

System Analysis and Design

What is System?

- *System is a group of elements or components which work together to accomplish a common task.*

What is System Analysis & Design ?

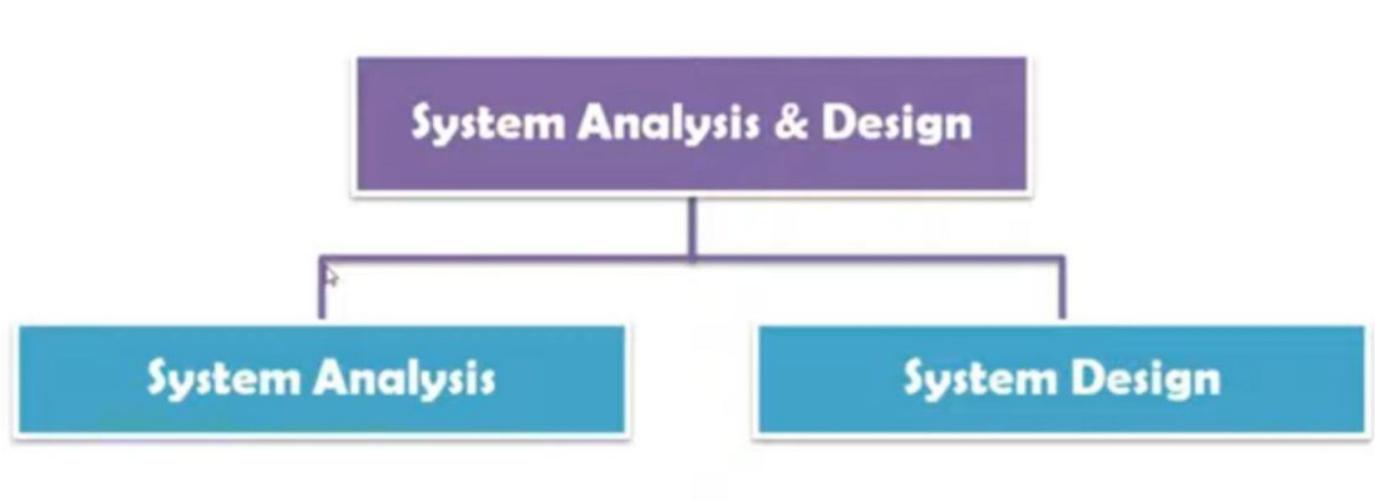
- *In very simple words, system analysis and design is a study in which we learn how to analyze an existing system and create a better one.*



Why we need it?

- *For System Development*
 - *Creating a new one*
 - *Updating the existing one*





System Analysis

- *System analysis is a process of studying and observing a system to know how it works and to identify its goal and purposes.*
- *System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives.*
- *It specifies "what the system should do".*



System Design



- *It is a process of planning a new system or replacing an existing system.*
- *It is done by defining its components or modules to satisfy the specific requirements.*
- *It focuses on "how to accomplish the objective of the system".*

System Analysis and Design

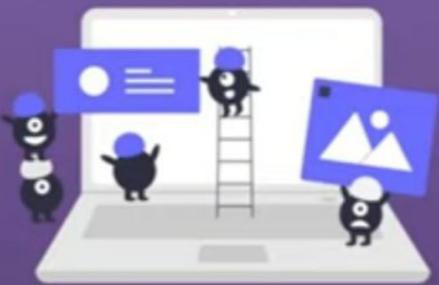


System Analysis and Design

- What is System***
- Properties / Characteristics of a System***
- Elements of a System***

What is System?

- *System is a group of elements or components which work together to accomplish a common task.*



Properties / Characteristics of a System

- A system must have following properties / characteristics :
 - Organization
 - Integration
 - Interaction
 - Interdependence
 - Central Objective



Properties / Characteristics of a System

□ Organization

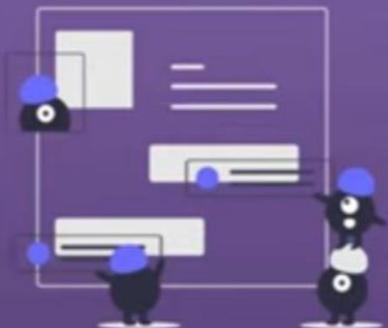
- It implies Structure and Order.**
- Elements should be well arranged in order to achieve predetermined objective.**



Properties / Characteristics of a System

□ Integration

- It implies how components of a system are tied together.**
- The parts of the system must work together within the system even though each part performs a unique function.**



Properties / Characteristics of a System

□ Interaction

- It implies manner in which the components operate with each other.***



Properties / Characteristics of a System

□ Interdependence

- It implies how components of a system depend on one another.***
- The components must be well linked and they must coordinate with each other for proper functioning.***
- Interdependence is important because the output of one subsystem may be required by other subsystem as input.***

