STE Assignment 01.

Name : Gaurav Amarnani
Roll no. 21

Q1 List and explain objectives of Software testing.

Ans. Objectives of Software Testing :

1. The main objective of Software Testing is fulfilling customer requirements, so firstly understanding what are customer requirements and then ensuring that software works in such manner.

2. To find defects and issues in the software and fix them before they are encountered by end user.

3. To fulfil the BRS that is, Business requirement specification and SRS, System Requirement specification.

4. Software testing is to assure that the testing is being done correctly and hence, application is ready for use. This helps in gaining the confidence of customers by providing them with quality software.

Q2. Differentiate between Verification & Validation

Ans.

Verification	Validation
1. Checks whether the product is built as per the specified requirements & design specification.	1. It determines whether the software is fit for use and satisfy the business needs.
2. Checks "Are we building the product right"	2. Checks "Are we building the right product"
3. This is done without executing the software	3. Is done with executing the software.
4. Involves all static testing techniques.	4. Involves all dynamic testing techniques

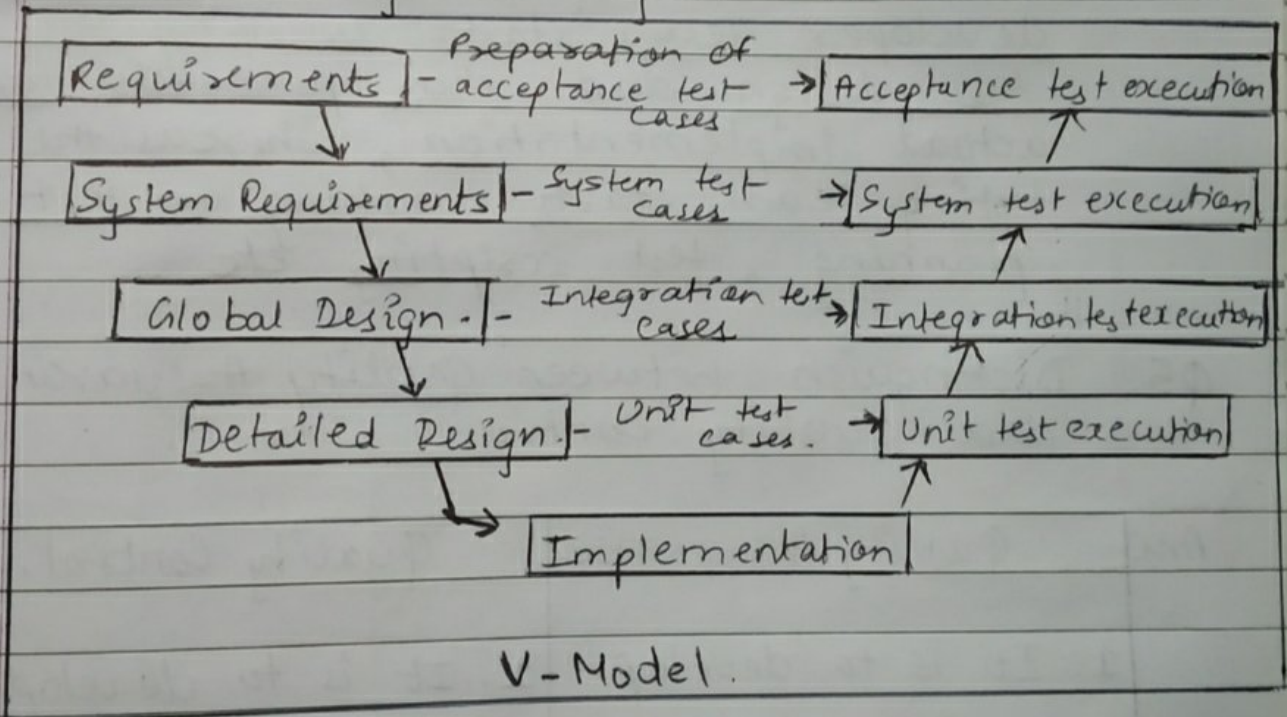Q3. List all the skills of a Software tester.

