

BVA

(Boundary value analysis)

Boundary value analysis is one of the widely used case design technique for black-box testing. It is used to test boundary value because the input value near the boundary have higher chances of error.

Whenever we do the testing by boundary value analysis, the tester focuses on whether entering boundary correct output or not.

BVA also those that contain the upper and lower limit of a variable. Assume that, age is a variable of any function, and its minimum value is 18 and maximum value is 30, both 18 and 30 will be considered as boundary values.

The basic assumption of BVA is the test cases that are created using Boundary value analysis are most likely to cause an error.

There is 18 and 30 are the boundary values that's tested pays more attention to these values, but this doesn't mean that the middle value like 19, 20, 21, 27, 29 are ignored. Test cases are developed for each every value of the range.

Name	Enter your name
Age	Between 18 to 30
Aadhar	Number of 12 digits

- Strategic Approach to SW testing:-
testing is a set of activity that can be planned in advance and conducted systematically. testing strategy should have the following characteristics
- Usage of formal technical review (FTR)
- Begins at component level and covers entire system
- Different techniques at different points
- conducted by developer and test group.
- should include debugging.

1)
2)
3)
4)



SW testing is one element of verification and validation.

Verification refers to the set of activities that ensure that SW correctly implements a specific function.

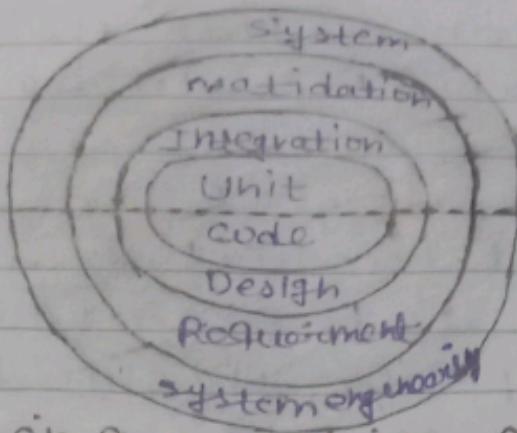
Ex: Are we building the product right?

validation refers to the set of activities that ensure that the SW built is traceable to customer requirements.

- Test strategy:- A high level document is used to validate the test types at level to be executed for the product and specify the SW development life cycle's testing approach is known as test strategy document. Test strategies for conventional SW level of testing.

- 1) unit testing
- 2) sub testing
- 3) system testing
- 4) validation testing

★ Test strategies Conventional SW:-
Conventional testing refers to the traditional approach of SW testing that has been widely used for several decades. This approach involves a series of activities that aim to identify ensure that the SW meets the specified requirement and performs as expected.



* Stage in conventional testing:-

The conventional testing process typically follows a structured and sequential approach that consists of several stages, starting with the planning phase and ending with the release of the SW. The following are the key stages involved:

Conventional testing:

Planning

Requirement analysis (1) Design (2) Execution

Deployment (3) Testing (4) Release

• type of conventional testing

Unit testing

Integration testing

System testing

Acceptance testing

Regression testing

Advantages of conventional testing:-
conventional testing is also known as manual testing, involves testing s/w applications by human tester without the use of automated tools. Some advantages of conventional testing include:

- (1) flexibility:-
- (2) human intuition
- (3) cost-effective
- (4) Better understanding of user experience.
- (5) testing of non-functional requirement
- (6) Better communication

* Disadvantages:-

- 1) Time-consuming
- 2) Limited coverage
- 3) Subjective
- 4) Costly in the long run
- 5) Repetitive
- 6) Error-prone:-

⇒ Basic Terminologies:- Here are some s/w testing terminologies that are widely and popularly used in s/w testing and information technology industries.

Acceptance testing:- formal testing of user needs, requirements, and business

processes conducted to determine whether or not a system meets the acceptance criteria and to allow the user, customers, and other authority entity to determine whether or not to accept the system.

(2) Alpha testing:-

(3) Attack:- Directed and focused attempt on quality assessment, especially reliability of a test object, trying to force specific failures.

Beta testing

Black Box testing

Boundary value Analysis.

Branch coverage:- The percentage of branches that were evaluated in a set of tests. One hundred percent effective coverage implies 100% decision coverage and 100% declaration coverage.

- 8) code coverage:- An analysis method that determines which parts of the ~~program~~ were run by the test suite and which parts were not executed.
- 9) compiler:- A ~~software~~ tools that translates programs expressed in a machine language.
- 10) complexity:- The degree to which a component or system has an internal design and / or structure that is difficult to understand, maintain and verify.
- 11) component integration testing:- Test performed to identify defects in interfaces and interactions between integrated components.
- 12) component testing:- Testing individual components.
- 13) configuration control:- A configuration management element which consists of enabling, coordinating, approving, or disapproving, and implementing changes to configuration items after the formal establishment of your

configuration ID.

