# User interface design

**User interface (UI) design** or **user interface engineering** is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing usability and the user experience. In computer or software design, user interface (UI) design primarily focuses on information architecture. It is the process of building interfaces that clearly communicates to the user what's important. UI design refers to graphical user interfaces and other forms of interface design. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals (user-centered design).

User interfaces are the points of interaction between users and designs. There are three types:

- Graphical user interfaces (GUIs) - Users interact with visual representations on a computer's screen. The desktop is an example of a GUI.
- Interfaces controlled through voice - Users interact with these through their voices. Most smart assistants, such as Siri on smartphones or Alexa on Amazon devices, use voice control.
- Interactive interfaces utilizing gestures- Users interact with 3D design environments through their bodies, e.g., in virtual reality (VR) games.

Interface design is involved in a wide range of projects, from computer systems, to cars, to commercial planes; all of these projects involve much of the same basic human interactions yet also require some unique skills and knowledge. As a result, designers tend to specialize in certain types of projects and have skills centered on their expertise, whether it is software design, user research, web design, or industrial design.

Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to itself. Graphic design and typography are utilized to support its usability, influencing how the user performs certain interactions and improving the aesthetic appeal of the design; design aesthetics may enhance or detract from the ability of users to use the functions of the interface. The design process must balance technical functionality and visual elements (e.g., mental model) to create a system that is not only operational but also usable and adaptable to changing user needs.

## Best Practices for Designing an Interface

Everything stems from knowing your users, including understanding their goals, skills, preferences, and tendencies. Once you know about your user, make sure to consider the following when designing your interface:

- **Keep the interface simple and consistent.**
- **Be purposeful in page layout.**
- **Strategically use color and texture.**
- **Use typography to create hierarchy and clarity.** Carefully consider how you use typeface.
- **Make sure that the system communicates what's happening.**
- **Think about the defaults.**
- **Enable frequent users to use short-cuts -**
- **Offer informative feedback**
- **Design dialog to yield closure** - Sequences of actions should be organized into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and this indicates that the way ahead is clear to prepare for the next group of actions.

- **Offer simple error handling** - As much as possible, design the system so the user will not make a serious error. If an error is made, the system should be able to detect it and offer simple, comprehensible mechanisms for handling the error.
- **Permit easy reversal of actions** - This feature relieves anxiety, since the user knows that errors can be undone. Easy reversal of actions encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data entry, or a complete group of actions.
- **Support internal locus of control** - Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Design the system to make users the initiators of actions rather than the responders.
- **Reduce short-term memory load** - The limitation of human information processing in short-term memory requires the displays to be kept simple, multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequences of actions.

## 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. 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.
- All these objectives are obtained using the knowledge of basic design principles regarding −
    - What are the inputs needed for the system?
    - How end users respond to different elements of forms and screens.

## Objectives for Input Design

The objectives of input design are −

- 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. A system should prevent user from making mistakes by −

- Clear form design by leaving enough space for writing legibly.
- Clear instructions to fill form.
- Clear form design.
- Reducing key strokes.
- Immediate error feedback.

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.

Audit trails for data entry and other system operations are created using transaction logs which gives a record of all changes introduced in the database to provide security and means of recovery in case of any failure.

# Output Design

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

The objectives of input design are –

- 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.

Let us now go through 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.

Some of the external outputs are designed as turnaround outputs, which are implemented as a form and re-enter the system as an input.

## 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.

Printed or screen-format reports should include a date/time for report printing and the data. Multipage reports contain report title or description, and pagination. Pre-printed forms usually include a version number and effective date.

## 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.

- During form designing, the designers should know –
  - who will use them
  - where would they be delivered
  - the purpose of the form or report
- During form design, automated design tools enhance the developer's ability to prototype forms and reports and present them to end users for evaluation.

## 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.

## GUI Implementation Tools

There are several tools available using which the designers can create entire GUI on a mouse click. Some tools can be embedded into the software environment (IDE). GUI implementation tools provide powerful array of GUI controls. For software customization, designers can change the code accordingly. There are different segments of GUI tools according to their different use and platform.