Properties / Characteristics of a System

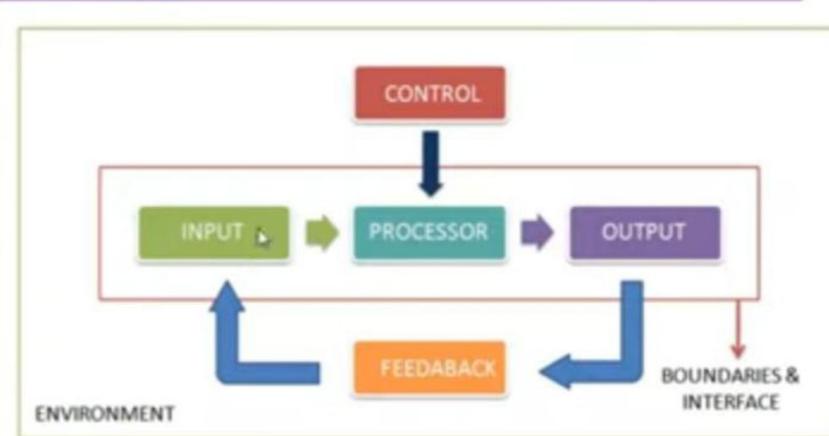
Central Objective

- The objective of the system must be central.**
- It means, the user must know the main objective of the system in the early phase for successful design and conversion.**



Elements of a System

- There are 6 elements in a system:
 - Inputs and Outputs
 - Processor
 - Control
 - Feedback
 - Boundaries & Interface
 - Environment



Elements of a System

□ Inputs and Outputs

- The information that enters in a system is treated as Input.***
- Output is the outcome of input after processing.***
- The main aim of a system is to produce an output which is useful for its users.***

Elements of a System

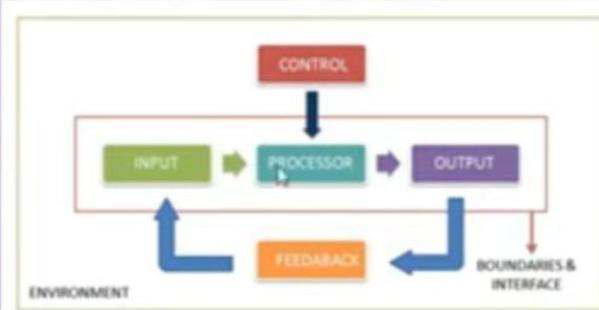
Processor

- It transforms the input into useful output.*
- It is an operational element.*
- It may modify the input either totally or partially, depending on the output specification.*

Elements of a System

Control

- This element guides the system.*
- It is a decision making element that controls the pattern of activities governing input, processing and output.*



Elements of a System

□ Feedback

- The Output is checked with the desired output set and then necessary steps are taken for achieving the output as per the standards.**
- Positive feedback encourages the performance of the system.**
- Negative feedback is informational in nature.**

Elements of a System

□ Boundaries and Interface

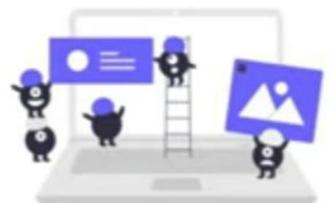
- The boundaries are the limits under which the components of a system interact with each other.**
- Each system has boundaries that determine its sphere of influence and control.**
- The interconnection and the interaction between the sub-systems is known as the Interface.**

Elements of a System

□ Environment

- The things outside the boundary of the system are known as environment.**
- It is the source of external elements that strike on the system.**
- Change in the environment affects the working of the system.**

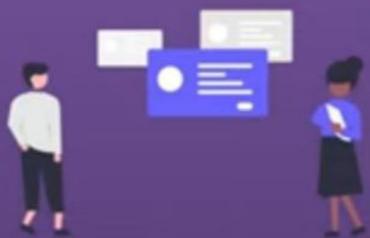
System Analysis and Design



Types of a System

A system can divided into following types :

- Physical or Abstract System
- Open or Closed System
- Adaptive and Non-Adaptive System
- Permanent or Temporary System
- Natural or Manufactured System



Types of System

Physical

- *Physical systems are tangible entities.*
- *It may be static or dynamic in nature.*
- *The physical parts of the computer center are desks, and chairs that facilitate operation of the computer are static. A computer is a dynamic system in which programs, data, and applications can change according to the user's needs.*

Abstract

- *Abstract systems are non-physical entities or conceptual that may be formulas, representation or model of a real system.*

Types of System

Open

- *An open system must interact with its environment.*
- *It receives inputs and delivers outputs to the outside of the system.*
- *It must adapt to the changing demands of the users.*

Closed

- *A closed system is isolated from environment influences.*
- *A completely closed system is rare.*

Types of System

Adaptive

- *Adaptive System responds to the change in the environment in a way to improve their performance and to survive.*
- *For example: human beings, animals.*

