BSc.(Information Technology)

(Semester VI)

2018-19

Software Quality Assurance

(USIT 601 Core)

University Paper Solution

By

Ms. Seema Vishwakarma

Question 1

Q1 a. What is quality? Explain its core component. .

Ans: *Quality means different things to different people at different times, different places and for different products. It can also be defined as conformance to specification and fitness for use.*

Core Component of Quality

Quality is based on Customer Satisfaction

- The effect of quality product delivered & used by a customer, on his satisfaction and delight is the most important factor in determining whether quality has been achieved or not.
- It talks about the ability of a product or service to satisfy a customer by fulfilling his/her needs.
- Manufacture must understand the purpose or usage of a product and devise a quality plan to satisfy the purpose of the product

The organisation must define Quality parameters before it can be achieved:

1. Define : Defining the product in terms of features, functionalities, attributes & characteristics of a product. What should be , could be and must be present in the project
2. Measure: The quantitative measures must be defined as an attribute of quality of a product. Measurement also gives a gap between what is expected by a customer & what is delivered to him when the product is sold.
3. Monitor: There must be a mechanism used by the manufacturer to monitor the development, testing and delivering of a product process.
4. Control: Control gives the ability to provide desired results & avoid the undesired things going to a customer.
5. Improve: Continuous improvement are necessary to maintain ongoing customer satisfaction & overcome the possible competitionManagement must lead the organisation through improvement efforts

Management must lead the organization through improvement efforts

- Management should lead the endeavor of quality improvement program in the organization by defining vision, mission, policies, objectives, goals etc. to improve the quality improvement program.
- And the same must be followed by the employees which is called as 'cultural change brought in by management'.

Continuous process:

- It is an older belief that quality can be improved by more inspection, testing and rework which leads to the increase in the cost of inspection, segregation, failure and reduces the profit margin for manufacturer.
- For improving the quality and to have a win-win situation for both manufacturer & customer, quality must be produced at the first time and must be improved continuously

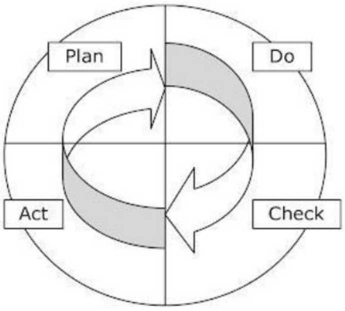
Q1 b. Differentiate between tools and techniques.

Ans:

Tools	**Techniques**
Tool is of no use unless technique is available	Technique is independent of any tool
Different technique may use same tools to get the different result	Same technique may use different tools to get the same result
Tool improvement need technological changes	Technique change can be affected through procedural change
Contribution of tool in improvement is limited	Contribution of technique in improvement is important

Q1 c. Explain continual (continuous) improvement cycle.

Ans: Continual(Continuous) improvement cycle also known as Plan, Do, Check, Act(PDCA) cycle is based on systematic sequence of plan do check and act activities representing a never ending cycle of improvements.



PLAN : An organization must plan for an improvement on the basis of its vision and mission definition . Planning includes answering questions such as who, when, where, how etc about various activities. Quality planning as unit level must be in sync with quality planning at various organization level.

DO : An organization must work in the direction set by the plan. Actual execution of plan can determine whether the results as expected are achieved or not . Plan sets the tone while execution makes the plan work. 'Do' process needs inputs like resources like hardware, software, training

CHECK : An organization must compare actual outcome of 'Do' stage with reference or expected result which are planned outcomes. It must be done periodically to check of the progress is in right proper direction and whether the plan is right or not

ACT: If any deviations are observed in actual outcomes with respect to planned results. The organization must decide actions to correct the situation. One may need to implement corrective and preventive actions as per the outcome of 'Check'

Q1 d. List and explain any five requirements of a product.

Ans:

Must/Must not requirement /Primary requirement

Must requirements are primary requirements for which customer is going to ay for while acquiring a product. These are essential requirements and the value of the product is decided based on the accomplishment of must requirement. It has highest priority and denoted by P1

Should/Should not requirement / Secondary requirement

These are the requirements that can be appreciated by the customer if present/absent in the product. Customer may pay extra for the satisfaction of these requirements. These requirements gives customer delight and are at lower priority. These requirements are denoted by P2.