## Example

Mobile GUI, Computer GUI, Touch-Screen GUI etc. Here is a list of few tools which come handy to build GUI:

- FLUID
- AppInventor (Android)
- LucidChart
- Wavemaker
- Visual Studio

## Physical File Design & Data Base Design

A database is a collection of information that is organized so that it can be easily accessed, managed and updated. Data is organized into rows, columns and tables, and it is indexed to make it easier to find relevant information. Data gets updated, expanded and deleted as new information is added. Databases process workloads to create and update themselves, querying the data they contain and running applications against it. A database is a model of a thing in the real world. Like their physical model counterparts, data models enable you to get answers about the facts that make up the objects being modelled. Database design is the craft of relating things in the real world to data on a computer under the constraints and affordances of computer technology (read/write; supported data types, storage and access). Tellingly, popular database data types include Booleans, strings, numbers, time – but not one explicitly for money. In addition to its use (in many forms) in what was once known as new media, databases pervade many parts of the postmodern condition (electronic voting, banking records, known and unknown government "collection lists", etc.). The strengths and danger of databases lies in the ability to selectively access and combine entries from large amounts of structured data. Furthermore, relationships between data entries can be found (via queries) in a way that reveals additional information.

Database Management System (DBMS) is a collection of interrelated data. It consists of a set of programs to access the data. DBMS contains information about a particular enterprise. It provides an environment that is both convenient and efficient to use. Some of the Database Applications are as following:

- Banking: all transactions
- Airlines: reservations, schedules
- Universities: Registration, grade
- Manufacturing: production, inventory, order, supply chain.
- Sales: customers, products and purchases

**Advantages of File System**

- Easy to understand.
- Easy to implement.
- Less hardware and software requirements.
- Less Skills set are required to hand flat database systems.
- Best for small databases.

**Disadvantages of File System**

- Less security easy to extract information.
- Data Inconsistency
- Redundancy
- Sharing of information is cumbersome task
- Slow for huge database
- Searching process is time consuming
- Be Sociable, Share!

**Difference between DBMS and File System**

• In File System, files are used to store data while, collections of databases are utilized for the storage of data in DBMS.
• Although File System and DBMS are two ways of managing data, DBMS clearly has many advantages over File Systems. Typically when using a File System, most tasks such as storage, retrieval and search are done manually and it is quite tedious whereas a DBMS will provide automated methods to complete these tasks.
• Using a File System will lead to problems like data integrity, data inconsistency and data security, but these problems could be avoided by using a DBMS.
• Unlike File System, DBMS are efficient because reading line by line is not required and certain control mechanisms are in place.

## Design of Physical Records

A physical record is a group of fields stored in adjacent memory locations and are retrieved together as a unit.They have fixed Length and variable fields. A Record is a collection of fields. Records are common to both databases and files. Records are collection of fields in a predefined format. The design of physical record involves putting the collection of fields in a single logical unit so that the fields are stored in adjacent locations for better storage and retrieval. The main objective of the design of physical records is to store and retrieve them efficiently. Also, the fields should be stored in adjacent locations in such a way that the storage is used efficiently and speed of data processing is appropriate. Physical pages or blocks are units of information moved between disk and memory buffers. They hold not only records, or table entries, but other information such as the amount of free space currently available in the block, the starting position of each record, etc. Blocks of data (pages) are normally read or written by the operating system. Page is referred to as the amount of data written in one I/O operation of operating system. Blocking factor refers to the number of physical records per page. If a record size is 1340 bytes and the page size is 2048 bytes, then 708 bytes are wasted if DBMS does not allow physical records to span different pages. Selecting a block size involves a trade-off. In principle, the larger the block size, the fewer read-write operations need be performed to access a file by the operating system and therefore the more efficient is the processing. However, it requires a correspondingly large allocation of buffer space in memory. Since this is limited (and perhaps shared by many users), there is in practice, an upper bound. Moreover, large block sizes are primarily advantageous for sequential access.

Denormalization is the process of transforming normalized relations into unnormalized physical record specifications. The motivation behind denormalization is poor performance of normalized table. The following may be of use for denormalization.
• Combine two entities with one-to-one relationship to one entity. This avoids the cost of joining two tables when the data are required from both the tables.
• Another form of de-normalization is to repeat the non key attribute (field) of one table in another table to facilitate the execution of query faster. However, it depends on the application at hand.

| Customer_ID | Customer_Name | Address | City | State | Zip | Order_ID | Order_Date |
|---|---|---|---|---|---|---|---|
| 1001 | Raj | Flat 102 | Patna | Bihar | 8004 | 0342 | 04/01/2012 |
| 3002 | Abhi | Flat 202 | Gaya | Bihar | 80011 | 0441 | 09/02/2012 |

Figure: Depicts denormalization for optimized query processing

In a particular application it is seen that queries about order also require the cutomer_name. In case of normalized table, this would always require joining Customer table and order table each time the query is processed. We have modified the order table by adding back the

customer_name from the customer table in order table. Now all queries will require only the order table as all relevant information are available in this table.