Non-Adaptive

- *Non Adaptive System is the system which does not respond to the environment.*
- *For example, machines.*

Types of System

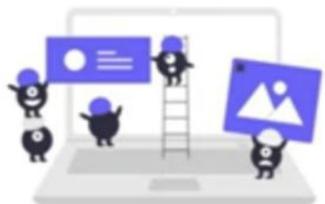
Temporary

- *Temporary System is made for specified time and after that they are demolished.*
- *For example: A DJ system is set up for a program and it is dissembled after the program.*

Permanent

- *Permanent System persists for long time.*
- *For example, business policies.*

System Analysis and Design



System Analyst

- A *systems analyst* is a person who uses *System Analysis and Design techniques to solve system/business problems.*
- A *system analyst analyzes, designs and implements system to fulfill organizational needs.*
- *System Analyst conducts system study, identifies requirements & determines the procedures to achieve system objectives.*
- *System Analyst designs and implements the system to suit organizational requirements for effective results.*

System Analyst

- *System analysts carry the responsibilities of researching problems, finding solutions, recommending courses of actions and coordinating with other members in order to meet specified requirements.*
- *In the IT industry, System Analyst figures out how to solve a problem by linking different computers and by specifying what platform, protocols, software, hardware and communications medium can be used to solve a problem.*
- *System Analyst is responsible for the system from its birth to death.*

Role And Responsibilities of System Analyst

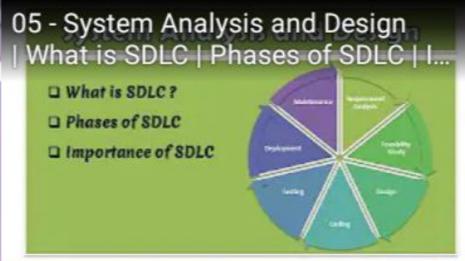
- Research and evaluate new technologies*
- Identify the organizational needs*
- Analyze costs and benefits*
- Add new functionality to systems*
- System Analysis*
- System Design*
- System Upgradation*
- System Implementation*
- System Maintenance*

System Analyst is also a

- **Change Agent** : System Analyst evaluates the current system and as well as researches for new future updates for better System Development.
- **Architect** : System Analyst is the architect of the system.
- **Motivator** : System Analyst is a motivator who motivates his team to work properly and do their best.
- **Monitor** : System Analyst monitors the complete system.

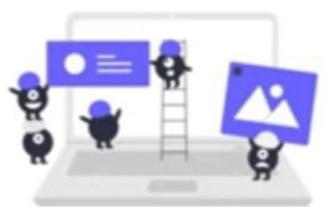
Qualities/Attributes of System Analyst

- Knowledge of the Organization***
- Knowledge of computer hardware and software***
- Problem-solving and critical thinking***
- Analytical skills***
- Adaptability***
- Motivator Skill***
- Good understanding, communication and teaching abilities.***



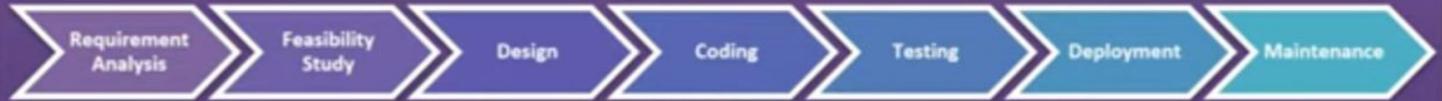


System Analysis and Design



SDLC

- *SDLC stands for “Software Development Life Cycle”.*
- *SDLC is a collection of processes which are followed to develop a software.*
- *SDLC is a methodology that defines some processes which are followed to develop a high-quality software.*



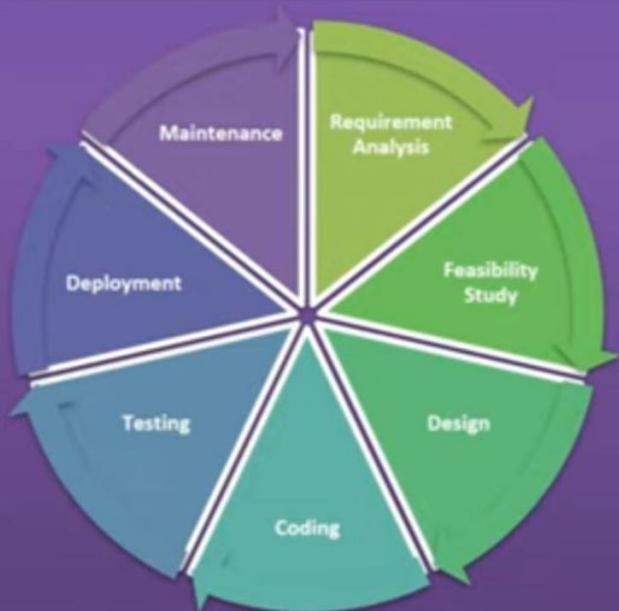
SDLC

- *It covers the detailed plan for building, deploying and maintaining the software.*
- *The main aim of SDLC is to define all the tasks required for developing and maintaining software.*
- *It is followed for a software project within a software developing organization.*