Ans. Skills required for becoming a good software tester :

1. Communication skills.
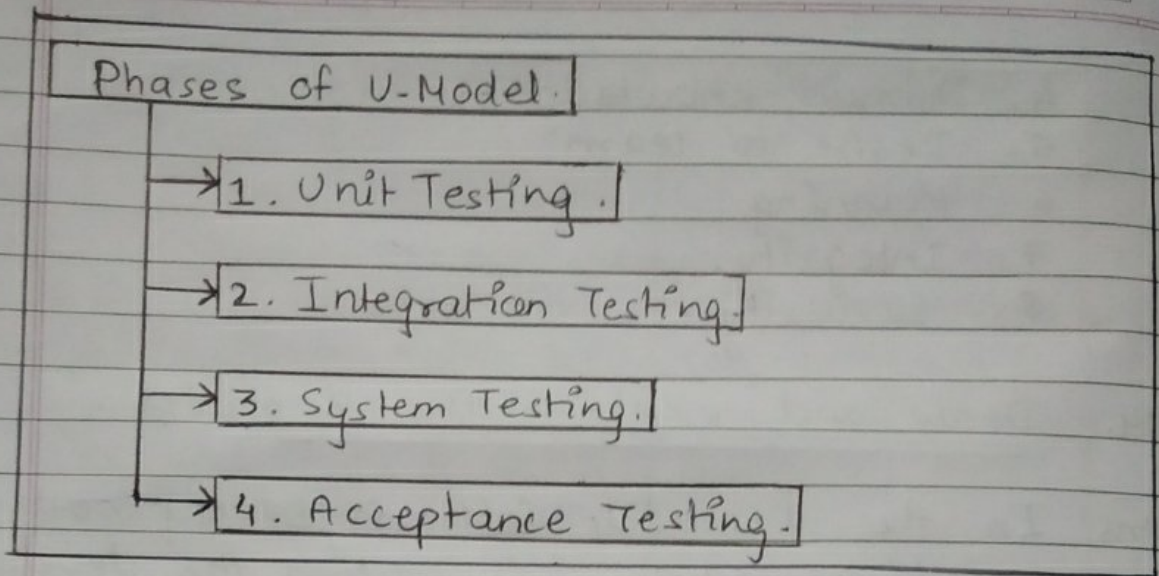2. Technical skills.
3. Analytical skills.

4. Domain knowledge.
5. Desire to learn.
6. Planning.
7. Integrity.
8. Curiosity.

Q4. Draw and explain the V-model.

Ans. In the V model software Development cycle, based on same information the development and testing activity is started.

```
┌──────────────────────────────────────────────────────────────────┐
│                                                                    │
│  ┌──────────────┐  Preparation of                                  │
│  │ Requirements │ - acceptance test → │ Acceptance test execution │ │
│  └──────────────┘     cases                                        │
│         ↓                                                          │
│  ┌──────────────────┐  System test                                │
│  │ System Requirements │ -  cases    → │ System test execution │   │
│  └──────────────────┘                                             │
│            ↓                                                       │
│    ┌──────────────┐  Integration tet                              │
│    │ Global Design │ -  cases      → │ Integration test execution │ │
│    └──────────────┘                                               │
│              ↓                                                     │
│     ┌────────────────┐  Unit test                                 │
│     │ Detailed Design │ -  cases.  → │ Unit test execution │       │
│     └────────────────┘                                            │
│                ↓                                                   │
│              ┌────────────────┐                                   │
│              │ Implementation │                                   │
│              └────────────────┘                                   │
│                                                                    │
│                      V-Model.                                      │
└──────────────────────────────────────────────────────────────────┘
```

Typical V-model shows Software Development activities on the Left hand side of model, whereas actual testing phases are performed on the right hand side.

Phases of U-Model.

→ 1. Unit Testing.

→ 2. Integration Testing.

→ 3. System Testing.

→ 4. Acceptance Testing.

Based on the requirement document, the developer team starts working on the design and then proceeds for actual implementation, whereas the testing team starts working on test planning, test scripting etc.

Q5. Distinguish between Quality Assuarance and quality control.

Ans.