Activities to enhance performance

1. Combining tables to avoid joins
2. Horizontal partitioning refers to placing different rows of a table into separate files. For example, in an order table order pertaining to different regions can be kept in a separate table for efficient retrieval of records.
3. Vertical partitioning refers to placing different columns of a table into separate files by repeating the primary key in each of the files.
4. Record partitioning refers to a combination of both horizontal and vertical partitioning as in distributed database processing
5. Data replication refers to the same data being stored in multiple places in the database.

Process of Denormalization of Tables

- Select the dominant process based on frequency of execution and frequency of data access;
- Define the join table for dominant process;
- Evaluate the cost of query, updates and storage for the schema. Consider possibility of renormalization to avoid table joins.

Fixed length records and variable length records

In fixed length record format, the length of record is always the same. The fixed length records are easier to design and implement but are more wasteful than variable length record formats when comes to utilization of space. In variable length record format, the records are of variable length. To identify the end of a record, variable length records use end of record marker. It can be a special character.

**Design of Physical Files**

As we know that Computers are used for storing the information for a permanent time or the Files are used for storing the Data of the users for a Long time Period. The files can contains any type of information means they can Store the text, any Images or Pictures or any data in any Format. So that there must be some mechanism which can be used for storing the information, accessing the information and also performing Some Operations on the files. Files are collection of logically connected records. In RDBMS, files are called tables. However, the way the files are stored in memory depend on the operating system. Many operating systems allow files to be split into pieces but the same is transparent to the user.

**Types of Files**

1. Master file This type of file holds descriptive data; the actual data that is supposed to be processed and holds the resultant data after the process is completed (ex. names, addresses, sales, etc.). The data can be organized using keys.
2. Transaction file It contains the transactions; changes that are supposed to be made to the data in the master file.

- In batch processing all transactions are collected in the transaction file and the changes are applied to the master file sequentially in a single pass. For this to be possible, both the master and transaction file have to be sorted first.
- In an online system the changes are applied to the master file the moment the transactions occur or are recorded.

Algorithms to update the Master file Algorithm 1

1. Order transaction file by key.
2. Create new master file.
3. Compare first keys in master file to first key in transaction file.
4. Continue this until end of both master and transaction file.

Algorithm 2

1. Apply any changes defined by the transaction file to the record.
2. Write the record with the lowest key to the new master file.

3. Go to the next record in the original master file and the transaction file.
4. Compare keys
5. Close all files.

3. Archive file An Archive file is a file of data in permanent storage (usually, the data is stored for legal reasons or to perform trend analysis). Archive files will contain all information about the past dealings of the business and would normally be stored on a different site to facilitate recovery in case of a disaster such as fire.

4. Audit file An Audit file is a file that does not store business data but data related to transaction log. For example, data and time of access, modification etc. of data, values of fields before and after modification etc.

5. Work file A Work file is file temporarily created to hold intermediate result of the data processing. For example, a sorted file of list of customers.

**File Organization** The physical organization of records on the disk is known as file Organization. There are different types of file organizations depending on the organization of records in the disk and other secondary storage are as following:
1. Serial file organization .
2. Sequential file organization.
3. Indexed sequential file organization.
4. Hashed file organization.
Before deciding on a specific file organization, we should ensure that its application leads to the following:
• Fast retrieval of records
• Reduce disk access time
• Efficient use of disk spaces.

Serial file organization A serial file is created by placing the record as it is created. It leaves no gap between the records that are stored on the disk. The utilization of space called packing density approaches 100 percent in this case. Examples of serial files are print file, dump file, log files, and transaction files. These files are created once and are not used for addition or deletion or any kind of record searching operation.
Sequential file organization In this organization, the records are physically ordered by primary key. To locate a particular record, the program starts searching from the beginning of the file till the matching primary key is found. Alphabetic list of customers is a common example of sequential file organization. Deletion of record may cause wastage of space and adding a new record requires rewriting of the file. This type of file organization is suitable for master files and is not used where fast response time is required.
Indexed sequential file organization Records are not physically ordered in this type of organization. Index is created to facilitate searching of records. Index Records give physical location of each data record. Indexes are separate files with a link to the main file. This type of file organization is used where faster response time is required.
Hashed file organization Hashing algorithm is used to physically ordered the records. The address where each record is stored is determined using hashing algorithms. The following is a typical hashing algorithm:
1. Uses a field in record called the hash field (generally the key filed).
2. Divides by prime number known as hash function.
3. Produces record address called the hash address.