Phases of SDLC

- Phase 1: Requirement Analysis**
- Phase 2: Feasibility Study**
- Phase 3: Design**
- Phase 4: Coding**
- Phase 5: Testing**
- Phase 6: Deployment**
- Phase 7: Maintenance**



Phases of SDLC

□ Phase 1: Requirement Analysis

- It is the first phase of SDLC in which all the necessary information is collected from the customer to develop the software as per their expectation.**
- Some important questions like: what is the need of software, who will be the end-user, what is the future scope of that software etc. are discussed.**
- The main aim of this phase is to collect the details of each requirement of the customer so that the developers will clearly understand what they are developing and how to fulfill the customer's requirements.**
- This phase gives a clear picture of what we are going to build.**



Phases of SDLC

Phase 2: Feasibility Study

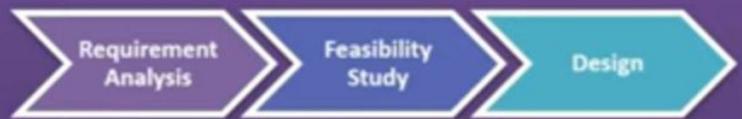
- It is the second phase of SDLC in which an organization discusses about the cost and benefits of the software.*
- It is an important phase because profit from the software plays an important role as if cost is very high then company may face loss.*
- After the feasibility study, the project may be accepted, accepted with modifications or rejected.*
- It measures how much beneficial the product is for the organization.*



Phases of SDLC

□ Phase 3: Design

- It is the third phase in which architects start working on logical designing of the software.***
- In this phase a SRS (System Requirement Specification) document is created which contains all logical details like how the software will look like, which language will be used, database design, modular designs etc.***
- This phase provides a prototype of the final product.***
- Basically, all it includes is design of everything which has to be coded.***



Phases of SDLC

□ Phase 4: Coding

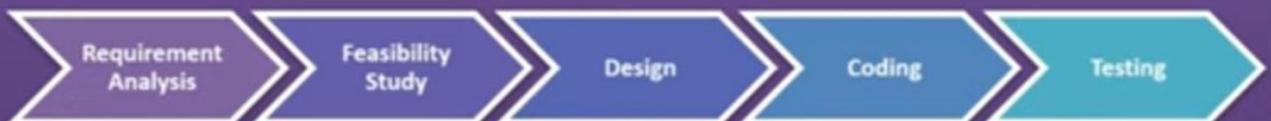
- When the designing of the software is completed, then, a group of developers starts coding of the design using a programming language.***
- The interface of the software and all its internal working according to design phase is implemented in coding phase.***
- A number of developers code the modules and then all modules are arranged together to work efficiently.***
- It is the longest phase of SDLC.***



Phases of SDLC

□ Phase 5: Testing

- Once the software development is completed, then it is sent to the testers. The testing team starts testing the functionality of the entire system.***
- In this phase, the software is checked for bugs or errors.***
- Whenever a bug is found, then the software is resent to the coders to fix it and then overall software is re-tested.***
- This is done to verify that the entire application works according to the customer requirement.***



Phases of SDLC

Phase 6: Deployment

- After overall testing of the software and after checking that is bug free, then the software is launched and available for the users to use it.
- Even after deployment of the software, if any bugs or errors are still found then the software is re-evaluated by the maintenance team and then it is re-deployed with a new version.



Phases of SDLC

□ Phase 7: Maintenance

- The maintenance team look over the software usage and user's feedbacks.***
- Maintenance is necessary to eliminate errors in the system during its working life and to tune the software.***
- The bug fixing, upgrade and enhancement of the software is looked over by the maintenance team.***



What is Feasibility Study

- As the name suggests, feasibility study is a study to reveal whether a project is feasible or not.
- It is conducted in order to find answers to the following questions:
 - Do we have required resources and technologies to build the project
 - Do we receive profit from the project
- It tells us whether a project is worth the investment.
- After the feasibility study, the project may be accepted, accepted with modifications or rejected.

