ Load "
→ Integra" "	→ Stress "
→ Regression "	→ Scalability "

(7) Design Verifica"

Its a process that examines & shows proof to certify that the outcome of a specified s/w product satisfies the i/p. requirements.

Design Verifica" process → Identifica" & Prepara".
→ Planning
→ Developing
→ Execution
→ Reports.

(8) Deployment Testing

→ Testing the app" before, during & after its deployment to avoid any product" issue is k/a Deployment Testing.
→ Done after the s/w is completely developed.
→ Also k/a Installa" / Implementa" Testing.

Date
Page

Objectives :-
→ To identify end user prob. when they will initially try to use / deploy the s/w.
→ To measure scalability & stability of sys.
→ To measure performance rates like data collec^n rate etc.
→ To identify issues proactively that would occur in the deployment site.

⑨ Beta-Testing
→ performed by real users of the s/w appⁿ in a real env.
→ One of the types of User Acceptance Testing
→ A pre-version of s/w is released to a ltd. no. of end-users of the product to obtain feedback on the product quality.
→ Its the last test before shipping a product to the customers.

Types ⓐ Traditional BT :- Prod. released to target market.
ⓐ Public BT :- Prod. released publicly to world.
ⓐ Technical BT :- ——— to a grp. of employees of the org.
ⓐ Focused B.T :- Prod. released to market for collecting feedback on specific features of prog.
ⓐ Post-Release BT :- released to market to make improvements for future release of the prod.

Adv.
→ Reduces prod. failure risks via customer validat^n
→ Improves product quality via customer feedback.
→ Cost effective
→ Creates goodwill with customers

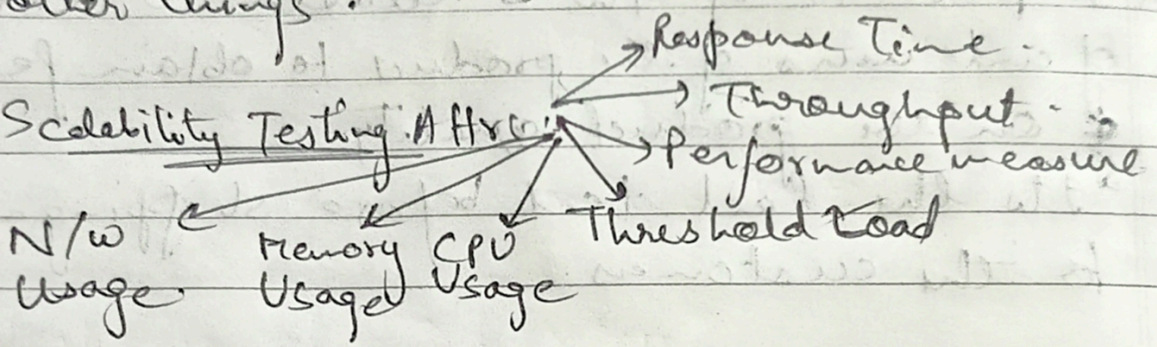
Disadv.
→ complex to follow errors as testing env. varies from user to user.
→ chance of having duplica^n of error.
→ Time consuming
→ Dvelopment & Testing team don't have control over real time test env.
→ The testing users should have enough knowledge about working of entire prod.

(10) Scalability Testing

→ non-functional testing

→ Defined as the ability of a product to perform effectively correctly when changes are made in the size/vol. of the system to meet a growing demand.

→ It ensures that a s/w product can manage the scheduled rise in user traffic, data vol., & many other things.



Adv:	Disadv:
→ provides more accessibility to prod.	→ fails to find final errors.
→ provides customer satisfaction.	→ costlier
→ fixes the issues in earlier stage.	→ Time consuming
	→ Testers should have high level of testing skills.

(11) Reliability Testing

→ Defined as a type of s/w testing that determines whether the s/w can perform failure free operation for a specific period of time in a specific env.

→ It ensures that the product is fault-free & is reliable for its intended purpose.

Types → Feature :- each f^n in the s/w should be executed at least once.
→ Regression :- new f^nality does not create new bugs.
→ Load :- performance of s/w under max workload.

Measurement of reliability Testing :- done in terms of MTBF
→ Mean time b/w failures (MTBF)
→ Mean Time To failure (MTTF)
→ Mean Time To Repair (MTTR)

MTBF = MTTF + MTTR

(12) Stress Testing

→ determines robustness & error handling of s/w by testing beyond the limits of normal operation.
→ Imp. for critical s/ws but is used for all s/ws.
→ It verifies stability & reliability of system.
→ It analyses the behaviour of system under extreme condi^ns.
→ It ensures that the system does not crash under crunch situations.
→ Ala Endurance / Torture Testing.

Pros:
- makes the system work in normal as well as abnormal condi^ns.
- ensures system failure doesn't cause security issues.
- ensures system recovers quickly.

Cons:
- Time taking & complex
- Needs extend resource for testing.
- costly.
- Good scripting knowledge is reqd. for implementing script test cases.

(13) Acceptance Testing

→ It is a method where a system is tested for acceptability.

→ Aim is to evaluate the compliance of the system with the business requirements & assess whether it is acceptable for delivery or not.

Acceptance Criteria → Functional Correctness & Completeness
- Data Integrity
- Data Conversion
- Usability
- Performance
- Timeliness
- Confidentiality & Availability
- Installability & Upgradability
- Scalability
- Documentation

Test Case Selection & Execution

→ Test cases should be selected in a way that can cover most of the acceptance testing scenarios.

→ Test case execution is done by appropriate i/p values. The testing team collects i/p values from end user, then all test cases are executed by both tester & end user to make sure s/w is working correctly in the actual scenario.

Adv.	Disadv
→ Client satisfaction.	→ Customers are not willing to do it
→ Automated test execution.	→ Users should have basic knowledge
→ easier for user to describe requirements	→ Time Consuming
→ It covers only Black-Box testing	→ Development team does not participate in testing process