**Design of Database**

Database design is similar to the pillars of a building. Any negligence, errors in Database design may lead to degraded performance of the software. In some cases such as real time applications, it may also lead to disasters. The following are various steps in Database design:

- Selection of database architecture
- Designing database schema
- Selecting indexes
- Estimating capacity of the database.

Selecting database architecture is one of the most challenging parts of database design for any information system. Before deciding on the target DBMS where the database is to be implemented, few considerations are required.

Hierarchical database structure is a kind of database management system that links records in tree data structure such that each record has only one owner. For example an order is owned by only one customer.

Network database structure is more flexible structure than Hierarchical model as well as relational model, but not preferred due to the high processing time.

Relational database structure is most commonly used database model. Reference key joins different tables together. Indexes provide rapid access to specific record in the database. For example, DB2, MySQL, Oracle are some RDBMS.

Object Oriented Database management system is based on object oriented paradigm. In object oriented database, data is stored as objects and can be interpreted only by using the methods specified by its class.

A blue print of the database is a physical model. A schema defines the database in terms of tables, keys, indexes and integrity rules. A relational schema consists of a relation(table), name of the attributes in the relations and restrictions on the relations called integrity constraints. A database schema is a set of relation schemas. Changes to a schema or database schema are expensive. So, careful thought must be given to design of a database schema. The following are some guidelines for the design of a database schema.

- Each entity should be implemented as a database table.
- Each attribute should be implemented as a field.
- Each table must have a primary key and an index based on the key.
- Each table may have zero or more secondary keys.
- Appropriate foreign keys.

Estimating capacity of the database: Database administrator needs to calculate the amount of disk space required for the database.It can be calculated by using the following formula:

Table size = record size * number of records in the table

where the number of records that will be present in each table at a particular period of time should be forecast. Database size is sum of sizes of all tables in that database. As rule of thumb, add a factor of 50% for indexes and other overheads to get the expected database size. While designing a database, future growth of database should also be kept in mind. Most of the business databases have a liner growth trend.

# 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.

A successful test is one that finds the errors. It executes the program with explicit intention of finding error, i.e., making the program fail. It is a process of evaluating system with an intention of creating a strong system and mainly focuses on the weak areas of the system or software.

## Characteristics of System Testing

System testing begins at the module level and proceeds towards the integration of the entire software system. Different testing techniques are used at different times while testing the

system. It is conducted by the developer for small projects and by independent testing groups for large projects.

## Stages of System Testing

The following stages are involved in testing –

**Test Strategy**

It is a statement that provides information about the various levels, methods, tools, and techniques used for testing the system. It should satisfy all the needs of an organization.

**Test Plan**

It provides a plan for testing the system and verifies that the system under testing fulfils all the design and functional specifications. The test plan provides the following information –

- Objectives of each test phase
- Approaches and tools used for testing
- Responsibilities and time required for each testing activity
- Availability of tools, facilities, and test libraries
- Procedures and standards required for planning and conducting the tests
- Factors responsible for successful completion of testing process

**Test Case Design**

- A number of test cases are identified for each module of the system to be tested.
- Each test case will specify how the implementation of a particular requirement or design decision is to be tested and the criteria for the success of the test.
- The test cases along with the test plan are documented as a part of a system specification document or in a separate document called **test specification** or **test description**.

**Test Procedures**

It consists of the steps that should be followed to execute each of the test cases. These procedures are specified in a separate document called test procedure specification. This document also specifies any special requirements and formats for reporting the result of testing.

**Test Result Documentation**

Test result file contains brief information about the total number of test cases executed, the number of errors, and nature of errors. These results are then assessed against criteria in the test specification to determine the overall outcome of the test.

## 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 –

### Unit Testing

Also known as Program Testing, it is a type of testing where the analyst tests or focuses on each program or module independently. It is carried out with the intention of executing each statement of the module at least once.

- In unit testing, accuracy of program cannot be assured and it is difficult to conduct testing of various input combination in detail.
- It identifies maximum errors in a program as compared to other testing techniques.

## Integration Testing

In Integration Testing, the analyst tests multiple module working together. It is used to find discrepancies between the system and its original objective, current specifications, and systems documentation.

- Here the analysts are try to find areas where modules have been designed with different specifications for data length, type, and data element name.
- It verifies that file sizes are adequate and that indices have been built properly.