Quality Assuarance	Quality Control.
1. It is to develop good process.	1. It is to develop a good team.
2. QA is for process.	2. QC is for product.
3. It is seperate process.	3. It is part of SDLC.
4. It is preventing defects	4. It is correcting defects.

Q6. Define static testing & Dynamic testing.

Ans. Static testing involves verification, whereas Dynamic testing involves validation. Together they help improve software quality.

1. Static Testing:

Static testing is a verification activity. Testing of program is done without executing of program.

In static testing, code is not run to uncover bugs, but to verify by experts to find any flaw.

2. Dynamic Testing:

Dynamic Testing is a validation activity. Tester has Software Requirement Specifications (SRS) to check whether the product is as per user requirements.

In Dynamic testing, code is executed to find bugs, or to confirm that its working well.

Q7. Define Test case and write five test cases for standard calculator.

Ans. A test case is a <u>documentation</u> which specifies <u>input values</u>, expected output and the <u>preconditions</u> for executing the test.

It is a <u>set of conditions</u> or <u>variables</u> <u>under which a tester will determine</u> <u>whether a system under test satisfies</u> <u>requirements or work correctly.</u>

5 Test Cases for a Standard Calculator:

Test Case Id	Test Case	Steps to be executed	Expected result	Actual result	Pass/fail.
ID_1	Test if number is selected onclick	(i) click on the number/sign.	Number must to selected	Number is selected	Pass.
ID_2	Test if numbers are added correctly	(i) Select a number (ii) select (+) sign (iii) Select another number	Result must be correct	Result is correct	Pass.
ID_3	Test if results is displayed successfully	(i) Select any no. (ii) select any sign (iii) select any other number	Result must be displayed on screen	Result is disp-layed on screen	Pass.

Test ID	Test Case	Steps	Exp Result	Actual Result	Pass/Fail.
ID_4	Test if the system display (e) correctly while multi-plication.	(i) Select a 10 digit no. (ii) Select (x) sign (iii) Select another 10 digit no.	Result no. should have a (e) with it.	Result no. have (e) with it.	Pass.
ID_5	Test if scientific calculations work.	(i) Select a no. (ii) Select a scientific sign (iii) Select another no.	Expected correct result should be displayed	Correct Result is displayed.	Pass.

★ Note : Like ID_2 Test case, we have to perform such more Test cases with other signs like (-, x, ÷) too.

Q8. Explain Inspection, walkthrough and Peer review.

Ans. 1. Inspections:

It is the most formal review type. It is led by trained moderators.

It involves peers to examine the product.

The defects found are documented in a logging list or issue log. A formal follow up is carried out by the moderator applying exit criteria.
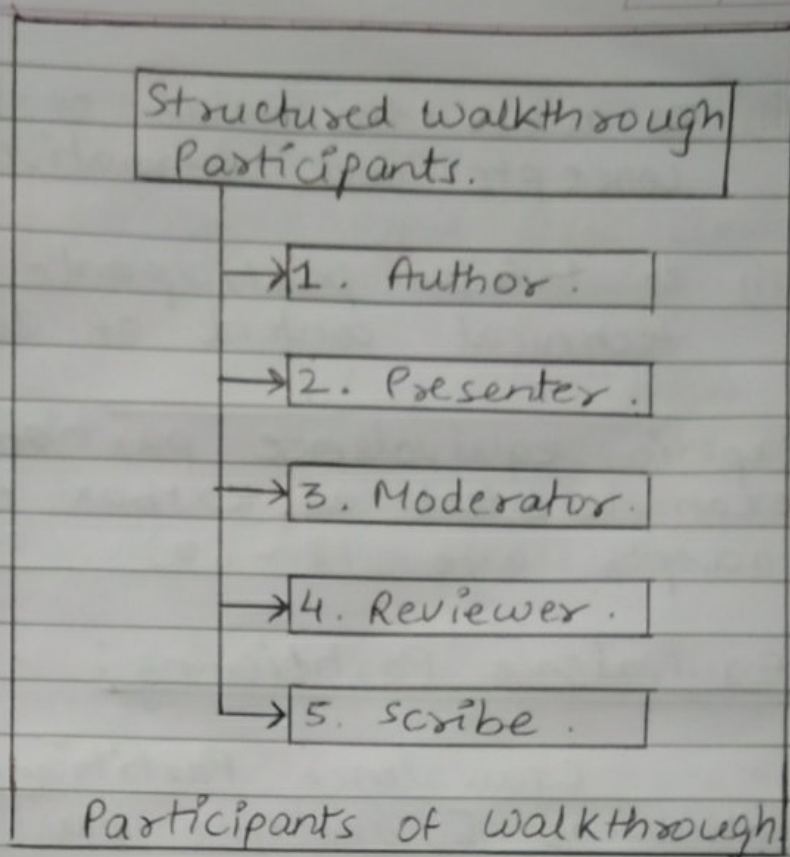
The goals of Inspection :