(i)

configuration item:- An aggregation of hardware, software, which is intended for configuration management and treated as a single entity in the configuration management process.

(ii)

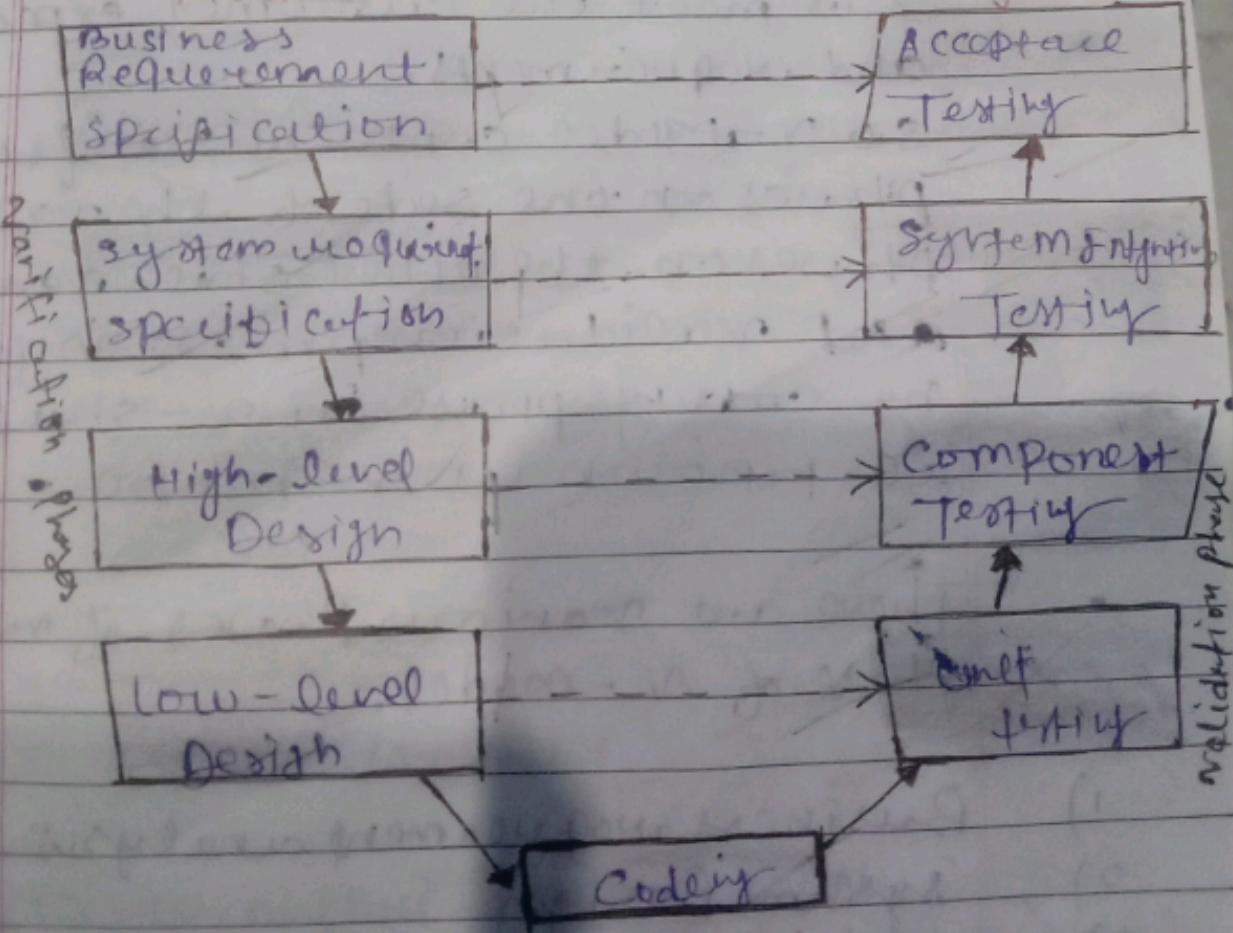
Configuration Management:- A discipline that applies selection, techniques and administrative surveillance to identify and document the functional and physical characteristics of a configuration item, track changes in those characteristics, record and report change processing and implementation status, and verify compliance with the specified requirements.

* V-shaped SW life cycle

V-Model also referred to as the verification and validation model. In this, each phase of SDLC must complete before the next phase starts. It follows a sequential design process same as the waterfall model. Testing of the device is planned in parallel with a corresponding stage of development.