Could/Could not requirement /Tertiary requirement

These requirements may add a competitive advantage to the product but may not add values in terms of price paid by a customer. These requirements have lowest priority and denoted by P3

Generic/Specific requirement

Some requirements are generic in nature and are accepted for a type of product for all the users while some others are specific to the product. Eg Additon of two numbers should be correct is generic requirement while the accuracy of 8 digits after decimal. Usability is generic requirement while authentication to users may be driven by specific requirement

Present/Future requirement

Present requirements are essential when an application is used in present circumstances while future requirements are for future needs which will be needed after some time span. Definition of future has a direct relationship with usable life of an application.

Q1 e. Explain types of products based on criticality to the users?

Ans: Life affecting product

Products that affects the life of an individual indirectly or directly are considered as most critical product by the users. The quality requirements are very stringent and testing is very critical as failure may result into loss of life or disablement of the user. They are further grouped into five categories

1. Any product failure resulting into death of a person
2. Any product failure resulting into permanent disablement of a person
3. Any product failure resulting into temporary disablement of a person
4. Any product failure resulting minor injury to a person
5. Other product which do not affect health or safety direclty

Product affecting huge sum of money

It is second in the list of criticality of the product which includes products having direct relationship with loss of huge sum of money. Such product may need large testing efforts and have many regulatory as well as statutory requirements. Security, confidentiality and accuracy are some of the important quality factors for such product

Product that can be tested only by simulator

Product which cannot be tested in real life scenario but need simulated environment for testing are third in the ranking if criticality. Products used in aeronautics, space research etc falls into this category

Q1 f. List and explain any five quality principles of Total Quality Management.

Ans: Quality principles of Total Quality Management (TQM)

Develop constancy of purpose of definition and deployment of various initiatives :

Management must create constancy of purpose for product and processes allocating resources adequately to provide for long term needs rather than concentrating on short term profitability. Decision taken by the management should be consistent

Adapting to the new philosophy of managing people :

Management should adopt to the new philosophies of doing work and getting the work done from its people and supplier. Skills makes an individual indispensable. Transformation of management style to total quality management is necessary to take the business on the path of continued improvement

Declare freedom from Mass inspection of produced output :
Mass inspection results into huge cost overrun and product produced is of inferior quality. Improving the quality of product needs setting up the right processes of development and measurement of process capabilities and statistical evidences of built in quality in all department

Stop Awarding of lowest proce tag contracts to suppliers :
Vendor selection must be done on the basis of total cost including price, rejections etc. Organization must perform measurement of quality of supply along with price and do the source selection on the basis of final cost paid by it in terms of procurement, rework, maintenance, operations etc

Drive out fear of failure from Employees:
An organization must encourage effective two way communication and other means of driving out the fear. Employees can work effectively and more productively to achieve better quality output when there is no fear of failure

Question 2

Q2 a. Explain salient features of good testing

Ans: Good testing involves following steps:
Capture user requirement
Intended requirements defined by the users or customer and the implied requirements are to be analyzed and documented by testers so that they write the test scenario and test cases for these requirements.
Capturing user needs
User needs includes present, future, process and implied requirements. Elicitation of requirements is to be done by the development organization to understand and interpret the requirements
Design Objective
Design objectives state why particular approach has been selected for building software. Functional requirement, user interface requirement are some of the requirements mentioned in software design and how can they achieve it.
User Interface
It is the way how user interacts with the system. This includes screens, displays and reports generated by the system. User interface should be simple so that the user can understand what he is supposed to do and what the system is doing
Internal structures
Internal structures are mainly guided by the software designs or standards used for designing and development. It also talks about reusability
Execution of gray box testing of code
Testing ensures that it works as intended by customer and is prevented from any probable misuse or risk of failure. Execution can only prove that application module and program works correctly
Q2 b. Differentiate between verification and validation.

Ans: Verification: *It is a disciplined approach to evaluate whether a software product fulfils the requirements or conditions imposed on them by the standards or processes. It is done to ensure that the processes and procedures defined by the customer and/or organisation for development & testing.*

Following are the techniques of Verification:

1.Self-Review: It may not be considered as an official way of review, because it assumes that everybody does a self-check before giving work product for further verification.

