Unit-3 - STQA

Software Testing Methodologies.

(1) White Box Testing.

* It test internal coding and infrastructure of a Software focus on checking of predefined input against expected and desired output.
* internal working structure knowledge necessary
* Good for algorithm testing.
* tester and developer
* time consuming

(2) Black Box testing.

* It examine the functionality of Software without peering into internal structure or coding.
* No knowledge of structure needed
* end user, tester or developer
* trial and error
* least time consuming.

(3) Grey Box testing

* combination of white Box testing and white Box testing.
* Testing is done on the basis of high level database diagrams & data flow diagram.

Test cases design technique.

(1) White Box technique.

(1) Static Techniques.

* It is type of testing in which software is tested without code Execution.
* Early Stage of development.
* verification stage
* less expensive.

(2) Dynamic Technique.

* code is executed during testing.
* system functionality, Memory/CPU uses, and overall system performance.
* validation stage
* More expensive.
* verify functional & non functional requirement.

Static technique

① Informal review.

① the work product is given to domain Expert and the feedback/comments are reviewed by owner/author

* unplanned meeting, time meeting,
* B/w 2-3 person.

2) Walkthrough.

* informal and by author
* review and discuss the technical aspects & software development process.
* find defects. (not correct the error).

3) Technical Review

* less formal review
* By trained Moderator or technical Expert.
* ensure the technical concept is correct, and understand to all participants.

4) Inspection.

* highly formal
  1. Author - programmer
  2. Moderator - run the inspection process
  3. Inspector - code reviewer
  4. Scribe - notes taker.

Stages. (1) preparation before inspection
(2) Enlisting multiple diverse review
(3) Assigning role to participants.
(4) Going sequentially through the code in a structured manner.

Structural Testing.

→ It is used to test the internal design of the software or structure of the coding for the particular software.
→ White box testing & carryout at various levels
→ opp to behavioral testing
→. developer + tester.

Types:

(1) Control flow testing.
→ Based on how the control is executed during the code.

(2) Data flow testing
→ Uses control flow graph
→ check where code lead to an alteration in the data.

(3) Slice based testing:
→ maintaine software, debugging
→ divide whole program to slices and test

(4) Mutation testing:
→ development of new test as mutant of old test.

Advantages of Structural testing

1) Complete checkup
2) Smooth execution from an early stage
3) Dead code are removed easily

4) Automated process
5) Easy coding & implementation.

Disadvantages:
1) Complete knowledge required
2) Complicated tools
3) time and energy consuming

* Technique to carry out structural testing

1) Statement coverage Testing.
→ examining all the statement by calling out them in practice.

Statement coverage = $\frac{\text{no. of executed statement}}{\text{total statement}} \times 100$

→ So, It cover dead cover, Unused code, and branches.

Advantages (1) check flow of program statement
(2) conform quality code by executing program at atleast once.

Disadvantages (1) not sufficient
(2) missing statement are not covered
(3) costly.

2). Path Coverage Testing.

→ It consider all possible path present in block code

→ Path coverage = $\frac{\text{Total path exercised}}{\text{Total no. of path.}} \times 100$

→ If loop present than, there are infinite path.

= sequential testing in which each individual line accessed

$\frac{1}{4} \times 100 = 25\%$

Advantage. (1) reducing redundant test

(2) focus on program logic

Disadvantages (1) require expert

(11) difficult to test all path.

(3). Conditional Coverage Testing.

= $\frac{\text{Total decision exercised}}{\text{Total no. of decision in program}} \times 100$

→ Condition Coverage reports the true or false outcome of each boolean sub-expression.

Advantage → validate all the condition.

→ remove issue of Statement coverage

Disadvantage → hard to determine test cases.

(4) Branch Coverage Testing

Branch coverage = $\frac{\text{No. of Executed Branches}}{\text{Total no. of Branches}}$

* every outcome from a code module is tested.
* It help to measure fraction of independent code segments and find out section having no branches.

Adv. (I) approve branches
(II) remove issue of statement Coverage

Disadv. (I) costly

(5) loop Coverage Testing

* validity of loop constructs.
(I) To solve the problem of endless loop repeating
(II) To be aware of the performance.
(III) The goal is to figure out what's wrong with the loop's Startup
(IV) To find the variable that haven't been initialized.