Types of Feasibility Study

There are five types of feasibility study :

- Technical Feasibility*
- Economic Feasibility*
- Legal Feasibility*
- Operational Feasibility*
- Scheduling Feasibility*

Types of Feasibility Study

Technical Feasibility :

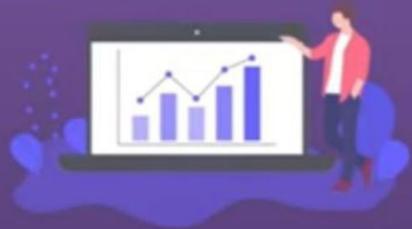
- In technical feasibility, we check whether we have required technical resources (like hardware and software) to develop the project.*
- This feasibility study also analyzes technical skills and capabilities of technical team, existing technology can be used or not, maintenance and up-gradation is easy or not for chosen technology etc.*



Types of Feasibility Study

Economical Feasibility :

- In Economic Feasibility study, cost and benefit of the project is analyzed.*
- In this feasibility study, a detail analysis is carried out to know what will be cost of the project including hardware and software resource required, design and development cost and so on.*
- It is also analyzed whether project will be beneficial for organization or not.*



Types of Feasibility Study

Legal Feasibility :

- In legal feasibility study we investigate whether the project is legal or not.**



Types of Feasibility Study

Operational Feasibility :

- In operational feasibility study, we examine whether the project satisfies the requirements identified in the requirement analysis phase.**



Types of Feasibility Study

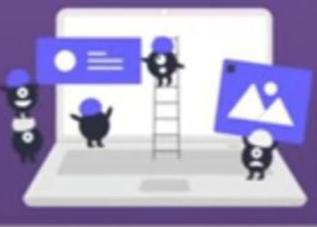
Scheduling Feasibility :

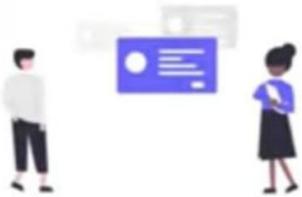
- In scheduling feasibility study, we estimate the time necessary to complete the project.
- We also consider the organization's capabilities and determine whether that amount of time is available or not.



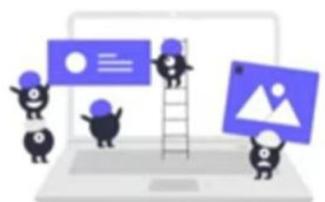
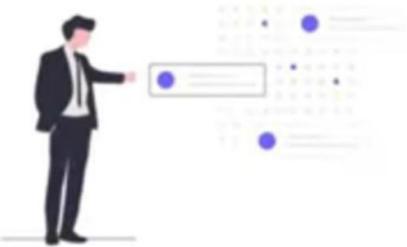
Importance of Feasibility Study

- Get a clear-cut idea of whether the project is likely to be successful, before allocating budget, manpower, and time.*
- Improves project teams' focus*
- Provides valuable information for a "go/no-go" decision*
- Identifies a valid reason to undertake the project*
- Diagnoses errors and aids in troubleshooting them*
- Enhances the success rate by evaluating multiple parameters*





System Analysis and Design

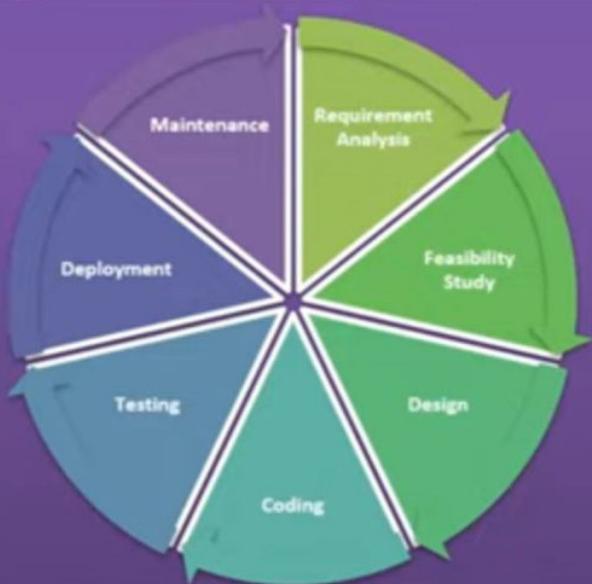


System Analysis and Design

- What is SRS (System Requirement Specification) ?***
- Features/Characteristics of a good SRS***

Phases of SDLC

- Phase 1: Requirement Analysis***
- Phase 2: Feasibility Study***
- Phase 3: Design***
- Phase 4: Coding***
- Phase 5: Testing***
- Phase 6: Deployment***
- Phase 7: Maintenance***



What is SRS

- *SRS stands for “System Requirement Specification”.*
- *A SRS is a detailed description of a software system to be developed with its functional and non-functional requirements.*
- *SRS document contains all logical details like how the software will look like, which language will be used, database design, modular designs etc.*
- *SRS is a formal report, which acts as a representation of the software.*

What is SRS

- A good SRS defines the how Software System will interact with all internal modules, hardware, communication with other programs etc.
- SRS allows the customers to review whether all their requirements are fulfilled by the software or not.
- SRS serves two main purposes:
 - It is used to defines the need and expectation of the users.
 - It serves as a contract document between customers and company.