2.Peer Review: It is the most informal type of review where an author & a peer are involved. Review records are maintained.

3.Walkthrough: It is a semi-formal type of review as it involves larger teams along with the author reviewing a work product.

4.Inspection: It is a formal review where people external to the team may be involved as inspectors. They are the 'Subject Matter Experts' who review the work product.

5.Audit: It is a formal review based on samples. Audits are conducted by the auditors who may or may not be the experts in the given work product.

Validation: *It is used to evaluate whether the final built software product fulfils its specific intended use. It is also called as 'Dynamic Testing'. It must be done by the independent users, functional experts, and black box testers to ensure independence of testing from development activities. It helps in analysing whether the software product meets the requirements as specified in requirement statement.*

Following are the levels of validation:

1. *Unit Testing*
2. *Integration Testing*
3. *Interface Testing*
4. *System Testing*
5. *Cause & Effect Graphing*
6. *Path Expression & Regular Expression*

Q2 c. List and explain any two approaches of software testing team with its advantages and disadvantages.

Ans: Following approaches of software testing

- **Independent testing team**

  An organization may create a separate testing team with independent responsibility of testing

  The team would be having people with sufficient knowledge and ability to test the software

  Advantages:

  Separate test team is supposed to concentrate more on test planning, test strategies and approaches

  There is an independent view about the product

  Disadvantages

  Separate teams means additional cost for an organization. Testing teams needs ramping up and knowledge transfer. Organization needs to keep a check on development and testing team

- **Domain expert doing software testing**

  An organization may employ domain experts for doing testing. Domain experts may use their expertise on subject matter for performing such type of testing

  Advantages:

  Fitness for use. Domain may provide facilitation to developers about defects and customer expectations. Domain expert understands the scenario faced by actual users and hence their testing is realistic

Q2 d. What is test strategy? Explain different stages involve in process of developing test strategy..

Ans: A test strategy defines the project's testing objectives and the means to achieve them. The test strategy therefore determines testing effort and costs. Selecting an appropriate test strategy is one of the most important planning task decisions the test manager has. The goal is to choose a test approach that optimizes the relation between costs of testing and costs of defects.

Steps involved in developing test strategies

Select and rank test factors for given application

Identify the critical test factors for software product under testing. Test factors are analysed and prioritized. The trade of decision must be taken consulting the customer

Identify system development phases and related test factors

The critical success factors have varying importance depending upon the developing life cycle phase. The test approach changes according to factors influencing the life cycle phase

Identify associated risks with each selected test facto in case if it is not achieved

Customer must do trade offs of test factors and poss

Identify phase in which risk if not meeting a test factor need to be addressed

.

Q2 e. Explain gray box testing with its advantages and disadvantages

Ans: Gray box testing

- Gray box testing is done on the basis of internal structures of software as defined by requirements, design, coding standards and guidelines as well as functional and non functional specifications
- Gray box testing combines verification techniques with validation techniques where one can ensure that software is build correctly and also works
- Gray box testing talks about combination of approaches namely black box testing and white box testing at the same time

Advantages of gray box testing:

It checks whether the work product works in correct manner, both functionally as well as structurally

Disavantages of gray box testing:

Knowledge of some automation tools along with their configuration is essential for performing gray box testing

Q2 f. List and explain different testing skills required by tester.

Ans:

- Tester must be selected on the basis of available skills and requirements of tools and its techniques.
- If the testers are conversant with a tool, it can help in easy entry for a new tool in the organization. Users skills are required for all tools used by the testers.
- Programming skills are required when testers have to develop scripts for using the tools for testing. Programming skills are needed to write scripts in specific language

