Testing

Testing is the process or activity that checks the functionality and correctness of software according to specified user requirements in order to improve the quality and reliability of system. It is an expensive, time consuming, and critical approach in system development which requires proper planning of overall testing process.

Types of Testing

Testing can be of various types and different types of tests are conducted depending on the kind of bugs one seeks to discover –

1. Unit testing

A unit test is the smallest testable part of an application like functions, classes, procedures, interfaces. Unit testing is a method by which individual units of source code are tested to determine if they are fit for use. Unit tests are basically written and executed by software developers to make sure that code meets its design and requirements and behaves as expected. Unit testing is basically done before integration

Unit Testing Techniques:

- ❖ **Black Box Testing -** Black-box testing is a method of software testing that examines the functionality of an application based on the specifications.

  Behavioral Testing Techniques:

  There are different techniques involved in Black Box testing.

  - Equivalence Class
  - Boundary Value Analysis
  - Domain Tests
  - Orthogonal Arrays
  - Decision Tables
  - State Models
  - Exploratory Testing
  - All-pairs testing
- ❖ **White Box Testing -** White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code.

White Box Testing Techniques:

- **Statement Coverage -** This technique is aimed at exercising all programming statements with minimal tests.
- **Branch Coverage -** This technique is running a series of tests to ensure that all branches are tested at least once.
- **Path Coverage -** This technique corresponds to testing all possible paths which means that each statement and branch is covered.

❖ **Gray Box Testing -** Grey Box testing is testing technique performed with limited information about the internal functionality of the system.

Gray-box testing Techniques:

- Regression testing
- Pattern Testing
- Orthogonal array testing
- Matrix testing

2. Integration Testing

After the unit testing has been done with satisfaction for each component or script, the integration testing is started to ensure the CW-MSLD System components worked together smoothly. In Integration Testing, the analyst tests multiple modules working together. The functional and non-functional requirements were tested in this stage.

3. Functional Testing

Function testing determines whether the system is functioning correctly according to its specifications and relevant standards documentation. Functional testing typically starts with the implementation of the system, which is very critical for the success of the system.

Functional testing is divided into two categories −

- **Positive Functional Testing −** It involves testing the system with valid inputs to verify that the outputs produced are correct.
- **Negative Functional Testing** − It involves testing the software with invalid inputs and undesired operating conditions.

4. System Testing

System testing is the type of testing to check the behavior of a complete and fully integrated software product based on the software requirements specification (SRS) document.

This is black box type of testing where external working of the software is evaluated with the help of requirement documents & it is totally based on Users point of view.

5. **Acceptance testing:** This is a final testing, if client is satisfied, sign for acceptance. If clients get errors or problems, developer has to modify again the system.

Verification: The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.

Verification is a static practice of verifying documents, design, code and program. It includes all the activities associated with producing high quality software: inspection, design analysis and specification analysis. It is a relatively objective process.

Methods of Verification:

- Static Testing
- Walkthrough
- Inspection
- Review

Validation: The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements.

Validation is the process of evaluating the final product to check whether the software meets the customer expectations and requirements. It is a dynamic mechanism of validating and testing the actual product.

Methods of Validation:

- Dynamic Testing
- Testing
- End Users

Implementation

Implementation is a process of ensuring that the information system is operational. It involves −

- Constructing a new system from scratch
- Constructing a new system from the existing one.

Implementation allows the users to take over its operation for use and evaluation. It involves training the users to handle the system and plan for a smooth conversion.

Implementation Planning

The Implementation Planning describes how the information system will be deployed, installed and transitioned into an operational system. The plan contains an overview of the system, a brief description of the major tasks involved in the implementation, the overall resources needed to support the implementation effort (such as hardware, software. facilities, materials, and personnel), and any site-specific implementation requirements. The plan is developed during the Design Phase and is updated during the Development Phase; the final version is provided in the Integration and Test Phase and is used for guidance during the Implementation Phase. The outline shows the structure of the Implementation Plan.

Conversion

It is a process of migrating from the old system to the new one. It provides understandable and structured approach to improve the communication between management and project team.

Conversion Plan

It contains description of all the activities that must occur during implementation of the new system and put it into operation. It anticipates possible problems and solutions to deal with them.

