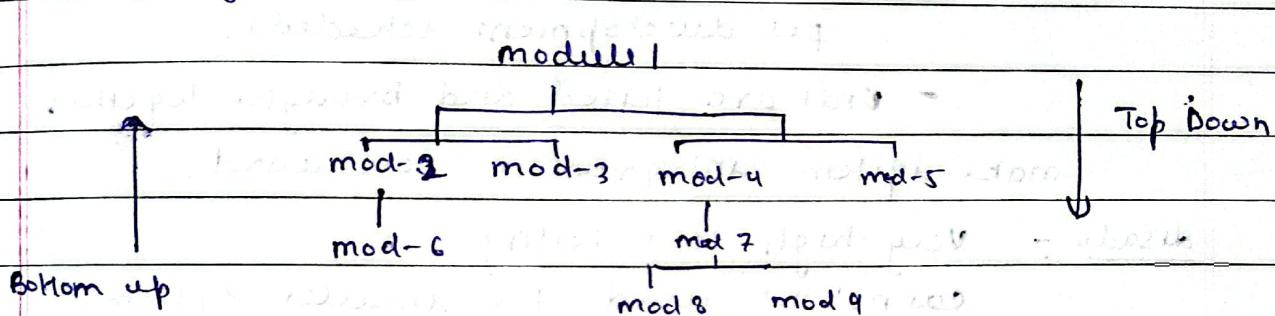


Unit-3* Top down and bottom up integration

Top down - also called incremental integration testing.

Higher level modules are tested then lower then these modules go for integration.

* Bottom up - In this lower level are tested first then higher level. lower module are called submodules and higher are called main modules.

Top down→ Top to bottom

→ Stubs are used for simulate the submodules if invoked submodules not developed.

Momentary replacement

- beneficial if defect occurs in top of program.

- implemented on structure procedure oriented programming langt.

- complexity is simple.

- big to small component.

- stub must be produced.

Bottom upBottom to top

→ Drivers are used for simulate the main module if main module not developed.

- beneficial if defect is at bottom

- OOP language.

- more complex

- small to big

- drivers must be produced.

* Bidirectional integration testing -

- It is a kind of integration testing that combine top down and bottom up testing.
- Drives are used to provide upstream connectivity.
- Units provide downstream connectivity.
- Also called sandwich integration.

Advantages - useful for large projects.

- Both top down & bottom up start at a time per development schedule.
- Units are tested and brought together to make system integration downward.

disadv - Very high cost testing

cannot be used for smaller system.

* System integration testing - type of testing carried out

in a integrated hardware, software to verify the complete system.

- testing of whole system
- Objective is to ensure all software module dependencies are functioning properly and data integrity is preserved.
- Control data flow
- To find error
- minimise memory usage.

* Scenario testing - It uses scenario i.e. speculative.

stories to help the tester work through a complicated problem or test system.

scenario testing is performed to ensure that end to end functioning of software and all process flow working properly.

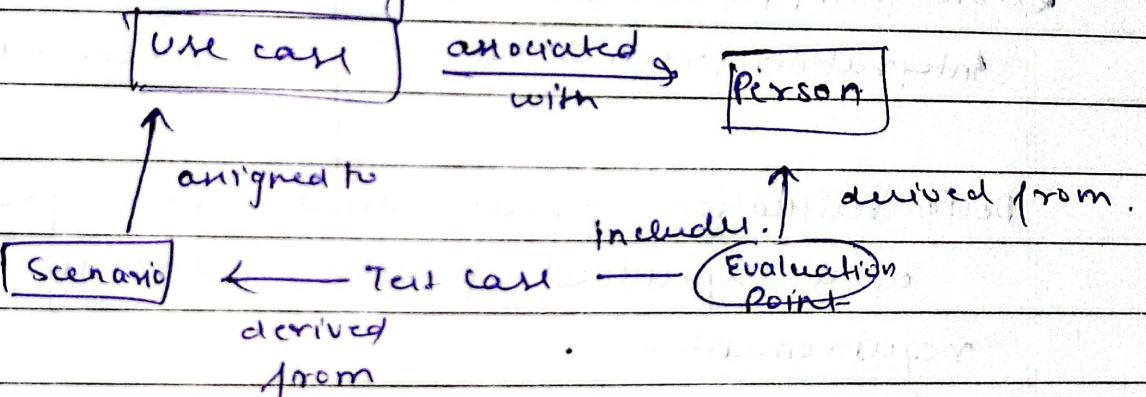
In this tester assume themselves as end user and find real world scenarios which can be carried out on software.