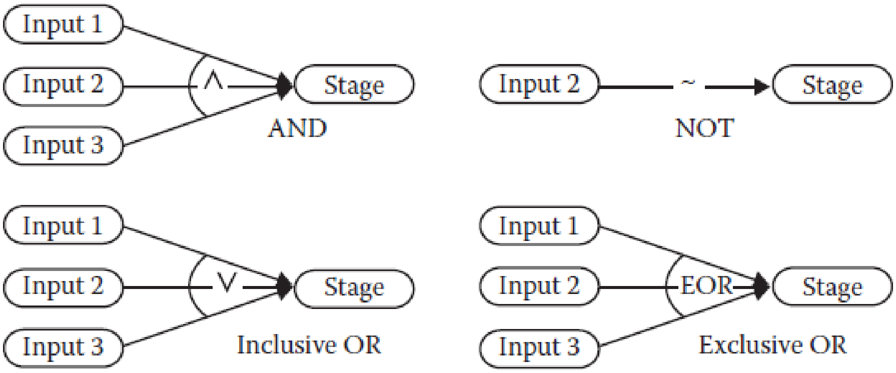
- System skills are required when application tool needs some configuration to be made. Good skills related to systems like working with database, configure and install new tools
- Technical skills are needed to understand the prerequisite of tools and testing. Understanding of user manual, design, requirements are need for usage and troubleshooting

Question 3

Q3 a. What are cause-effect graphs? Explain with the help of an example..

Ans: Cause – effect graph

- Cause and Effect is a tool used to identify possible causes of a problem by representing the relationship between an effect and and its possible cause**.** It is also known as the "fishbone" diagram, this method can be used in brainstorming
- Cause-and-effect graphs works by showing unit inputs on the left side of a drawing, and using AND, OR, and NOT "gates" to express the flow of data across stages of a unit.
- Cause-and-effect graph structure for commission problem is as follow



Q3 b. Define equivalence class. Explain systematic approaches for selecting equivalence.

Ans:

- Equivalence class testing is based on creating partitions. It removes the redundancy gaps that appears in the boundary value analysis.Input or output data is grouped or partitioned into sets of data that is expected to behave similarly using an Equivalence relation.
- An equivalence relation describes how data is going to be processed when it enters a function.
- The equivalence class testing requires to test only one condition from each partition. This is because all the conditions in one partition will be treated in the same way by the software.
- If one condition in a partition works, then all the conditions in that partition will work, and so there is little point in testing any of these others.
- Conversely, if one of the conditions in a partition does not work, then we assume that none of the conditions in that partition will work so again there is little point in testing any more in that partition
- Consider a function of two variables x1,x2 having the following boundaries and intervals within the boundaries:
- a ≤ x1≤ d with interval [a,b)[b,c)[c,d)
- c ≤ x2≤ d with interval [e,f)[f,g)

Q3 c. What is boundary value testing? Explain robust boundary value testing.

Ans: Boundary Value Testing (BVA)

The BVA depends on the concept that errors tend to occur near the extremities of the input variables. Boundary value analysis (BVA) is based on testing at the boundaries between partitions.

Range checking is an example of using the boundary value analysis technique. BVA concentrated more on the boundary of the input space to identify the test cases. Most of the programs can be viewed as a function F. The input variables of F will have some possibly boundaries where a,b and c,d are the range of x1 and x2 respectively.

a ≤ x1≤ b

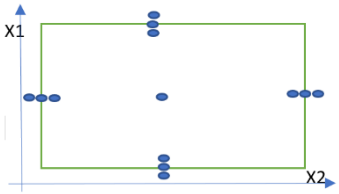
c ≤ x2≤ d

Robust Boundary Value Testing

Robustness testing can be seen as an extension of Boundary Value Analysis.

The idea behind Robustness testing is to test for variables that lie in the legitimate input range and variables that fall just outside this input domain.

Two more values for each variable (min-, max+) are added such that it fall just outside of input range.In addition to 5 testing values (min, min+, nom, max-, max), two more values (min-, max+) are added to fall just outside of the input range



Q3 d. Explain slice-based testing with an example.

Ans: Slice-based testing

Definition:

Given a program P and a set V of variables in P, a slice on the variable set V at statement n, written S(V, n), is the set of all statement fragments in P that contribute to the values of variables in V at node n

P-use used in predicate decision

C-use used in computation decision

O-use used for output

L-use used for location

i-used used for iteration

Example:

• S(price, 5) = {5}

• S(price, 6) = {5, 6, 8, 9}

• S(price, 7) = {5, 6, 8, 9}

• S(price, 8) = {8}

• Lines 1 to 4 have no bearing on the value of the variable at line 7 (and,for that matter, for no other variable at any point), so they are not added to the slice.