(i) Improves Product Quality.

(ii) Removes defects as early as possible.

(iii) Create common understanding by exchanging information.

2. Walkthroughs :

A structured walkthrough is a static testing technique performed in an organized manner between a group of peers to review and discuss technical aspects of software development process.

Benefits of walkthrough :

(i) saves time and money.

(ii) Notifies the project management team about progress of development process.

```
┌─────────────────────────────────────┐
│        ┌──────────────────────┐      │
│        │ Structured walkthrough│     │
│        │    Participants.      │      │
│        └──────────┬───────────┘      │
│                   │                   │
│              ┌────┴──────────┐        │
│          →   │ 1. Author.    │        │
│              └───────────────┘        │
│              ┌───────────────┐        │
│          →   │ 2. Presenter. │        │
│              └───────────────┘        │
│              ┌───────────────┐        │
│          →   │ 3. Moderator. │        │
│              └───────────────┘        │
│              ┌───────────────┐        │
│          →   │ 4. Reviewer.  │        │
│              └───────────────┘        │
│              ┌───────────────┐        │
│          →   │ 5. Scribe.    │        │
│              └───────────────┘        │
│                                       │
│      Participants of walkthrough      │
└───────────────────────────────────────┘
```

3. Peer Review :

It is less formal review. It is
led by trained moderator but can
also be led by a technical expert.

Defects are found by the experts
who focus on the content of document.
In practice, peer reviews vary from
quite formal to very formal.

✗ Goals of Peer Review :

(i) To ensure that technical concepts
are used correctly.

(ii) To access the value of technical concepts and alternatives in product.

(iii) To inform participants about the technical content of document.

Q9. Explain equivalence partitioning for example (test a textbox field which accepts age (8 - 60).

Ans. **Equivalence Partitioning:**

Equivalence Partitioning also called as Equivalence Class Partitioning. It is abbreviated as ECP.

It is a software testing technique that divides the input test data of the application under test into each partition at least once of equivalent data from which test cases can be derived.

An advantage of this approach is it reduces the time required for performing testing of software due to less number of test cases.

For Example :

1. The below example best describes
   the equivalence partitioning.

   - Assume that the application accepts
     integers in the range 18 to 60

   - Valid Equivalence Class Partition :
     18 to 60 inclusive.

   - Invalid Equivalence Class Partition :
     less than 18 , more than 60 ,
     decimal numbers and non-numeric
     numbers.

2. For diploma result analysis :

   we can identify 1 valid equivalence
   Partition and 2 invalid partition
   as shown below :

Invalid Partition	Valid Partition	Invalid Partition
Age 0 - 17	Age 18 - 60	Age 60 - 100

Q10. Define Boundary value analysis with example.
(Password field that accepts 6 character minimum and 10 characters maximum)

Ans. <u>Boundary Value Analysis</u> :

      <u>Boundary value analysis</u> (BVA) is also called <u>boundary condition data testing</u> and is based on testing at the boundaries between partitions. Values on the <u>minimum and maximum edges</u> of Equivalence Partition are tested. Since these boundaries are common locations for error that result in software faults they are frequently exercised in test cases.

<u>For Example</u> :

If we have a password field which can intake 6 characters min and 10 characters max,

Invalid Partition	Valid Partition	Invalid Partition
5	6 - 10	11 and above