V-Model

Development life cycle Testing life cycle



Verification:- It involves a static analysis method (reviews) done without executing code. It is the process of evaluation of the product development process to find whether specified requirement meet.

Validation:- In involves dynamic analysis method (functional, non-functional), testing is done by executing code. Validation is the process to classify the SW after the complement of the development process to determine whether the SW meet the customer expectation and requirements.

So V-Model contains verification phases on one side of the validation phases on the other side. Verification and validation process is joined by coding phase in V-shape that is known as V-model.

There are various phases of verification phase of V-model.

- 1) Business requirement analysis
- 2) System Design
- 3) Architecture Design

- u) Module Design
- s) Coding phase

- There are various phases of validation

v-model :-

- 1) Unit testing
- 2) Integration testing
- 3) System testing
- 4) Acceptance testing

- When to use v-Model?

⇒ When the requirement well defined and not ambiguous.

⇒ The v-shaped model should be used for small to medium-sized projects where requirement are clearly defined and fixed.

⇒ The v-shaped model should be chosen when sample technical resources are available with essential technical expertise.

ADVANTAGES :-

1) Easy to understand.

2) Testing method like planning, test designing happens well before coding.

- 3) This saves a lot of time. Hence a higher chance of success over the waterfall model.
- 4) Avoids the downward blow of the defects.
- 5) Works well for small projects where requirements are easily understood.

DISADVANTAGES (cons) V-Model:-

- 1) very rigid and least flexible.
- Not a good for a complex project.

* Equivalence class testing:-
equivalence class testing is a black box testing technique used in software testing as a major step in the software development life cycle (SDLC).
this testing technique is better than many of the testing techniques like BVA worst case testing. Robust case testing is many more in term of precision of the test cases. Since testing is done to identify possible inputs, equivalence class testing performs better than other techniques as the test cases generated using it are logically identified with partitions in between to create different input and output classes. this can be shown from the next - date problem which is stated below:

Given a day in the format of day-month-year, you need to find the next date for the given date. perform boundary value analysis and equivalence-class testing for this.

Conditions:-

$$D : 1 \leq \text{Day} \leq 31$$

$$M : 1 \leq \text{Month} \leq 12$$

* Basic Path testing:- Basic path testing is a white-box testing technique based on the control structure of a program or a module. Using this structure, a control flow graph is prepared and the various possible paths present in the graph are executed as a part of testing. Therefore, by observation, basic path testing is a technique of selecting the path in the control flow graph, that provide a basis set of execution path through the program or module. Since this testing is based on the control structure of the program, it requires complete knowledge of the program's structure. To design test cases using this technique, four steps are followed:

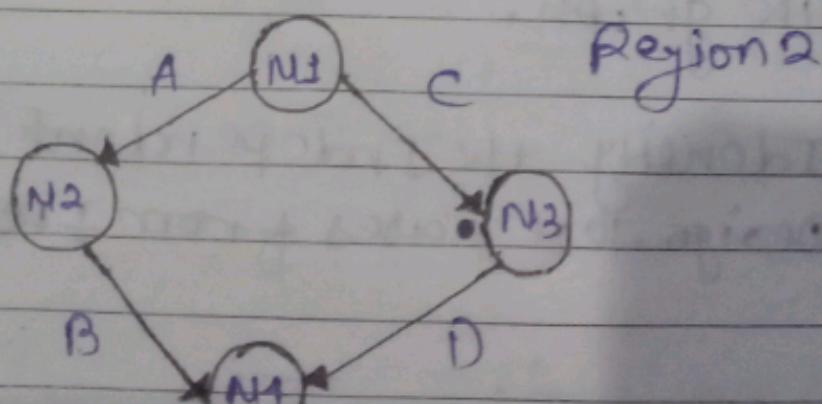
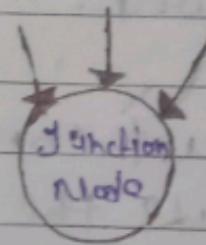
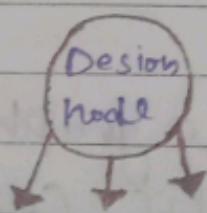
- Construct control flow graph
- Compute the cyclomatic complexity of the graph.
- Identify the independent path
- Design test cases from independent path

Control flow graph:- A control flow graph is a directed graph which represents the control structure of a program or module. A control flow graph (V, E) has V number of nodes and E number of edges in it. A control graph can be also have:-

Function Node:- A node with more than one arrow entering it.

Decision Node:- A node with more than one arrow leaving it.