• Line 5 contains a defining node of the variable price that can affect the value at line 7, so 5 is added to the slice.

• Line 6 can affect the value of the variable as it can affect the flow of control of the program. Therefore, 6 is added to the slice.

Line 7 is not added to the slice, as it cannot affect the value of the variable at line 7 in any way.

• Line 8 is added to the slice – even though it comes after line 7 in the program listing. This is because of the loop: after the first iteration of the loop, line 8 will be executed before the next

Q3 e. Explain DD-paths and basis path testing.

Ans: DD-path testing

- The best known form of structural testing is based on construct known as decision to decision path
- The reason that program graphs play such an important role in structural testing is because it forms the basis of a number of testing methods, including one based on a construct known as decision-to-decision paths (DD-Paths).
- The idea is to use DD-Paths to create a condensation graph of a piece of software's program graph, in which a number of constructs are collapsed into single nodes known as DD-Paths.
- DD-Paths are chains of nodes in a directed graph that adhere to certain definitions.
- Each chain can be broken down into a different type of DD-Path, the result of which ends up as being a graph of DD-Paths. The length of a chain corresponds to the number of edges that the chain contains.

Basic Path Testing

- The basis is always define in terms of vector space, which is a set of element as well as operations that corresponds to multiplication and addition defined for the vectors.
- The basis of a vector space contains a set of vectors that are independent of one another, and have a spanning property; this means that everything within the vector space can be expressed in terms of the elements within the basis.

Q3 f. Write a note on decision table technique

Ans: Decision Table Technique

- A decision table is a good way to deal with combinations of things (e.g. inputs).
- This technique is sometimes also referred to as a 'cause-effect' table. If different combinations of inputs result in different actions being taken, this can be more difficult to show using equivalence partitioning and boundary value analysis, which tend to be more focused on the user interface.

Advantages

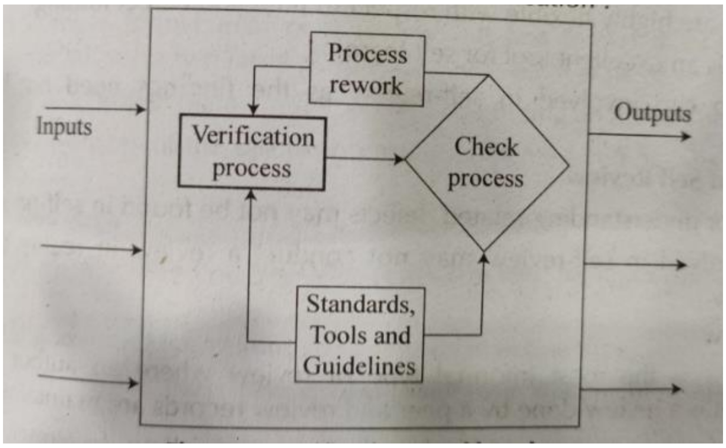
- Decision tables can be used in test design whether or not they are used in specifications, as they help testers explore the effects of combinations of different inputs and other software states that must correctly implement business rules
- Helping the developers do a better job can also lead to better relationships with them
- Decision tables aid the systematic selection of effective test cases and can have the beneficial side-effect of finding problems and ambiguities in the specification.
- It is a technique that works well in conjunction with equivalence partitioning

Question 4

Q4 a. Explain the concept of workbench.

Ans:

The following basic things are required for workbench
Input : There must be some entry check criteria when inputs are entering the workbench and it should match with the output criteria of the earlier work bench
Output: There must be some exit criteria from work bench which should match with input criteria for the next work bench. Output must include review comment



Verification Process: must describe the step by step activities to be conducted in each work bench
Check Process: Describes how the verification process has been checked. Quality plan must describe the objective to be achieved.
Standard tools and guideline: There may be tools, coding, guidelines or standards for verification