Features/Characteristics of a good SRS

□ Correctness:

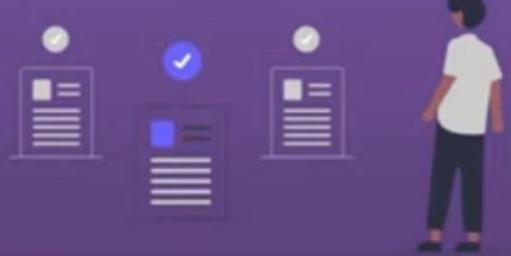
- SRS is said to be correct if it covers all the requirements that are actually expected from the system.***

□ Completeness:

- SRS is complete if it includes all essential requirements of the user and a detailed representation of every module to be developed by the user.***

□ Consistency:

- SRS is consistent if it has no conflicts between any sets of requirements.***



Features/Characteristics of a good SRS

- **Unambiguousness:**
 - *Sometimes in SRS, some words have more than one meaning and this might confuse developers in getting the exact reference. It is advisable to check for such ambiguous words and make the meaning clear for better understanding.*
- **Ranking for importance and stability:**
 - *Typically, all requirements are not equally important. Some may be essential and some may not. There should a criterion to classify the requirements as less or more important. An identifier mark can be used with every requirement to indicate its rank or stability.*

Features/Characteristics of a good SRS

□ Modifiability:

- SRS should be made as modifiable as possible and should be capable of easily accepting changes to the system to some extent.***

□ Design Independence:

- There should be an option to choose from multiple design alternatives for the final system.***

□ Understandable by the customer:

- The language used in SRS should be kept easy understandable.***



System Analysis and Design

System Analysis and Design

- What is Structured Analysis ?***
- Structured Analysis Tools***
- What is Structured Design ?***

What is Structured Analysis

- *Structured analysis is a technique that uses graphical diagrams to develop and portray system specifications that are easily understood by users.*
- *Structured Analysis is a development method that allows the analyst to understand the system and its activities in a logical way.*



What is Structured Analysis

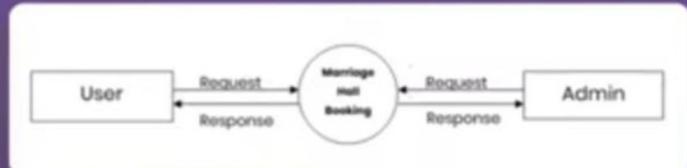
- *It is a systematic approach, which uses graphical tools that analyze and refine the objectives of an existing system and develop a new system specification which can be easily understood by user.*
- *The basic goal of SA/SD is to improve quality and reduce the risk of System failure.*



Structured Analysis Tools

□ *The tools and techniques used during Structured Analysis are :*

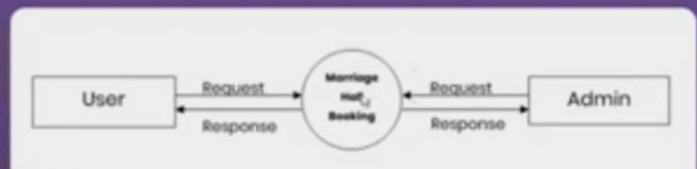
- *DFD (Data Flow Diagram)*
- *Data Dictionary*
- *Structured English*
- *Pseudo Code*
- *Decision Tree*
- *Decision Table*



Structured Analysis Tools

□ *The tools and techniques used during Structured Analysis are :*

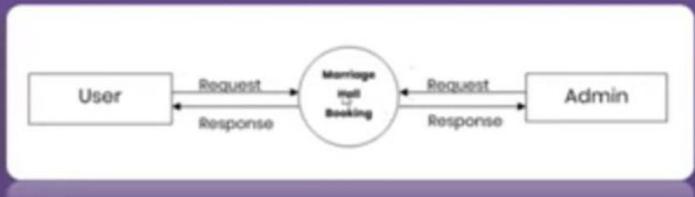
- DFD (Data Flow Diagram)
- Data Dictionary
- Structured English
- Pseudo Code
- Decision Tree
- Decision Table



Structured Analysis Tools

□ *The tools and techniques used during Structured Analysis are :*

- *DFD (Data Flow Diagram)*
- *Data Dictionary*
- *Structured English*
- *Pseudo Code*
- *Decision Tree*
- *Decision Table*



Structured Analysis Tools

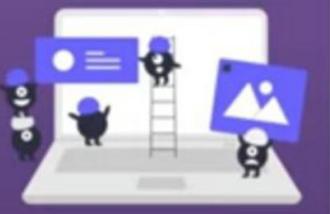
- *The tools and techniques used during Structured Analysis are :*

- *DFD (Data Flow Diagram)*
- *Data Dictionary*
- *Structured English*
- *Pseudo Code*
- *Decision Tree*
- *Decision Table*