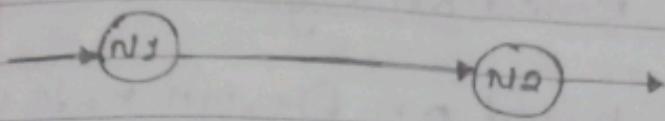
Region :- A local bounded by edges and nodes (area outside the graph is also counted as a region).



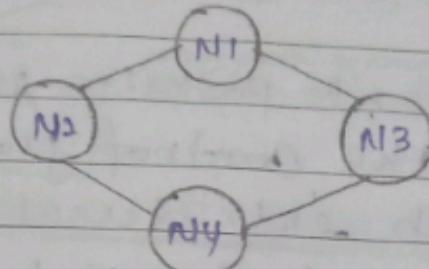
Below are the notations used while constructing a flow graph:-

sequential statements:-

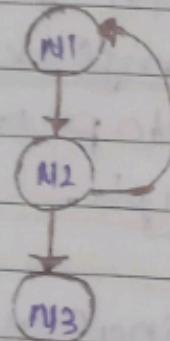
sequential



If - Then - Else :-

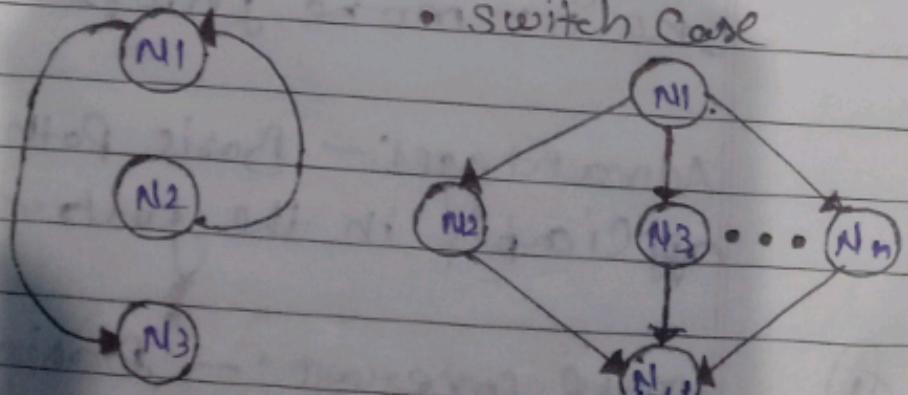


Do-while



do-while / while-Do

switch case



* **Cyclomatic complexity** : — The cyclomatic complexity $v(G)$ is said to be a measure of the complexity of a program. It can be calculated using

Three different formula

(1) formula Based on edges and node :-

formula based on Design node :-
formula based on Regions

(1) Independent path:- An independent path in the control flow graph is the one which introduces at least one new edge that has not been traversed before the path is defined. The cyclomatic complexity gives the number of independent paths present in a flow graph.

Design

* Test cases:- finally, after obtaining the independent paths, test cases can be designed where each test case represents one or more independent paths

Advantages:- Basic Path testing can be applicable in the following cases:-

(1) more coverage:- Basic path testing provide the best code coverage if it aims to achieve maximum logical coverage instead of maximum path coverage. they results in an overall

through testing of the code.

(2) Maintenance testing:- when a SW is modified it is still necessary to test the changes made in the SW which has a result in required path testing.

(3) unit testing:-

(4) Integration testing

(5) Testing effort:- since the basic path testing technique takes into account the complexity of the SW (i.e. program modules)

Ques:- what are advantages and disadvantages of path coverage in SW testing?

Answer:- Path coverage testing:- A structured white-box testing method called path coverage testing is used in SW testing to examine and confirm that every possible path through a program's control flow has been tested at least once.

Advantage:-

i) Provides comprehensive code coverage, ensuring all possible execution paths are tested.

Effectively uncovers complex logical bugs and issues related to code branching and loops.

Help improve SW quality and reliability by thoroughly testing all code paths.

Utilizes a standardized metric, Cyclomatic Complexity, for assessing code complexity.

Useful for demonstrating maintainability and reliability in industrial settings.

Disadvantages:-

- Demands a high testing effort, particularly for complex code, leading to resource-intensive testing.
- Requires an exponential growth in the number of test cases as code complexity increases.
- Focuses on code path but may not cover all potential runtime conditions or input combinations.
- Maintaining a comprehensive set of test cases as code evolves can be challenging.
- There is risk of overemphasizing coverage quantity over quality, potentially neglecting lower-priority code paths.

Ques:- How decision table will be helpful in logic based testing?