Q4 b. List all the methods of verification. Explain all.
Ans: Self Review : It is not an official way of review. One must capture the self review records and defects found in self review to improve the process. It is a kind of self learning and defect prevention method
Peer Review: It is the most informal type of review where an author and a peer are involved. It is a review done by the peer and review records are maintained. The organization defines a checklist for doing peer review
Walkthrough: It is the most formal review but less formal then inspection. It is also known as semi formal as only related people are involved. Some members of a project team are involved in examining an artifact under review
Inspection: It is a very formal way of reviewing the product. A presenter presents a work product and a recorder makes notes of comments given by inspector

Q4 c. Discuss different types of reviews in verification.
Ans:
In-process review : Review conducted during different phases of software development life cycle. The are intended to check whether inputs to the phase are correct or not and whether all the applicable process are followed

Milestone review : It is conducted on periodic basis depending on the completion of a particular phase or a time frame defined or a milestone achieved. These reviews confirms that the output from the phase matches with predefined quality

Phase end review: It is conducted at the end of the development phase under review such as requirement phase, design coding and testing. Waterfall cycle is suitable for doing a phase end review where distinction between different phases are clearly defined

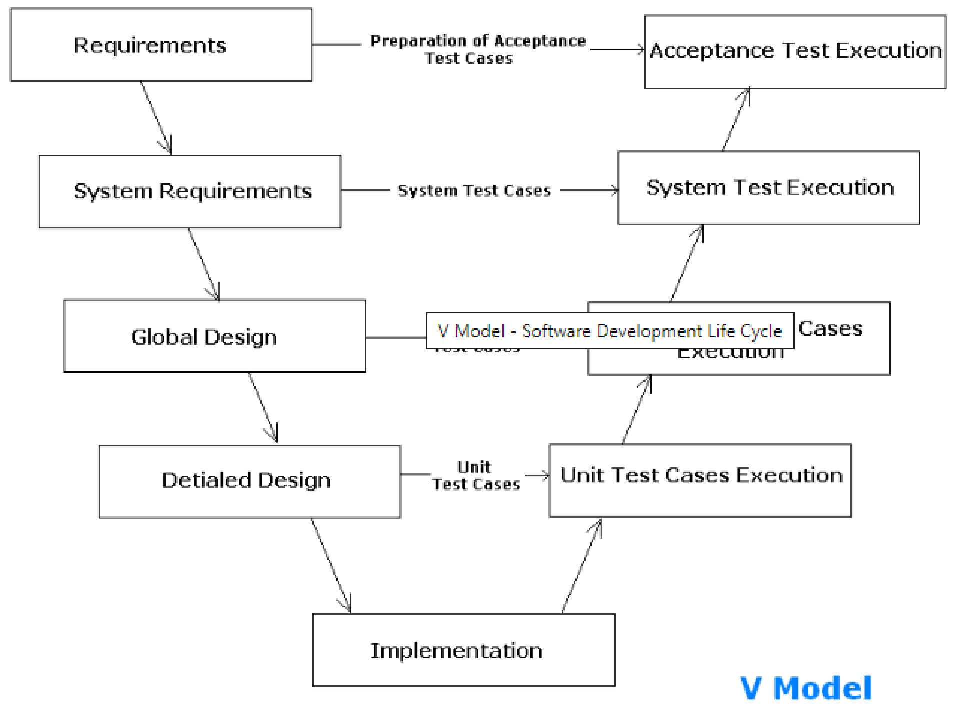
Periodic review: When one phase ends in reality another phase may have half way through. In such cases one must review some on periodic basis such as weekly, monthly and quarterly.

Percent completion review: Percent completion review is a combination of periodic review and phase end review where the project activities or product development activities are assessed on the basis of percent completion

Q4 d. Explain V model for software.

Ans: V model

- Test policy and test strategy for performing verification and validation activities are documented beforehand to avoid any problem in final deliverables
- Testing activities are referred in project plan but they are detailed in quality plan of verification
- Plan of activities should decide about 5 Ws(What, When, Where, Why and Who) and H(How) with respect to people, process, training etc
- Activities prepared and documented should be analyzed for coverage, relationship with different entities, structure and traceability
- Functional test scenario and structured test scenario are developed for design specification.
- High level and low level design must ensure that requirement are completely covered so that software development covers everything
- The output of one phase must match the input criteria of other phase
- The test artifacts that developed must be reviewed and updated



Q4 e. Describe V &V activities during designs.