Conversion Methods

The four methods of conversion are −

Parallel Conversion: Old and new systems are used simultaneously.

Direct Cutover Conversion: New system is implemented and old system is replaced completely.

Pilot Approach: Supports phased approach that gradually implements system across all users

Phase-In Method: Working version of system implemented in one part of organization based on feedback, it is installed throughout the organization all alone or stage by stage.

Input Design

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc. Therefore, the quality of system input determines the quality of system output.

A well designed input forms and screens have following properties −

- It should serve specific purpose effectively such as storing, recording, and retrieving the information.
- It ensures proper completion with accuracy.
- It should be easy to fill and straightforward.
- It should focus on user’s attention, consistency, and simplicity.

Objectives for Input Design

- To design data entry and input procedures
- To reduce input volume
- To design source documents for data capture or devise other data capture methods
- To design input data records, data entry screens, user interface screens, etc.
- To use validation checks and develop effective input controls.

Data Input Methods

It is important to design appropriate data input methods to prevent errors while entering data. These methods depend on whether the data is entered by customers in forms manually and later entered by data entry operators, or data is directly entered by users on the PCs.

Some of the popular data input methods are −

- Batch input method (Offline data input method)
- Online data input method
- Computer readable forms
- Interactive data input

Input Integrity Controls

Input integrity controls include a number of methods to eliminate common input errors by end-users. They also include checks on the value of individual fields; both for format and the completeness of all inputs.

<u>Output Design</u>

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

Objectives of Output Design

- To develop output design that serves the intended purpose and eliminates the production of unwanted output.

- To develop the output design that meets the end users requirements.
- To deliver the appropriate quantity of output.
- To form the output in appropriate format and direct it to the right person.
- To make the output available on time for making good decisions.

Various types of outputs −

External Outputs: Manufacturers create and design external outputs for printers. External outputs enable the system to leave the trigger actions on the part of their recipients or confirm actions to their recipients.

Internal outputs: Internal outputs are present inside the system, and used by end-users and managers. They support the management in decision making and reporting.

There are three types of reports produced by management information −

Detailed Reports − They contain present information which has almost no filtering or restriction generated to assist management planning and control.

Summary Reports − They contain trends and potential problems which are categorized and summarized that are generated for managers who do not want details.

Exception Reports − They contain exceptions, filtered data to some condition or standard before presenting it to the manager, as information.

Output Integrity Controls

Output integrity controls include routing codes to identify the receiving system, and verification messages to confirm successful receipt of messages that are handled by network protocol.

Forms Design

Both forms and reports are the product of input and output design and are business document consisting of specified data. The main difference is that forms provide fields for data input but reports are purely used for reading. For example, order forms, employment and credit application, etc.

Objectives of Good Form Design

A good form design is necessary to ensure the following −

- To keep the screen simple by giving proper sequence, information, and clear captions.
- To meet the intended purpose by using appropriate forms.
- To ensure the completion of form with accuracy.

  - To keep the forms attractive by using icons, inverse video, or blinking cursors etc.
  - To facilitate navigation.

Types of Forms

Flat Forms

- It is a single copy form prepared manually or by a machine and printed on a paper. For additional copies of the original, carbon papers are inserted between copies.
- It is a simplest and inexpensive form to design, print, and reproduce, which uses less volume.

Unit Set/Snap out Forms

- These are papers with one-time carbons interleaved into unit sets for either handwritten or machine use.
- Carbons may be either blue or black, standard grade medium intensity. Generally, blue carbons are best for handwritten forms while black carbons are best for machine use.

Continuous strip/Fanfold Forms

- These are multiple unit forms joined in a continuous strip with perforations between each pair of forms.
- It is a less expensive method for large volume use.

No Carbon Required (NCR) Paper

- They use carbonless papers which have two chemical coatings (capsules), one on the face and the other on the back of a sheet of paper.
- When pressure is applied, the two capsules interact and create an image.

CRT Screen Design

Many online data entry devices are CRT screens that provide instant visual verification of input data. The operator can make any changes desired before the data go to the system for processing CRT screen is actually a display station that has a buffer for storing data. A common size display is 24 rows of 80 characters each or 1920 character.

There are two methods for entering data on CRT screens

1. Manual method: The manual method uses a worksheet similar to a printout chart. The menu or the data to be displayed is blocked out in the reserved areas.