Ans:- Decision tables are used in various engineering fields to represent complex logical relationships. This testing is a very effective tool in testing the SW and its requirement management. The output may be dependent on many input conditions and decision table give a tabular view of various combinations of input conditions and these conditions are in the form of true(1) and false(0). Also, it provides a set of conditions and its corresponding action required in the testing.

PART OF Decision-table:-

Condition stubs:

Action stubs

Condition Entries

Action Entries

Type:-

① Limited Entry

② Extended Entry

① Limited Entry:- In the limited entry decision table the condition entries are restricted to binary value.

(2). Extended entry :— In the extended entry decision tables, the condition entries have more than two values.

• Applicability of Decision tables : -

- 1) The order of rule evaluation has no effect on the resulting action.

2) The decision table can be applied easily at the until level only.

3) Once a rule is satisfied and the action selected, no other rule needs to be examined.

4) The restrictions do not eliminate many applications.

• Advantages : -

1) Decision table are one of the most effective and fail-proof design testing technique.

2) It gives the developer to state and analyze complex business rules.

3) Decision table testing is the most preferred black box testing and requirement management.

A decision table is an excellent tool to use in both testing and requirement management. Essentially it is a structured exercise to formulate requirement when dealing with complex business rules. Decision tables can be used to model complicated logic.

Qn:- ③ Explain different type of test and when they need to be carry out?

Ans:- There are basically 10 type of testing.

① unit ② Integration ③ system
functional ④ Acceptance ⑤ smoke

⑥ Regression ⑦ performance ⑧ security
⑨ user Acceptance testing.

① Unit Testing:- Unit testing is a method of testing individual units or components of a software application. It's typically done by developers and it used to ensure that individual unit of the software are working as intended. Unit tests are usually automated and are designed to test specific parts of the code such as a particular function or method.

2) Integration testing:- Integration testing is a method of testing how different unit or components of a software interact with each other.

③ Regression testing: — Regression testing is a method of testing that is used to ensure that changes made to the system do not introduce new bugs or cause existing functionality to break.

- Regression testing can be performed in different ways, such as:-

- 10 Repeating ② Re-execution!
③ Comparison

④ Smoke testing:- smoke testing is done to make sure that the SW under test is ready or stable for further testing.

it's called Smoke test as the testing of an initial pass is done to check if it did not catch fire or smoke in the initial switch-on.

Alpha Testing :- Alpha testing is a type of validation testing. It is a type of acceptance testing.

that is done before the product is released to customers, it is typically done by QA people.

(c) Beta testing:- The beta testing is conducted at one or more customer sites by the end-user. If the software revision is released for a limited number of users to verify in a real-time environment.

(7) System testing:- System testing is carried out on the whole system in the context of either system requirement specifications or functional requirement specification or in the context of both. The SW is tested such that it works fine for the different OS.

(d) Stress testing:- In stress testing, we give unfavorable conditions to the system and check how they perform in those conditions.

(e) Object-oriented testing:- Object-oriented testing is a combination of various testing techniques that help to verify and validate object-oriented SW. This testing is done in the following

manner.

- Testing of environment
- Design and analysis of testing
- Testing of code.
- Integration testing.
- System testing
- User testing.

• Acceptance testing:-

Acceptance testing is done by the customers to check whether the delivered products perform the desired tasks or not, as stated in requirements. We use object-oriented testing for discussing test plans and for executing the projects.

* Advantages of software testing:-

- Improved software quality and reliability.
- Early identification and fixing of defects.
- Improved customer satisfaction.
- Increased stakeholder confidence.
- Reduced maintenance costs.

* Disadvantages:-

- Time-consuming and adds to the project cost.

- This can slow down the development process.
- Not all defects can be found.
- can be difficult to fully test complex systems.
- potential for human error during the testing process.

Ques:- Q) what is control flow graph explain how to generate control flow graph with example?

Ans:- A control flow graph (CFG) is the graphical representation of control flow of computation during the execution of programs by application.

Control flow graphs are mostly used in static analysis as well as compiler applications, as they can accurately represent the CFG was originally developed by Frances E. Allen.

* characteristics of CFG:-

- CFG is process oriented.

CFG show all the path that can be traversed during a program execution.

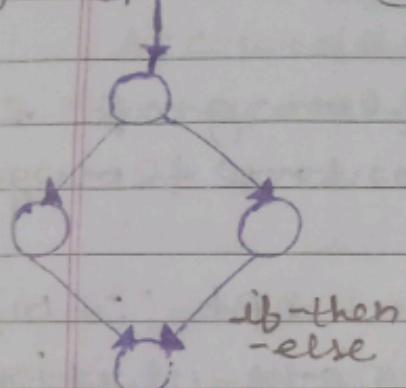
CFG is a directed graph.