## 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.

## Rules for System Testing

To carry out system testing successfully, you need to follow the given rules –

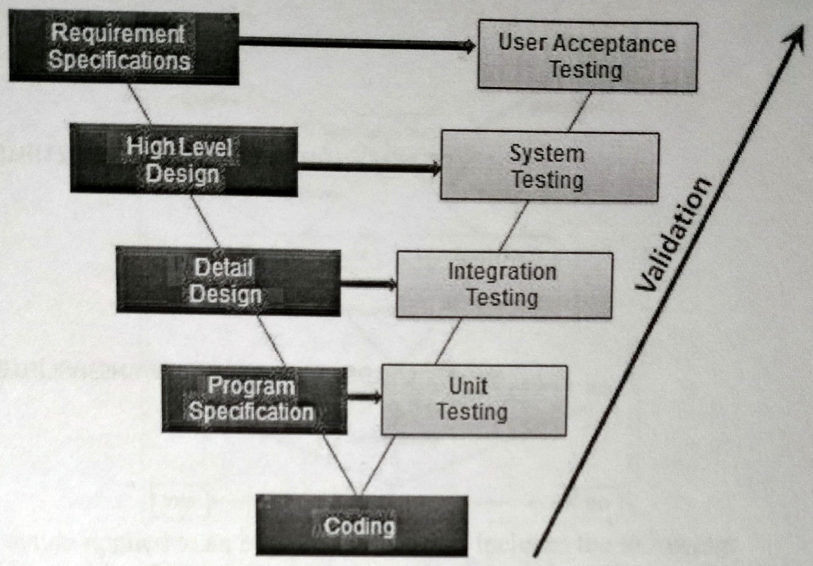
- Testing should be based on the requirements of user.
- Before writing testing scripts, understand the business logic should be understood thoroughly.
- Test plan should be done as soon as possible.
- Testing should be done by the third party.
- It should be performed on static software.
- Testing should be done for valid and invalid input conditions.
- Testing should be reviewed and examined to reduce the costs.
- Both static and dynamic testing should be conducted on the software.
- Documentation of test cases and test results should be done.

## Software Testing - Validation Testing

The process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements. Validation Testing ensures that the product actually meets the client's needs. It can also be defined as to demonstrate that the product fulfills its intended use when deployed on appropriate environment. It answers to the question, Are we building the right product?

## Validation Testing - Workflow:

Validation testing can be best demonstrated using V-Model. The Software/product under test is evaluated during this type of testing.



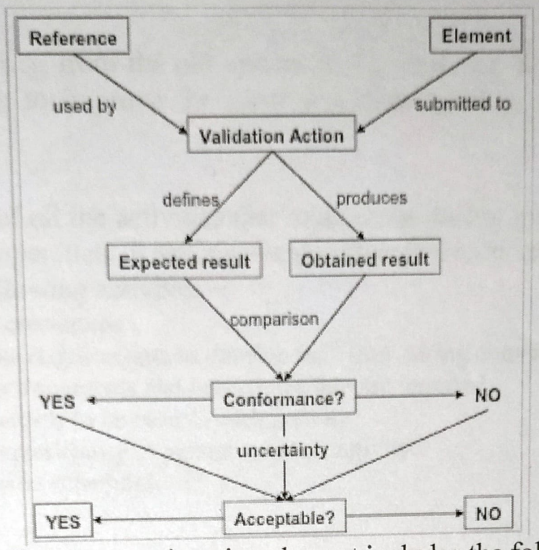

# System Validation

**System Validation** is a set of actions used to check the compliance of any element (a **system element**, a **system**, a document, a **service**, a task, a **system requirement**, etc.) with its purpose and functions. These actions are planned and carried out throughout the life cycle of the system. Validation is a generic term that needs to be instantiated within the context it occurs. When understood as a process, validation is a transverse activity to every life cycle stage of the system. Particularly during the development cycle of the system, the validation process is performed in parallel with the system definition and system realization processes and applies to any activity and product resulting from this activity. The validation process is not limited to a phase at the end of system development, but generally occurs at the end of a set of **life cycle** tasks or activities, and always at the end of each **milestone** of a development project. It may be performed on an iterative basis on every produced engineering element during development and may begin with the validation of the expressed **stakeholder requirements**. When the validation process is applied to the system when completely integrated, it is often called *final validation*. It is important to remember that while system validation is separate from **verification**, the activities are complementary and intended to be performed in conjunction.

## Concept of Validation Action

***Why Validate?***

The primary goal of **systems engineering** (SE) is to develop a solution that meets the needs and requirements of stakeholders. Validation is the process by which engineers ensure that the system will meet these needs and requirements. A **validation action** is defined and then performed (see Figure, below).