2. Software utility method: This method designs screen layouts through software utility.

Procedure for Hardware /Software Selection

Selecting hardware and software for implementing information system in an organization is a serious and time-consuming process that passes through several phases. The main steps of the selection process are listed below:

Major phases in selection

The selection process should be viewed as a project, and a project team should be organized with management support. Several steps make up the selection process.

1. Requirements analysis

2. System specifications

3. Request for proposal (RFP)

4. Evaluation and validation

5. Vendor selection

6. Post-installation review

Software selection: Software selection is a critical aspect for system development. There are 2 ways of acquiring the software.

- Custom -made
- Packages

Criteria for Software selection:

Reliability – It is the probability that the software will executed in a specific period of time without any failures.

Functionality – It is the definition of the facilities, performance and other factors that the user requires in the finished product.

Capacity – Capacity refers to the capability of the software package to handle the users requirements for size of files, number of data elements, and reports.

Flexibility – It is a measure of effort required to modify an operational program.

Usability – This criteria refers to the effort required to operate, prepare the input, and interpret the output of a program.

Security – It is a measure of the likelihood that a system's user can accidentally or intentionally access or destroy unauthorized data.

Performance – It is a measure of the capacity of the software package to do what it is expected to do.

Serviceability –This criteria focuses on documentation and vendor support.

Ownership – Who owns the software ,and to consider whether he has the right to access the software, or he can sell or modify the software.

Minimal costs – Cost is a major consideration in deciding between in-house and vendor software.

Hardware Selection Criteria

- Hardware must support current software as well as software planned for procurement over the next planning interval.
- Hardware must be compatible with existing or planned networks
- Hardware must be upgradeable and expandable to meet the needs of the next planning interval
- Hardware warranties must be of an appropriate length
- Hardware maintenance must be performed by [local/remote vendor, in-house personnel]
- Whenever feasible, hardware standards will dictate procurement of like brands and configurations to simplify installation and support
- Routine assessments of installed infrastructure will feed an upgrade/replace decision process

The Make or Buy Decision

The make-or-buy decision is the act of making a strategic choice between producing an item internally (in-house) or buying it externally (from an outside supplier). In a make-or-buy decision, the most important factors to consider are part of quantitative analysis, such as the associated costs of production and whether the business has the capacity to produce at required levels. Make or buy decision is always a valid concept in business.

Four Numbers You Should Know

When you are supposed to make a make-or-buy decision, there are four numbers you need to be aware of. Your decision will be based on the values of these four numbers. Let's have a look at the numbers now. They are quite self-explanatory.

- The volume

- The fixed cost of making
- Per-unit direct cost when making
- Per-unit cost when buying

Reasons for Making

There are number of reasons a company would consider when it comes to making in-house. Following are a few:

- Cost concerns
- Desire to expand the manufacturing focus
- Need of direct control over the product
- Intellectual property concerns
- Quality control concerns
- Supplier unreliability
- Lack of competent suppliers
- Volume too small to get a supplier attracted
- Reduction of logistic costs (shipping etc.)
- To maintain a backup source
- Political and environment reasons
- Organizational pride

Reasons for Buying

Following are some of the reasons companies may consider when it comes to buying from a supplier:

- Lack of technical experience
- Supplier's expertise on the technical areas and the domain
- Cost considerations
- Need of small volume
- Insufficient capacity to produce in-house
- Brand preferences
- Strategic partnerships

Maintenance

Maintenance is fundamental service for the manufacturing industry to increase the performance of production facilities from both economic and environmental perspectives. The results obtained from the evaluation process help the organization to determine whether its information systems are effective and efficient or otherwise. The process of monitoring, evaluating, and modifying of existing information systems to make required or desirable improvements may be termed as System Maintenance. System maintenance is an ongoing activity, which covers a wide variety of activities, including removing program and design errors, updating documentation and test data and updating user support. For the purpose of convenience, maintenance may be categorized into three classes, namely:

i) Corrective,

ii) Adaptive, and

iii) Perfective.

Documentation

Documentation is one of the systems which is used to communicate, instruct and record the information for any reference or operational purpose. They are very useful for representing the formal flow of the present system. It is important that prepared document must be updated on regular basis to trace the progress of the system easily.