Edges in CFG portray CFG and the nodes in CFG portray a basic block.

- ★ there exist 2 designated blocks in CCGF
- Entry Block:- entry block allows the control to enter the CCGF.
- control flow leaves through the exit block.
- control flow is represented differently for all statements and loop.

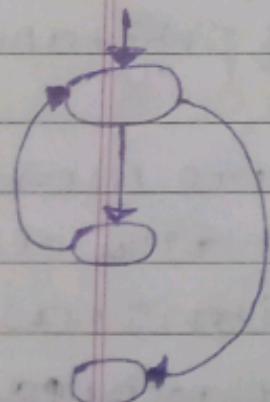
①

IF-else ④ For



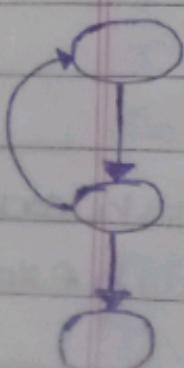
②

while



③

do-while

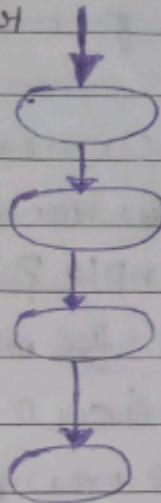


Entry Block:- entry block allows the control to enter the CCGF.

control flow leaves through the exit block.

control flow is represented differently for all statements and loop.

④ For



ex:- if $A = 10$ then,

if $B > C$

$A = B$

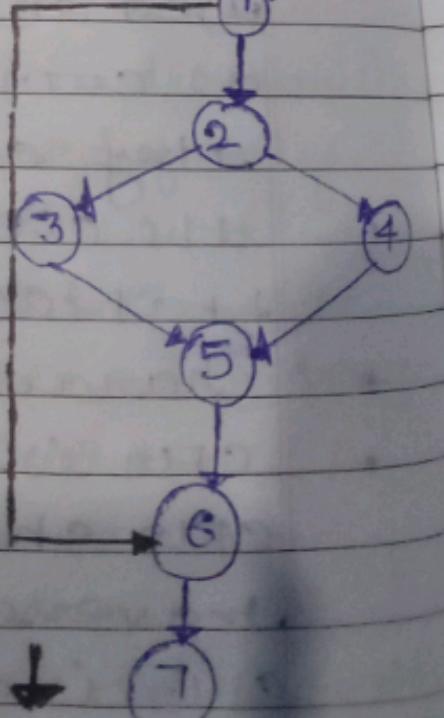
else $A = C$

endif

endif

print A,B,C

ex:- CCGF



Control flow graph

Ques:- Explain STLC?

The SW testing life cycle (STLC) is a systematic approach to testing a SW application to ensure that it meets the requirements and is free of defects. It is a process that follows a series of steps over phases, and each phase has specific objectives and deliverables. The STLC is used to ensure that the SW is of high quality, reliable, and meets the needs of the end-users.

The main goal of the STLC is to identify and document any defects or issues in the SW application as early as possible in the development process. This allows for issues to be addressed and resolved before the SW is released to the public.

- * Characteristics of STLC: -
- STLC is a fundamental part of SW Development Life Cycle (SDLC) but STLC consists of only the testing phase.

STLC starts as soon as requirements are defined and SW requirement document is shared by stakeholders.

- ATLC yields a step-by-step process to ensure quality assurance.

Phase:-

(1)

Requirement analysis :- Requirement analysis is the first step of the software life cycle. In this phase quality assurance team understandable with the stakeholders to better understand the detailed knowledge of requirements..

(2)

The activities that take place during the requirement Analysis stage include:-

1.)

Reviewing all the requirements(SR) and other related documents.

2)

Interviewing stakeholders together additional information.

3)

Identifying any missing or incomplete may impact the testing process.

4)

Identifying any potential which will issues that may impact the testing process.

(2.) Test Planning: - Test Planning is the most efficient phase of the software testing life cycle where all testing points are defined. In this phase manager of the testing team calculates the estimated effort and cost for testing work.

- The activities that take place during the test Planning stage include:
 - Identifying the testing objectives and scope.
 - Identifying the testing environment and resources needed.
 - Identifying the test cases that will be executed and the test data that will be used.
 - Reviewing and ~~approving~~ ^{approving} that test plan

③ Test case development: - The test case development phase gets started once the test planning phase is completed. In this phase testing team notes down the detailed test cases.

Ques: 8 Elaborate process of Black-box testing with example of any 4 techniques of black box testing.