Field Name	Data type	Field Length	Constraint	Description
Client_id	Number	10	Primary key	Client id,Auto generated
Client_name	Varchar	20	Not null	Name of client
Password	Varchar 2	30	Not null	Login Password for client
Client_Company	Varchar	20		Name of client company
Activation_date	Date		Not null	Starting date
Contact_no	Number	15	Not null	Landline or mobile number
Email_id	Varchar2	30	Not null	Any email id
Max_Users	Number	10	Not null	Maximum number of users
Current_users	Number	10	Not null	Currently present user

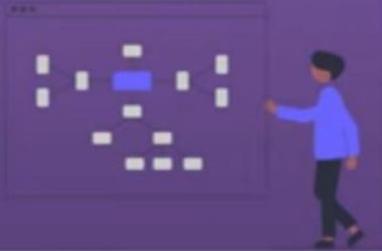
What is Structured Design

- *Structured design is a concept of decomposing problem into several well-organized elements of solution.*
- *It gives better understanding of how the problem is being solved.*
- *Structured design also makes it simpler for designer to concentrate on the problem more accurately.*



What is Structured Design

- *Structured design is mostly based on 'divide and conquer' strategy, where a problem is broken into several small problems and each small problem is individually solved until the whole problem is solved.*
- *The small pieces of problem are solved by means of solution modules. Structured design emphasizes that these modules be well organized in order to achieve precise solution.*



System Analysis and Design

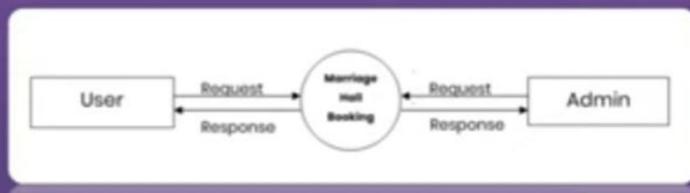


System Analysis and Design

- DFD (Data Flow Diagram)***
- Basic Elements of DFD / Components of DFD***
- Types of DFD***
- Levels of DFD***

What is DFD

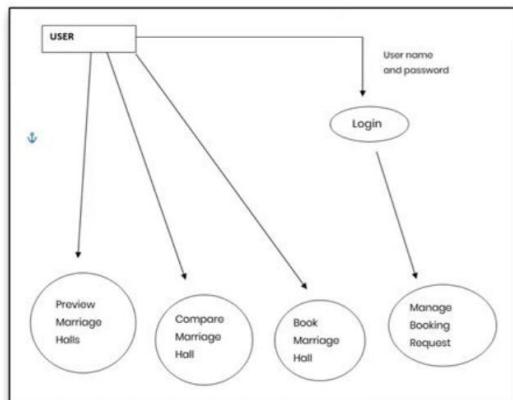
- DFD stands for “Data Flow Diagram ” and also known as “Bubble Chart”.
- DFD show the flow of data between various elements of a system in graphical form.
- It also expresses the requirement of the system and shows how the current system is implemented.



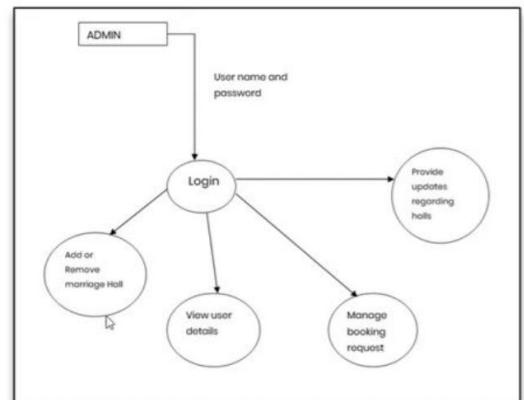
What is DFD

- *It gives an overview of what data a system processes, what transformations are being performed, what data are stored, what results are produced and where they flow.*
- *Its graphical nature makes it a good communication tool between user and system designer (System Analyst).*

What is DFD



Marriage
Hall
Booking



What is DFD

- *DFD is easy to understand and quite effective tool for communication (between user and system analyst) when the required design is not clear.*
- *It requires a large number of iterations for obtaining the most accurate and complete solution.*

Basic Elements of DFD / Components of DFD

- *Symbols used in designing the DFD are:*

SYMBOL NAME	SYMBOL	MEANING
Circle	○	Process
Open rectangle	□	Data store
Rectangle	■	Entity
Arrows	↑↓↔	Data flow

Types of DFD

Logical DFD

- *It shows how data flows in a system.*
- *It describes what data is moved from one entity to another.*
- *It focus on what happens in the data flow.*
- *It shows how system works not how the system can be implemented.*
- *It is implementation independent.*

Physical DFD

- *It shows how data flow is actually implemented in the system.*
- *It describes how data is moved from one entity to another.*
- *It shows how the current system operates and how a system will be implemented.*
- *It is implementation dependent.*