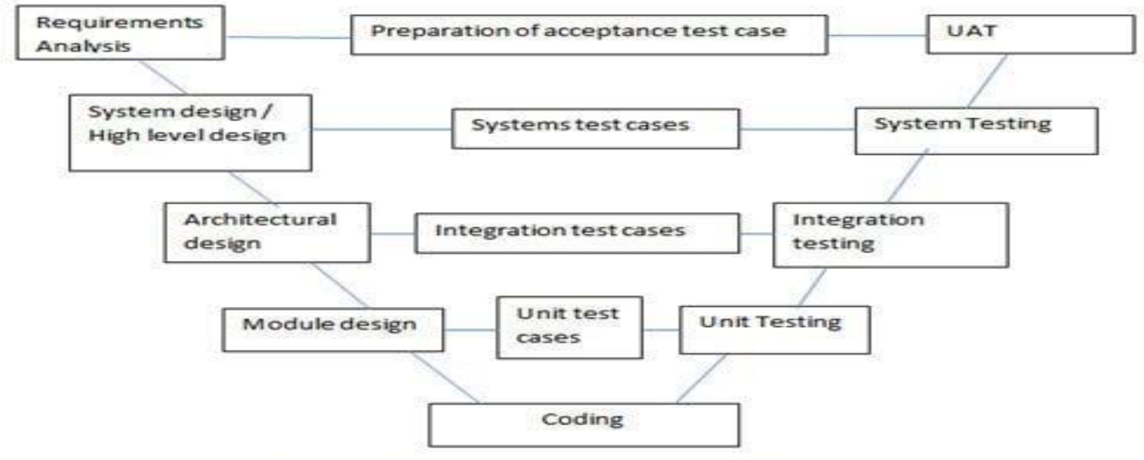
Ans: V & V model

Design may include high level architectural design and low level design or detailed design which are created by architect.

Design Verification: Verification of design may be a walkthrough of design document by design experts team members and stakeholders of the project.
Project team along with architect may walkthrough the design to find the completeness of the give component.
Specific tools or methodologies like UML are used to create design.

Design Validation: Validation of the design can happen at two or more stages during software development life cycle.

The first stage of validation happens when data flow diagram can be created by referring to the design document. If the flow of the data is complete, design is considered to be complete. The second stage happens at integration testing and interface testing.



STLC - V Model - © www.SoftwareTestingHelp.com

Q4 f. Explain different roles and responsibilities of development group.

Ans

Manager :
The development manager selects the objects to be reviewed and confirms that the base documents, as well as the necessary resources, are available. They also choose the participating people.

Moderator
The moderator is responsible for: the administrative tasks pertaining to the review, planning and preparation, ensuring that the review is conducted in an orderly manner and meets its objectives, collecting review data, and issuing the review report.

Author
The author is the creator of the document that is the subject of a review. If several people have been involved in the creation, one person with lead responsibility should be appointed; this person takes over the role of the author.

Reviewer
The reviewers, sometimes also called inspectors, are several (usually a maximum of five) technical experts that shall check the review object after individual preparation.

Recorder
The recorder (or scribe) shall document the findings (problems, action items, decisions, and recommendations) made by the review team. The recorder must be able to record in a short and precise way, capturing the essence of the discussion.

Question 5

Q5a. Explain the characteristic of design testing.

Ans: Clarity: A design must define all functions ,component, tables, stored procedures and reusable components very clearly

Complete: It must define the parameters to be passed/received formats of data handled etc

Traceable: A design must be traceable to requirements. The project manager must check if there is any requirement which does not have corresponding design
Implement: A design must be made in such a way that it can implement easily with selected technology and system
Testable: Testers makes structural test case on the basis of design. Thus a good test design must help in creating structural test case

Q5b. Discuss Bottom up and top down testing with an example..