Black box testing techniques apply to all level of testing, as well as functional and non-functional testing types. There are four main black box testing techniques: equivalence partitioning, boundary value analysis, decision table testing, and state transition testing.

• There are four main black box testing techniques:-

- 1) equivalence Partitioning, 2) boundary value analysis ③ Decision-table testing ④ State Transition testing.

(1) Equivalence Partitioning (EP)

EP is a technique that divides test object into partitions or classes that are treated and tested in the same way. This technique is used for partitioning of values, input, and outputs. There are valid and invalid equivalence partitions.

(2)

Boundary Value Analysis:— This technique test the application by using the BVA

input data test cases are designed to include values at the edges of input domain to ensure the system handles these boundary effectively.

(3) Decision table testing:- Decision tables are used to model complex business rules. Test cases are designed to cover all possible combinations and their respective actions.

(2)

(4) state transition testing:- this technique is used for system that can be in different state and transition from one state to another based on certain test case are designed to validate the correct

(3)

Ques:- (9) what are different type of test? In software testing different levels of testing are performed to insure the quality and functionality of the S/W. These level of testing are organized based on the different stages of SW development life cycle and specific objective they aim to achieve the various levels commonly include.

- (1) Unit testing:- this is lowest level of testing where individual unit or component of the sw are tested independantly. it focus on verifying the smallest testable part of the software, such as functions or methods to ensure they was as intended.
- (2) Integration testing:- it verifies the propuse interaction between different modules or unit of the sw. it ensure that integrate component work together as expected and that the interface between various modules function correctly.
- (3) System testing:- In system testing, complete and integrated sw are tested i.e. all the system elements forming the system is tested as a whole to meet the requirement of the system.
- (4) Acceptance testing:- it is a kind of testing conducted to ensure whether the requirement of the user are fulfilled prior to its delivery and the sw work correctly in the user working environment.

Ques:- (10) Give the difference between verification & validation

①

verification

• it include checking documents, design, codes and programs.

②

verification is the static testing

validation

it include validating actual products

validation is dynamic testing

③

it does not include the execution of the code.

it include the execution of the code.

④

Method used in verification are reviews, walkthroughs, inspections and desk-checking.

Method used in validation are Black box - testing and non-functional testing.

⑤

It can find the bugs in the early stage of the development

It can only find the bugs that could not be found by the verification process

It come before validation

It comes after verification

Ques :- drivers & stubs

stubs
drivers

drivers
stubs

- | | |
|---|--|
| <p>(1) stubs are used in top-down integration testing.</p> | <p>Drivers are used in bottom-up integration testing.</p> |
| <p>(2) Stubs are basically known as "calling programs" and are used in top-down integration testing while drivers are the "called program" and are used in bottom-up integration testing.</p> | |
| <p>(3) Stubs are basically used in the unavailability of low-level modules.</p> | <p>Drivers are mainly used in place of high-level modules and in some situations as well as for low-level modules.</p> |
| <p>modules are done and the lower level of the modules are under development process.</p> | <p>modules are done and the upper level of the modules are under developing process.</p> |

Ques:- Q) Software testing and debugging

Ans:- Testing:- Testing is the process of verifying and validating that a SW or application is bug-free, meet the technical requirements as guided by its design and development, and meets the user requirements effectively and efficiently by handling all the exceptional and boundary cases.

Debugging:- Debugging is the process of fixing a bug in the software. It can be defined as identifying, analyzing, and removing errors.

This activity begins after the SW fails to execute properly and concludes by SW fails to execute properly and concludes by solving the problem and successfully testing the SW. It is considered to be an extremely complex and tedious task because errors need to be resolved at all stages of debugging.

Ques:- What is requirement & specification?
SRS In order to form a good SRS, here you will see some points that can be used and should be considered to form a structure of good software requirement specification (SRS).

The production of the requirement stage of the SW development process is SW requirement specification (also called a requirement document). This report lays a foundation for all engineering activities and is constructed when entire requirement are elicited and analysed. SRS is a formal report, which acts as a representation of SW that enables the customers to review whether it is according to their requirements.

Characteristics of good SRS:-

(1) Correctness

Completeness :- All SRS is complete if and only if it includes the following element:-

All the element requirements, whether relating to functionality,

performance, design, construction, attributes, all external interfaces.

(3) consistency: (4) unambiguity:

(5) Ranking for importance and stability
(6) Modifiability (7) verifiability

(8) Traceability (9) Design independence

(10) Testability (11) Understandable by the customer

(12) The right level of abstraction

Ques:- (11) What are the benefit of software validation?

Ans:- here are some benefits:-

(1) Cost reduction:- validation helps reduce cost for business and improves product performance.