After the implementation of system if the system is working improperly, then documentation helps the administrator to understand the flow of data in the system to correct the flaws and get the system working.

Importance of Documentation

- It can reduce system downtime, cut costs, and speed up maintenance tasks.
- It provides the clear description of formal flow of present system and helps to understand the type of input data and how the output can be produced.
- It provides effective and efficient way of communication between technical and nontechnical users about system.
- It facilitates the training of new user so that he can easily understand the flow of system.
- It helps the user to solve the problems such as troubleshooting and helps the manager to take better final decisions of the organization system.
- It provides better control to the internal or external working of the system.

Types of Documentations

When it comes to System Design, there are following four main documentations −

Program Documentation

- It describes inputs, outputs, and processing logic for all the program modules.

- The program documentation process starts in the system analysis phase and continues during implementation.
- This documentation guides programmers, who construct modules that are well supported by internal and external comments and descriptions that can be understood and maintained easily.

Operations Documentation

Operations documentation contains all the information needed for processing and distributing online and printed output. Operations documentation should be clear, concise, and available online if possible.

It includes the following information −

- Program, systems analyst, programmer, and system identification.
- Scheduling information for printed output, such as report, execution frequency, and deadlines.
- Input files, their source, output files, and their destinations.
- E-mail and report distribution lists.
- Special forms required, including online forms.
- Error and informational messages to operators and restart procedures.
- Special instructions, such as security requirements.

User Documentation

It includes instructions and information to the users who will interact with the system. For example, user manuals, help guides, and tutorials. User documentation is valuable in training users and for reference purpose. It must be clear, understandable, and readily accessible to users at all levels.

The users, system owners, analysts, and programmers, all put combined efforts to develop a user's guide.

A user documentation should include −

- A system overview that clearly describes all major system features, capabilities, and limitations.
- Description of source document content, preparation, processing, and, samples.
- Overview of menu and data entry screen options, contents, and processing instructions.
- Examples of reports that are produced regularly or available at the user's request, including samples.
- Security and audit trail information.

- Explanation of responsibility for specific input, output, or processing requirements.
- Procedures for requesting changes and reporting problems.
- Examples of exceptions and error situations.
- Frequently asked questions (FAQs).
- Explanation of how to get help and procedures for updating the user manual.

System Documentation

System documentation serves as the technical specifications for the IS and how the objectives of the IS are accomplished. Users, managers and IS owners need never reference system documentation. System documentation provides the basis for understanding the technical aspects of the IS when modifications are made.

- It describes each program within the IS and the entire IS itself.
- It describes the system’s functions, the way they are implemented, each program's purpose within the entire IS with respect to the order of execution, information passed to and from programs, and overall system flow.
- It includes data dictionary entries, data flow diagrams, object models, screen layouts, source documents, and the systems request that initiated the project.
- Most of the system documentation is prepared during the system analysis and system design phases.
- During systems implementation, an analyst must review system documentation to verify that it is complete, accurate, and up-to-date, and including any changes made during the implementation process.

Disaster Recovery Plan (DRP)

A disaster recovery plan (DRP) is a documented, structured approach with instructions for responding to unplanned incidents.

This step-by-step plan consists of the precautions to minimize the effects of a disaster so the organization can continue to operate or quickly resume mission-critical functions. Typically, disaster recovery planning involves an analysis of business processes and continuity needs.

Recovery strategies

Recovery strategies define an organization's plans for responding to an incident, while disaster recovery plans describe how the organization should respond.

In determining a recovery strategy, organizations should consider such issues as:

- Budget

- Resources -- people and physical facilities
- Management's position on risks
- Technology
- Data
- Suppliers

Disaster recovery planning steps

The disaster recovery plan process involves more than simply writing the document.

In advance of the writing, a risk analysis and business impact analysis help determine where to focus resources in the disaster recovery planning process.

Disaster recovery plans are living documents. Involving employees -- from management to entry-level -- helps to increase the value of the plan.

Creating a disaster recovery plan

An organization can begin its DR plan with a summary of vital action steps and a list of important contacts, so the most essential information is quickly and easily accessible.

The plan should define the roles and responsibilities of disaster recovery team members and outline the criteria to launch the plan into action. The plan then specifies, in detail, the incident response and recovery activities.