Adv. (I) Infinite loop test
(II) issues related loop solved

Disadv. (I) Not significant.

Black Box Technique. - No need of internal structure

(I) Boundary value Analysis
- test cases are designed by using boundary values.
- Used in range checking
- valid boundaries & Invalid boundaries
- eg

(II) Equivalence class partitions.
- Input are divided into class with similarity
- and test any one value from class.
- reduce time required
- less no of test cases.
- Invalid & valid class.

(III) State transition Testing
- test finite State Machine
- used when System is defined in terms of finite number of states and transitions b/w States are governed by the rule of System.
- test when system is dependent on past event
- eg. 3 time login fail then Block ATM Card.

(iv) Cause effect Graphing

Causes - input & effect - result / output.

→ minimize the test case and maximize test coverage.

- uses graph.

Steps

1) Specification divided into pieces.
2) Causes and effect identified.
3) Cause and effect graph created
4) Decision table created
5) Convert to test cases.

(v) Decision Table

→ test various input combination
→ represented in tabular form.
→ cause effect table.
→ requirement are easily mapped.
→ Simple.
→ Complex when test cases increases.

4) Use Case Testing

- Identify the test cases the cover the entire system on a transaction to transaction basis, from start to finish.

→ possible interaction blw user & system tested.

Experience Based Technique.

- when we have no documentation & less time
- Tester's Experience and Skill work.
- find the user used area.

(1) Error Guessing testing.

→ tester's Skills, experience & intuition.
→ Guess the possible bug
→ Cover flaws of BVA & EPT(ECP)
→ focuses on.
  (1) More complex area
  (2) recently added area
  (3) Done by new developer
  (4) Done by multiple developers

(2) Exploratory Testing.

→ when we dont have requirement.
→ Explore the application in all possible way
→ To test cases are created.
→ thinking activity.
→ Understanding flow of application.
→ Prepare test document and then testing the application.

Types (1) freestyle – Donot follows any rule
(2) Strategy based - with help of other testing
(3) Scenario based - with help of multiple scenario.

functional Testing

- test the function, feature of app.
- test the, Business requirement.
- can be done manually
- 4 types (1) unit testing (2) Integration (3) System (4) Acceptance.

Non-functional Testing.

- check the behaviour of System
- Check performance of System
- accuracy, reliability, Security, usability, etc.
- can't done by manual, time taking.

levels of testing

(1) Unit testing. (white box testing).

- Unit testing is the testing of individual component of a product.
- done by developer.
- uses stub & driver.
- testing module one by one

Stub

developed → X — top down

developed → Y

not developed → Stub Z — dummy of Z

driver

Driver for X — bottom up → not developed

Y → developed

Z → developed

11) Integration testing

*. When 2 or more modules are integrated or merged to test if the product works fine