2) Quality improvement! - A comprehensive validation process can improve the quality of the output for a particular development project.

Business:
and Stakeholder
independence
by

3) Customer satisfaction:- validation helps in reducing the likelihood of a critical defect and ensures the clients and end user are happy with the product.

4) Complaint reduction:- validation helps ensure the quality of the product and reduces complaints, launch failures and other problems.

Ques:- Q. What are graph matrices explain with example?

Ans:- A graph matrix is a data structure that can assist in developing a tool for automation of path testing. Properties of graph matrices are fundamental for developing a test tool and here graph matrices are very useful in understanding the testing concepts and theory.

A graph matrix is a square matrix whose size represents the number of nodes in the control flow graph. If you do not know what control flow graphs are, then read this article. Each row and column in the matrix identifies a node and the entries in the matrix

- The graph metrics formed is shown below:-

	1	2	3	4
1	a	b	c	
2				d
3				e
4				f

Ques:- 1) Explain the verification activity in traditional SW?

Ans:- Activities performed in the normal course of SW development such as requirement review, design review, unit testing static, build audits are the example of verification. Verification evaluates SW artifacts (such as requirement, design code, etc.) to ensure they meet the specified requirements and standards. It ensures the software is built according to the need and design specifications. Validation evaluates SW to meet the user's need and requirements.

Ques 2) :- What are the test tools recommended for the design phase testing?

Ans:- The effect of test design tool One the testing phase, we should know about the concept that lies in

the course of their works, i.e., by design.

test design techniques :-

A tester can application from many corners rather than moving linearly in one methodology.
different methodologies helping us different type of test.

following are different type of test design techniques.

- ① state transitions test design.
- ② Equivalent class test design
- ③ Boundary value analysis test design.
- ④ error guessing test design.

Question) Explain functional and Structure testing in detail? with example.

Answer:-

- i) structure testing :- this test evaluates the code structure but not the implementation of the code.
- ii) it is also known as white-box or clear-box testing as thorough knowledge and access of the code is required.
- iii) find errors in the internal code logic and data structure usage.
- iv) It does not ensure that the user requirements are met.
- v) performance low-level modules/ sub component.
- vi) structure testing tools follow data analysis methodology.
- vii) writing a structure test case requires understanding the coding aspects of the application.

- * Different type of structure technique
 - (1) mutation
 - (3) slice-based
 - (2) Control flow
 - (4) Data flow

(2) (1) Functional testing: — This test check whether the SW is functioning in accordance with functional requirement and specification.

2) It's also known black-box testing as no knowledge of the internal code is required.

3) it ensures that the system is error-free

4) it is check that the output is given as per expected.

5) The functional testing tool works on event analysis methodology.

6) it provide information that prevent business loss.

7) it work on event analysis methodology

8) Before writing a functional test case a tester is required to understand the application's requirement.

the techniques
of flow
flow

Test checks
functioning is
functional
specification.

box directly
internal

system is
put in given

and work on
ogy.

that prevent

good methods of

test writing
stand the
ment.

★

Type :-

- (1) unit testing
- (2) Integration testing
- (3) system testing
- (4) Acceptance
- (5) regression
- (6) sanity testing
- (7) smoke testing
- (8) recovery testing

⇒ Example of structure testing :-

verifying the actual code for aspects like the correct implementation of conditional statement, and whether every statement in the code is correctly executed.

⇒ Example of functional testing :-

on an e-commerce webpage, users cannot add a particular product to their cart even when the stock is available.

~~Ques 11/23~~

Ques:- Difference Between directed & Undirected Graphs.

- (1) The Directed graph is also known as the digraph, which is a collection of a set of vertices edges.

DIRECTED GRAPH	UNDIRECTED GRAPH
(1) A Directed graph that contains ordered pair of vertices	(1) A undirected graph that contains unordered pair of vertices.
(2) Edges represent the direction of vertices	(2) Edges do not represent the direction of vertices.
(3) An arrow represents the edges.	(3) Undirected and unrepresent the edges.
check the all nodes are visited.	check the all nodes are visited.
The root is the node with no incoming edges.	Any node can be chosen as the root.

Difference between Testing and Debugging

Testing	Debugging
① Testing is the process to find bug and error.	Debugging is the process of correcting the bugs found during testing.
② it is the process to identify the failure of implemented code.	it is the process to give solution to code failure.
③ testing is the display of error done by tester.	Debugging is done by either programmer or the developer.
④ testing is the display of error.	Debugging is a deductive process.
⑤ testing can be done by insiders as well as outsiders.	Debugging is done only by insiders. outsiders can't do debugging.