Characteristic - Story

motivating
credible
complex

easy to evaluate.

Properly -



- Disadv - complex , time consuming

Defect bash - ad hoc testing

- people performing different role test the system at same time.
- not based on written cases
- All activity are planned activities, except for ~~tell~~ what to be tested.
- to find out all type of defects

functional	Non-functional
→ It verified the operation and action	It verified the behaviour based on expectation.
→ Based on requirement of customer	
- enhance behaviour of application to improve performance.	
- easy to execute manually	hard.
- It test what product does	describe how product does.
- based on business requirement	based on performance requirement
Ex-unit testing, Smoke testing, Integration, regression	Performance, load testing, stress, scalability,

Design verification - process that show proof that outcome of a specified product satisfy input requirement.

Design verification process - Identification And preparation
Planning
Developing
Execution
Reports

* Deployment testing -

- Testing application before, during & after its deployment to avoid any production issue
- done after development of software
- Also called installation / implementation testing.

object - identify - user problem when they use software
- measure scalability and stability
- measure performance
- identify issue that would occur in deployment.

- Beta testing - It's performed by real user in real env.
- kind of user acceptance testing.
 - A beta version of software whose feedback is needed
 - minimization of product failure
 - last test before shipping product to customer.

Characteristic → Performed by client

- Reliability, security & robustness are tested
- Black box testing
- Do not require lab

Type - Traditional BT (Product distributed to target market)

Public BT (Product is released to world by online)

Technical BT (Product released to employee)

Focused BT (Product released to market to check specific feature)

Post-released BT (Product is released and data for improvement is collected)

Adv - Reduce product failure

disadv - complex to follow error

improve product

- time consuming

create goodwill in customers

- knowledgeable user

cost effective

needed.

* Scalability testing - Type of non-functional testing

- Performance of app, network or process is tested in term of capability to scale up or down the no. of user load.

- basically ability of network, system, app, process to function correctly when API or volume user change.

Objective → to determine user limit for product

- end user experience under load
- service degradation and robustness

attribute - Response time

throughput

Performance with change in user.

threshold load

CPU usage.

memory usage.

network usage.

adv - high level of knowledge

Adv - Provide customer satisfaction

find performance issues.

- unexpected result

more accessibility to product

after launching

- time consuming.

* Reliability testing - it refers to ability of software to function in given environmental condition.

It determine whether a software can perform failure-free operation.

Objective - to find perpetual structure of repeating failure.

- to find no. of failure occurring in specific period of time.

Failure mode - main cause of failure.

Type - feature - each function is tested.

Regression - testing whenever new function is added.

load testing - determine whether application can handle required load.

measurement of Reliability testing -

- Mean time between failure (MTBF) -

- Mean Time to failure (MTTF)

- Mean time to Recover (MTTR)

$$\text{MTBF} = (\text{MTTF} + \text{MTTR})$$

Stress testing - technique that determines the robustness of software by testing beyond the limits of normal operation.

- verify stability and reliability
- test beyond the normal operating point and analyse how system work under extreme condition.
- also called endurance testing / torture testing.
- verifies unexpected failure do not cause security issue.

Advantages

- behaviour of system after failure and enclose it recover.
- failure do not cause security issue.
- work well in normal / abnormal condn.

Disadvantages

- good scripting knowledge for implementing test case
- if proper script not implemented then there will be failure.
- cost more.
- Acceptance testing - it is last phase of software testing performed under system testing.

unit testing → Integration testing → System testing → Acceptance testing

• it is formal testing according user needs, requirement and business process conducted to determine whether system satisfy the acceptance criteria or not. and enable user, customer & authorised entities to determine whether to accept system or not.

Acceptance critiquing - functional correctness

- Usability and Standard
- Performance
- Data integrity
- Scalability
- Usability
- confidentiality and availability
- installability, upgradability.