A validation action applied to an engineering element includes the following:

- Identification of the element on which the validation action will be performed.
- Identification of the reference that defines the expected result of the validation action.

Performing the validation action includes the following:

- Obtaining a result by performing the validation action onto the submitted element.
- Comparing the obtained result with the expected result.
- Deducing the degree of compliance of the element.
- Deciding on the acceptability of this compliance, because sometimes the result of the comparison may require a value judgment to decide whether to accept the obtained result as compared to the relevance of the context of use.

***What to Validate?***

Any engineering element can be validated using a specific reference for comparison, such as stakeholder requirements, system requirements, functions, system elements, documents, etc. Examples are provided in Table 1 below:

| Items | Explanation for Validation |
|---|---|
| **Document** | To validate a document is to make sure its content is compliant with the inputs of the task that produced the document. |
| **Stakeholder Requirement and System Requirement** | To validate a stakeholder requirement is to make sure its content is justified and relevant to stakeholders' expectations, complete and expressed in the language of the customer or end user. To validate a system requirement is to make sure its content translates correctly and/or accurately a stakeholder requirement to the language of the supplier. |
| **Design** | To validate the design of a system (logical and physical architectures) is to demonstrate that it satisfies its system requirements. |
| **System** | To validate a system (product, service, or enterprise) is to demonstrate that the product, service, or enterprise satisfies its system requirements and/or its stakeholder requirements. |
| **Activity** | To validate an activity or a task is to make sure its outputs are compliant with its inputs. |
| **Process** | To validate a process is to make sure its outcomes are compliant with its purpose. |

# 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. It includes the following activities –

- Name all files for conversions.
- Identifying the data requirements to develop new files during conversion.
- Listing all the new documents and procedures that are required.
- Identifying the controls to be used in each activity.
- Identifying the responsibility of person for each activity.
- Verifying conversion schedules.

## Conversion Methods

The four methods of conversion are –

- Parallel Conversion
- Direct Cutover Conversion
- Pilot Approach
- Phase-In Method

| Method | Description | Advantages | Disadvantages |
|---|---|---|---|
| Parallel Conversion | Old and new systems are used simultaneously. | Provides fallback when new system fails.<br>Offers greatest security and ultimately testing of new system. | Causes cost overruns.<br>New system may not get fair trail. |
| Direct Cutover Conversion | New system is implemented and old system is replaced completely. | Forces users to make new system work<br>Immediate benefit from new methods and control. | No fall back if problems arise with new system<br>Requires most careful planning |
| Pilot Approach | Supports phased approach that gradually implement system across all users | Allows training and installation without unnecessary use of resources.<br>Avoid large contingencies from risk management. | A long term phasein causes a problem of whether conversion goes well or not. |
| Phase-In Method | Working version of system implemented in one part of organization based on feedback, it is installed throughout the organization all | Provides experience and line test before implementation<br>When preferred new system involves new | Gives impression that old system is erroneous and it is not reliable. |

| | | | |
|---|---|---|---|
| | alone or stage by stage. | technology or drastic changes in performance. | |

## File Conversion

It is a process of converting one file format into another. For example, file in WordPerfect format can be converted into Microsoft Word.

For successful conversion, a conversion plan is required, which includes –

- Knowledge of the target system and understanding of the present system
- Teamwork
- Automated methods, testing and parallel operations
- Continuous support for correcting problems
- Updating systems/user documentation, etc

Many popular applications support opening and saving to other file formats of the same type. For example, Microsoft Word can open and save files in many other word processing formats.

### Control

Controls are technical, administrative or physical safeguards. Controls are the nexus used to manage risks through preventing, detecting or lessening the ability of a particular threat from negatively impacting business processes. Controls directly map to standards, since control testing is designed to measure specific aspects of how standards are actually implemented.

- ISACA Glossary: The means of managing risk, including policies, procedures, guidelines, practices or organizational structures, which can be of an administrative, technical, management, or legal nature.
- ISO 27000:2016: The policies, procedures, practices and organizational structures designed to provide reasonable assurance that business objectives will be achieved and undesired events will be prevented or detected and corrected. Measure that is modifying risk:
  - Controls include any process, policy, device, practice, or other actions which modify risk.
  - Controls may not always exert the intended or assumed modifying effect.
- NIST Glossary:Measure that is modifying risk. (Note: controls include any process, policy, device, practice, or other actions which modify risk.)
- NIST SP 800-53 R5: The safeguards or countermeasures prescribed for an information system or an organization to protect the confidentiality, integrity, and availability of the system and its information [security control]. The administrative, technical, and physical safeguards employed within an agency to ensure compliance with applicable privacy requirements and manage privacy risks [privacy control].