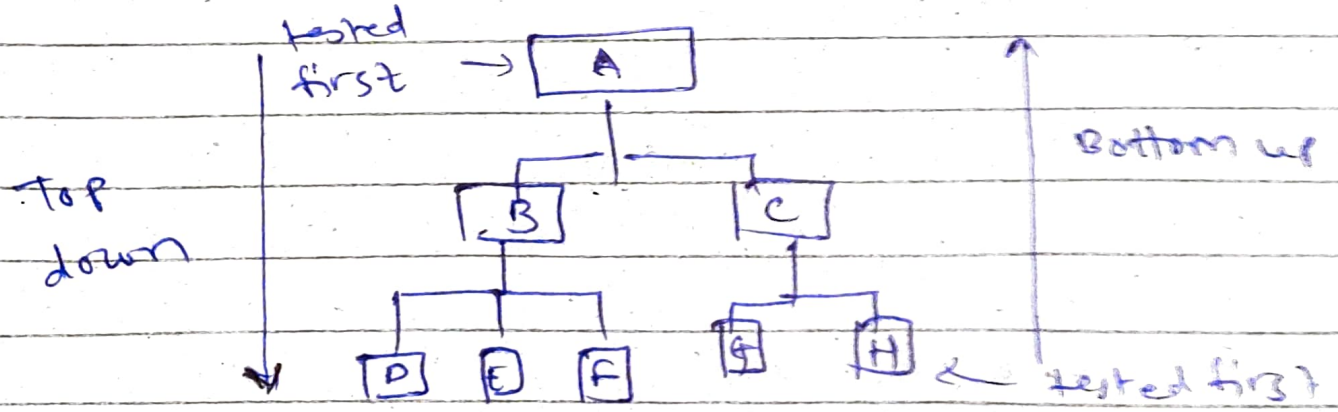
→ integration testing

→ After unit testing

Approaches

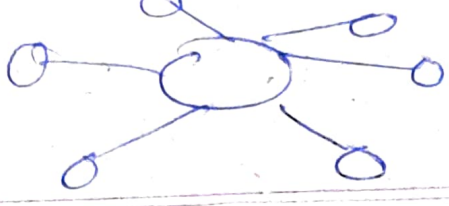
(I) Top down approach — uses Stub

(II) Bottom up approach — uses driver



(III) Sandwich or mixed Approach.

- combination of top-down & Bottom up.



(iv) Big Bang Approach
- All units are assembled and tested.
- drawback
  → if error found it will difficult to find error causing module.
  → time consuming.

(III)
System Testing
→ testing fully integrated Software System.
→ After unit & Integration testing
→ Black Box testing
→ end to end testing is performed to cover the Complete Scenario.
→ eg. Regression testing, load testing, functional testing, migration testing, usability testing etc.

(IV) User Acceptance Testing.
→ It validate end to end business flow
→ done by User.
→ Black Box testing
→ After system testing, user acceptance required for deployment.

Alpha Testing → done by Internal employee of Company
→ development site

Beta testing → performed by Client.
→ at end-user, customer site.
→ Beta version is released in real environment to get feedback from user

* Smoke testing - when new functionality is added.
→ Initial stage
→ It check the stability of the build received for software testing
→ to check the critical functionalities of the programs.
→ Done before Sanity & regression testing.
→ scripted

Advantages (I) easy
(II) Identify defects at early stages.
(III) Save time.

Disadvantages. (I) Not cover all functionality
(II) error may occur after Smoke test also.

* Sanity testing
→ performed on stable build.
→ It ensure that new modifications don't change the software's current functionalities.
→ subset of Regression testing
→ It save the time.
→ Specialized testing.

→ non scripted.

Advantages (I) help to identify defect quickly.
(II) If found defect, the project rejected and save the time.

Disadvantage (I) Difficult to fix the bug found in Sanity testing.
(II) unscripted so, future references are not available.

* Regression testing.
→ Regression testing is a software testing practice that ensure an application still functions as expected after any code changes, updates or improvements.
→ check stability and overall functionality

It apply when.
(I) new requirement is added
(II) new feature / function is added
(III) Change in configuration
(IV) Defect fixing.
(V) Performance issue fix
→ It ensure quality as the product improved.
→ generic testing

Defect Change → Prioritize change → Determine entry Point → Determine Exit Point → Schedule Test

Advantages → maintain quality & source code
→ automated.

Disadvantage → time & resource consuming.
→ require after small changes also
→ test cases may become large.

* Retest
→ test cases find the bug and send to developer to fix it. Then after fixing bug assig to back to tester for verification. This continuous process is called retest.
→ higher priority over regression.
→ manual.
→ planned

Advantages (I) Confirm that issue is fixed
(II) quality

dicadvantage (I) Manual
(II) require new build for verification & detect.

Non-functional Testing.

Performance Testing:

→ used for testing the speed, response time, Stability, reliability, Scalability and resource usage of a software application under a particular workload

→ Speed

→ Scalability - maximum load

→ Stable →

Types

(1) Load testing

- It test the how system response when load is increased on the system.

→ It should able to handle 1000 concurrent users.

→ Scalability check

(2) Stress Testing

→ It is done beyond the system's capacity.

→ The objective is to check break point of an application.

→ It check system's robustness and error handling capability under Extremly heavy load.