Ans: Top Down Testing:

- In top-down testing approach, the top level of the application is tested first and then it goes downward till it reaches the final component of the system. All top-level components called by tested components are combined one by one and tested in the process.
- Drivers may not be required as we go downward as earlier phase will act as driver for latter phase while one may have to design stubs to take care of lower-level components which are not available at that time.
- Top-level components are the user interfaces which are created first to elicit user requirements or creation of prototype. Agile approaches like prototyping, formal proof of concept, and test-driven development use this approach for testing.

Bottom Up testing

- It focuses on testing the bottom part/individual units and modules, and then goes upward by integrating tested and working units and modules for system testing.
- It is a mirror image of the Top-down approach, with the difference that stubs are replaced by driver modules that emulate units at the next level up in the tree. In Bottom-top integration, we start with the leaves of the decomposition tree and the test them with specially coded drivers
- Each component & unit is tested first for its correctness. If it found to be working correctly, then only it goes for further integration. It makes a system more robust since individual units are tested & confirmed as working.

Q5c. What is acceptance testing? Explain different forms of it..

Ans: The focus of acceptance testing is on the customer's perspective and judgment.The acceptance test might be the only test the customer is actually involved in or which they can understand. Acceptance tests can even be executed within lower test levels or distributed over several test levels.

Typical forms of acceptance testing include the following:

1. **Testing to determine if the contract has been met:** If customer specific software was developed, the customer (in cooperation with the vendor) will perform acceptance testing according to the contract. On the basis of the results of these acceptance tests the customer considers whether the ordered software system is free of (major) deficiencies and whether the development contract or the service defined by the contract has been accomplished.
2**. User acceptance testing**: Another aspect concerning acceptance as the last phase of validation is the test for user acceptance. Such a test is especially recommended if the customer and the user are different individuals.

3**. Operational (acceptance) testing:** Operational (acceptance) testing assures the acceptance of the system by the system administrators. It may include the testing of backup/restore cycles, disaster recovery, user management, maintenance tasks, and checks of security vulnerabilities.
4. **Field test (alpha and beta testing):** The objective of the field test is to identify influences from users' environments that are not entirely known or that are not specified, and to eliminate them if necessary

Q5d. Explain GUI testing with its advantages and disadvantages.

Ans: GUI Testing

GUI testing is defined as the process of testing the system's Graphical User Interface of the Application Under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars - toolbar, menu bar, dialog boxes, and windows, etc.
GUI is what the user sees. A user does not see the source code. The interface is visible to the user. Especially the focus is on the design structure, images that they are working properly or not.

Advantages:

- Tests the user interface from the users perspective.
- Efficiently reduces the number of risks towards the end of development life cycle.
- Offers developers and testers ease of use and learning.
- Helps validate the compliance of various icons and elements with their design specifications.
- Increases the reliability and improves quality of the product.

Disadvantages

- It requires more memory resources, which leads the system to perform slowly.
- The process of testing is time consuming and may require extra software for running GUIs.
- Since the interface of an application changes frequently, the team might have to refactor recorded test script to improve its accuracy.
- Limited access or no access to the source code makes the process of testing difficult.

Q5e. Write a short note on smoke testing.

Ans: Smoke Testing

- Smoke Testing is also known as "Build Verification Testing", is a type of software testing that comprises of a non-exhaustive set of tests
- It aims at ensuring that the most important functions work. The result of this testing is used to decide if a build is stable enough to proceed with further testing.
- Installation, navigation through the application, invoking or accessing some major functionalities involved in testing.
- If smoke testing fails the user will not be able to work with application and it may result into rejection of application
- Test manager and senior tester performs smoke testing and the entry criteria is depended on smoke testing

Q5 f. Explain compatibility testing in details.

Ans: Compatibility Testing

- Compatibility testing means testing the software on multiple configuration to check the behaviors of different system component and their combination
- The variables can be operating systems, browsers, databases and languages
- The hardware can be machine, server, routers and printers

- Integration with other communication systems can be mailing software, messaging softwares.
- **Friend compatibility**: happens when the application behavior on new plateform is as if it is working on its base platform. Cost benefit analysis has to be done to determine how much friend compatibility is required.
- **Neutral compatibility:** If the application has its own utilities and services and uses them as if nothing has been provided by any platform.
- **Enemy compatibility:** if the application is not compatible with targeted platform then this may be termed as enemy compatibility

----------------------------x--------------------------