### Procedure

Procedures are a documented set of steps necessary to perform a specific task or process in conformance with an applicable standard. Procedures help address the question of how the organization actually operationalizes a policy, standard or control. Without documented procedures, there can be defendable evidence of due care practices. Procedures are generally the responsibility of the process owner / asset custodian to build and maintain but are expected to include stakeholder oversight to ensure applicable compliance requirements are

addressed. The result of a procedure is intended to satisfy a specific control. Procedures are also commonly referred to as "control activities."

- ISACA Glossary: A document containing a detailed description of the steps necessary to perform specific operations in conformance with applicable standards. Procedures are defined as part of processes.
- ISO 704:2009: A detailed description of the steps necessary to perform specific operations in conformance with applicable standards. A group of instructions in a program designed to perform a specific set of operations.
- NIST Glossary: A set of instructions used to describe a process or procedure that performs an explicit operation or explicit reaction to a given event.

## User Manual

User manuals are hugely important for users seeking to understand products and processes. Sometimes they will even be legally required for a company to sell its products to customers. Customers will often turn to your user manual before reaching out to your customer support team, so your manual has the potential to save you money on support costs. It's worth investing significant time and effort in your user manual so you can provide the best customer experience possible.

A user manual is a document provided to a user that helps in using a particular system, product or service seamlessly. It is also known as an instruction manual or a user guide. Such documents cover detailed information around operations, standards & guidelines, troubleshooting guides, functionalities & more.

User manuals usually contain step-by-step instructions guiding users through how to use your product and potential troubleshooting in case something goes wrong. It's not necessarily meant to be read cover-to-cover, and should contain a table of contents and index to help customers find the section that's relevant to their issue. It should contain a getting started guide at the beginning of the manual so customers can get up to speed quickly. It could be delivered as printed or online, or a mixture of both.

## Types of User Manuals

There are a number of different types of user manuals that you should consider when embarking on your technical communication.

### 1. Instruction Manual

An instruction manual contains basic instructions that tells users how best to use a product.

### 2. Training Manual

A training manual is a set of instructions that tell users how to complete a job, process or task.

### 3. Service Manual

A service manual is a set of instructions guiding users on how to keep a piece of machinery running at different points in its life.

## 4. User Manual

As mentioned before, user manuals are technical communication documents that assist users with the operation of a product.

## 5. Operation Manual

An operation manual is the documentation of your company's information including roles, responsibilities and processes.

## 6. Organizational Policy Manual

An organizational policy manual documents the company's policies, procedures and best practices.

## 7. Standard Operating Procedures(SOPs) Manual

A standard operating procedure provides clear instructions telling members of an organization how to go about completing certain processes.

# What makes a good user manual?

## 1. Plain language

Don't use flowery prose when writing your user manual. Your writing should be clear, simple, and easy to understand, without requiring a dictionary to get through. Use short sentences and words to make your text accessible. If you have to use a technical term, make sure you define it or link to a glossary.

## 2. Visual

Without visuals, your user manual will end up being a long wall of text without anything to break it up or catch the attention of your users. In this way, a lot of user manuals can be, well, boring to read. Make your documentation interactive by adding relevant images, diagrams and videos for your users to engage with. Make it clear which step of the instructions your visuals are referring to so users can make sense of them.

## 3. Logical hierarchical structure

Your users need to be able to search through your user manual using a predefined sense of structure to guide them. Your contents should have a logical hierarchical structure that makes sense to users when they are looking for information.

## 4. Searchable content

Ideally you need to make your content searchable for users by delivering it as an online knowledge base with a clear and prominent search bar. Your search bar should predict the terms that your users are typing in and search both the title and body content of your articles.

## 5. Clear topics and relevant articles

You should organize your content into clear topics that make sense for the articles contained within them. You should not have too many topics for your documentation or that could be overwhelming for users, and at the same

time don't have too many levels of subtopics or your documentation could be difficult to penetrate.

### 6. Feedback and reviews

Actively seek feedback from your users on your user manual and take their suggested improvements into account. Find out whether your users are actually being successful with your manual and whether it is enabling them to solve their problems.