Adv - Bring confidence and satisfaction to client

- It is easier for user to decide requirement.
- covers only black-box testing

disadv - development team do not participate

- user should have knowledge.
- feedback take long time

* Test case selection and execution

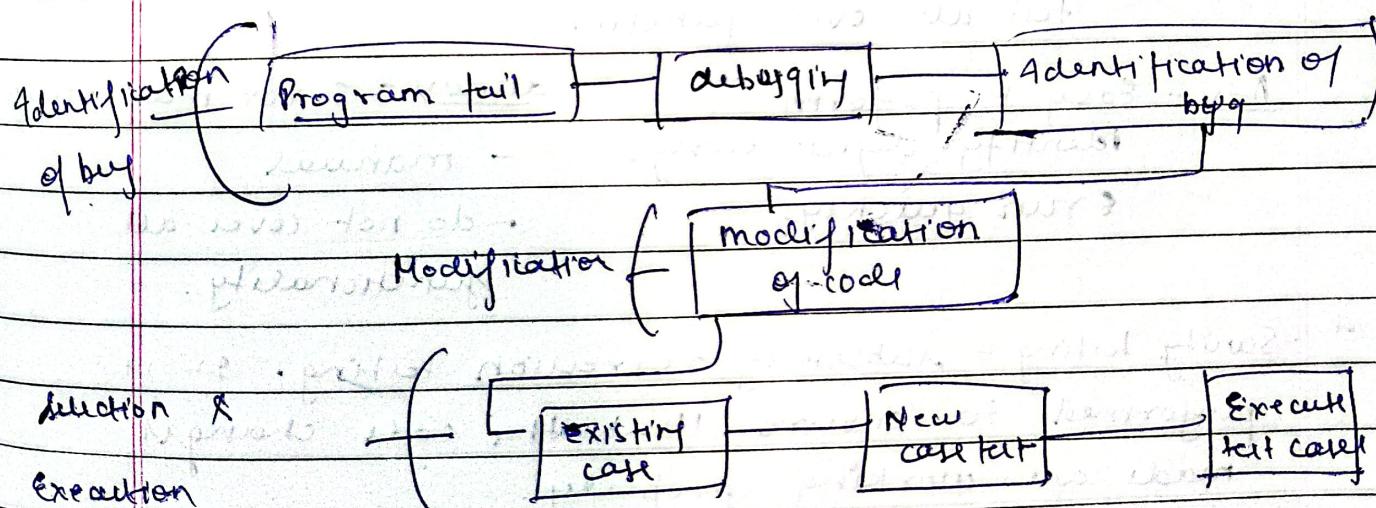
- test case should selected that cover most of acceptance testing scenarios
- test case execution is done by process of executing the code and comparing the expected and actual results.

* Regression testing - Process of testing modified parts of the code and part that might get affected due to modification to ensure that no new error have been introduced.

When to do regression testing - when new functionality is added
 - when defect are identified
 - when code is modified.

* Process of regression testing -

- Whenever we make changes to source code and then our program fails on previous case.
- After failure, source code is debugged.
- After identification of bug appropriate changes are made.
- Then appropriate test cases are selected to test again.
- New cases can also be used.



Techniques for test case

- select all test case
- select case randomly
- select higher priority test case.
- Select modification traversing test cases
 (test cases that cover modified part of code)

Tools for regression testing - selenium, QTP (Quick Test Professional)

Bots..... Pops.....

RPT (Rational Functional Tester), Winrunner, Silktest

Adv - ensure no new bug introduced

- most test cases are re-selected from existing test cases.

- maintain quality of code.

disadv - time consuming

required after very small change.

+ Smoke testing - type of testing that guarantees an application basic and critical feature are working fine before doing exhaustive testing or rigour testing.

It is mini-quick regression test of major function.

Characteristic - it is documented.

can be stable or unstable

type of regression testing.

Aim - detect early defects

check system stability

test all over function.

Adv - Easy to perform

Identify defect early

Run quickly

disadv - Error may occur

- manual

- do not cover all

functionality.

* Sanity testing - subset of regression testing. It is performed to ensure that the code changes made are working properly.

feature - Not documented

Unscripted

Narrow and deep (limited function tested deeply)

Performed by tester

- quickly identify defect
- less time required
- Not so expensive

- drawbacks - cover few function
- unanticipated.
- do not cover all function.

Adhoc /Testing/

slicing - It is a technique used ⁱⁿ software testing which takes a slice or a group of program statement for testing particular test condition or call that may effect a value at particular point.

Types -

- 1) static slicing
- 2) dynamic slicing.

static slicing - contain all statement that may affect value of variable at any point

- static slices are generally larger.
- consider every possible execution of program.

Dynamic slicing - contain all statement that actually affect value of variable at any point

- Dynamic slices are generally smaller.
- consider only particular execution of program.

* Execution trace -

- Test case execution log presents the output of test case.
- To analyze failed test case, the test case execution trace are captured.
- It presents which statements were touched during test case execution.