(3) volume Testing.

→ testing with huge amount of data to verify the efficiency and response time of the software.

→ check data loss.

performance testing process

① identify test environment.
② identify the performance acceptance criteria.
③ plan and design performane test.
④ configuring the test environment.
⑤ implement test design
⑥ run the test
⑦ Analyze, tune and retest.

* Memory testing.

→ confirm that each storage location in a memory device is working.

→ check portable, public domain Memory test function.

* Scalability Testing.

→ measure the performance of a system or network when the no of user request is scaled up or down.

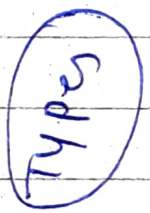
↳ performance testing

→ determine user limit for web application.
→ Attributes - throughput, Response time, Screen transition Time, CPU/Memory usage, Network Usage etc.
→ 7 steps of performance testing.

---

* Compatibility Testing.
→ verifies whether the application/product is compatible with all the hardware/ software platform or not.

① Hardware
② OS
③ Software
④ network
⑤ Browser
⑥ Mobile
⑦ Devices
⑧ versioning software.



forward
Compare with new version

&

Backward
Compare with older version.

*Security Test
→ Identify the threat in the system and measure its potential to handle threat.
→ System should not stop after attack also

(1) Authorization
(2) Authentication
(3) Error handling
(4) Data access
(5) URL access restrictions.

eg. * Password should be in encrypted.
* Should not allow Invalid User.
* Check cookie & session time

* Cookie Testing

→ It check cookie created in your browser.

Cookie - User data store on user's hardware and Sent back to server if request is made
- used to ~~end~~ rapid communicate with page & user.

Types - ① session cookie - active for when user is on web page
② persistent cookie - store on computer for long time.

Test (1) Disable cookies and test
(2) Corrupting cookies (change) and test
(3) Cookie encryption.
(4) test with multiple browser
(5) test if cookie is deleted
(6) test if cookie rejected
(7) test cookie access
(8) No overuse of cookie
(9) Test with diff. setting
(10) categorize cookie

* Session Testing - Session Based Testing.
- aim to combine accountability and exploratory testing to provide rapid defect discovery.

* Recovery Testing.
→ Recovery Testing is done to verify that the data or system is recovered if any fault occurs, application crashed or power goes off.

→ verify the system's ability to recover from varying degree of failure.
→ verify backup.

* Installation Testing.
→ It check whether the software is installed in new system with its prerequisites and necessary data and ~~its~~ is working as expected or not.

→ eg. Installation through internet
Test cases
(1) Bad network speed
(2) Broken connection
(3) concurrent installation.

* Adhoc Testing

→ randomly performed
→ no documentation, no test case no test design
→ done after formal testing.
→ informal and unstructured..
→ take less time

1. BUDDY Testing → one Buddy from development and one from ~~develop~~ tester

2. Pair Testing → 2 ~~to~~ tester
→ 1 for test cases
→ 1 for notes.

3. Monkey Testing
→ test without test cases with a goal to break the system.

* Risk Based Testing.
- testing based on probability of risk.
- It involve assessing the risk based on Complexity, criticality of business, frequency & use, possible area with Defects.

- Risk is occurence of uncertain events.
+ve Risk -ve risk

1. make a prioritized list of risk.
2. perform testing that explore each risk.

*I18N Testing. (Internationalization)

→ It test Internationalization.
→ It mean it is a process of designing a Software application. Such it can adapted to various languages, regions, culture.

→ first should be on user Interface
→ second on content localization.
→ third culture awarness
→ And file transfer

*LION Testing (Localization)

→ making product / App that adapt to meet culture, lang, etc of specific region, or a locale.
→ Time, Date, Currency, Mobile no., postal code etc

Globalization Testing → LION + I18N.

*Compliance Testing

→ Determine that the development and maintenance meets the prescribed Methodology.

→ Ensure whether the deliverable of each phase of the development meets the standards, procedure and guidelines.

→ Evaluate the documentation of the project to check for completeness and reasonableness.

* It is conformance testing

* Done to validate whether the system developed meets the organizations prescribed standard or not.