## How to create a user manual

### 1. Identify the users

The first thing you need to do when writing your user manuals is to identify exactly who your users are – demographics, their needs, problems, and requirements to start with. Finding out who your audience is tells you how much detail you need to include in your user manual and the way that you should present your content.

### 2. Focus on the problem

All user manuals are aimed at solving a problem for users. You need to find out what these problems are in order to create a truly helpful manual, and solve the problem with your instructions. Of course, if there is a deeply entrenched problem with the product itself then that should be fixed rather than simply providing workarounds in the documentation.

### 3. Use sequential steps in order

Your instructions should be broken down into sequential steps that are presented in order as a numbered list. Try organizing it so the easiest task to accomplish is presented first.

Keep just one point per step to make it easy for your users to follow the instructions. Tell your users what the completed task will look like before they move on to the next step.

### 4. Map user journey

Conduct research into how users actually use your product so you can produce the appropriate documentation to accompany each touchpoint on the user journey. The essential goal is to see the product from the user's point of view and understand exactly how they are interacting with your brand. Part of your user journey mapping is identifying exactly what problem or goal the user has when using your product. You may have to break your users up into different segments as users may have different reasons for using your product.

### 5. Choose a Template

To keep your documentation consistent, it's important to develop a set of templates you can use to write your content. Your template should be clear and easy to follow, and include the vital components needed for each document. Your template can include:

- Space for an introduction

- Sections and subsections
- Sequential steps
- Warnings and call-outs
- Space for a conclusion

Make sure to specify font size, text to background contrast, and use colour coding consistently.

## 6. Write simple and easy to follow content

If you've followed the previous steps and you understand your users and write in a clear and compelling manner, your content should be simple and easy to follow. Rigorously edit your documentation to streamline your content and make sure it includes only the most essential elements that users need to complete a task.

Each step of your instructions should include just a single task so users can work through your documentation step-by-step without getting confused.

## 7. Treat all users as laymen

Don't assume that your users have a technical background – the language you choose should treat the users as if they are laymen, avoiding all jargon and specialist terms unless absolutely necessary. It's best to assume that your users know nothing about the product and be as explicit as possible in your documentation.

## 8. Test instructions alongside the product using naive users

When you've written your user manual you should text whether it works on users who have never used your product before. Make a note of where users get stuck in your documentation and revise your content accordingly.

Users should be able to make use of your documentation without reaching out to support. You should provide everything they need to know in your manual.

## 9. Build content using a practical approach

When writing your user manual, make sure you include practical examples alongside your instructions to show users the results they can expect to see if they complete the task. Your instructions should clearly explain what users will see or hear and any feedback they might get from the product.

## 10. Explain symbols, icons and codes early

You may need to use symbols, icons and codes in your documentation to represent certain information. Make sure you explain them early on so users aren't left scratching their heads.

# Software Maintenance

Software Maintenance is the process of modifying a software product after it has been delivered to the customer. The main purpose of software maintenance is to modify and update software applications after delivery to correct faults and to improve performance.

**Need for Maintenance** – Software Maintenance must be performed in order to:

- Correct faults.
- Improve the design.
- Implement enhancements.
- Interface with other systems.
- Accommodate programs so that different hardware, software, system features, and telecommunications facilities can be used.
- Migrate legacy software.
- Retire software.

**Challenges in Software Maintenance:**

The various challenges in software maintenance are given below:

- The popular age of any software program is taken into consideration up to ten to fifteen years. As software program renovation is open ended and might maintain for decades making it very expensive.
- Older software programs, which had been intended to paintings on sluggish machines with much less reminiscence and garage ability cannot maintain themselves tough in opposition to newly coming more advantageous software program on contemporary-day hardware.
- Changes are frequently left undocumented which can also additionally reason greater conflicts in future.
- As era advances, it turns into high priced to preserve vintage software program.
- Often adjustments made can without problems harm the authentic shape of the software program, making it difficult for any next adjustments.

**Categories of Software Maintenance** – Maintenance can be divided into the following:

1. **Corrective maintenance:** Corrective maintenance of a software product may be essential either to rectify some bugs observed while the system is in use, or to enhance the performance of the system.
2. **Adaptive maintenance:** This includes modifications and updations when the customers need the product to run on new platforms, on new operating systems, or when they need the product to interface with new hardware and software.
3. **Perfective maintenance:** A software product needs maintenance to support the new features that the users want or to change different types of functionalities of the system according to the customer demands.
4. **Preventive maintenance:** This type of maintenance includes modifications and updations to prevent future problems of the software. It goals to attend problems, which are not significant at this moment but may cause serious issues in future.