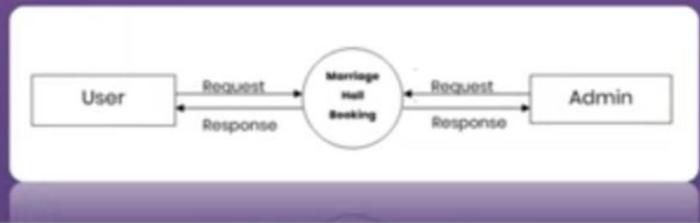
Levels of DFD

- *There are mainly 3 levels in DFD:*
 - *0-level DFD*
 - *1-level DFD*
 - *2-level DFD*

Levels of DFD

0-level DFD:

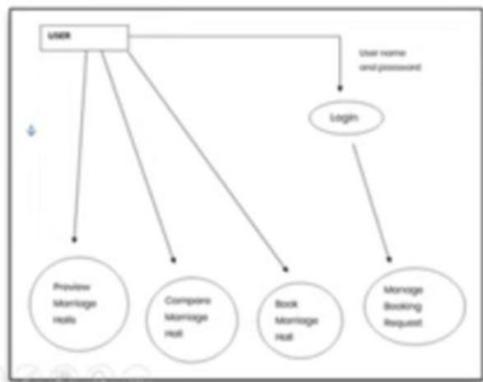
- It shows the system as a single process with its relationship to external entities.*
- It represents the entire system as a single process with input and output data indicated by incoming/outgoing arrows.*
- It is also known as fundamental system model or context diagram.*
- It shows an abstract view.*



Levels of DFD

□ 1-level DFD:

- In level 1 DFD, we decompose the context level DFD (0-level DFD) into multiple components and define each component in detail.



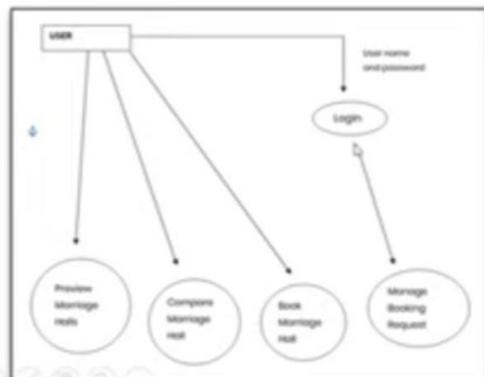
Marriage
Hall
Booking



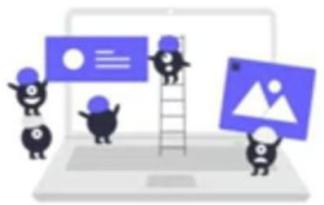
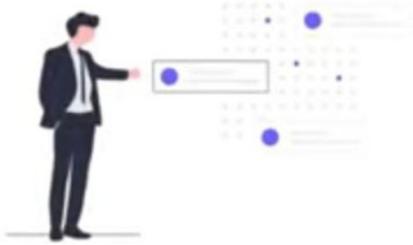
Levels of DFD

□ 2-level DFD:

- In level 2 DFD, we decompose the 1-level DFD into more detail and define each necessary detail about system functioning properly.



System Analysis and Design



System Analysis and Design

- What is Data Dictionary***
- Types of Data Dictionary***
- Advantages of Data Dictionary***

What is Data Dictionary

- A *data dictionary* contains metadata i.e. *data about the database*.
- *It contains information such as what is in the database, who is allowed to access it, where is the database physically stored etc.*
- *It plays an important role in building a database.*

What is Data Dictionary

client_id	client_name	Password	Contact_no	email
1	Raju	12345	777 777 777	raju@email.com
2	Rohan	Abcdef	666 666 666	rohan@email.com
3	Sohan	54321	555 555 555	sohan@email.com
4	Mohan	14145	222 222 222	mohan@email.com

Field Name	Data type	Field Length	Constrains	Description
client_id	number	10	primary key	Client id, Auto generated
client_name	varchar	20	not null	Name of client
password	varchar	30	not null	Login password
contact_no	number	10	not null	Contact of client
email	varchar	40	not null	Client email



What is Data Dictionary

- *The data dictionary in general contains information about the following:*

- *Names of all tables in the database*
- *Names of each field in the tables of the database*
- *Constraints defined on tables*
- *Physical information about tables like their storage location, storage method, etc.*

Field Name	Data type	Field Length	Constrains	Description
client_id	number	10	primary key	Client id, Auto generated
client_name	varchar	20	not null	Name of client
password	varchar	30	not null	Login password
contact_no	number	10	not null	Contact of client
email	varchar	40	not null	Client email



What is Data Dictionary

- *The users of the database normally don't interact with the data dictionary, it is only handled by the database administrator.*
- *A data dictionary is also called a metadata repository.*

Types of Data Dictionary

- *There are two types of data dictionary :*

- *Active*
 - *Passive*

Types of Data Dictionary

□ Active:

- It may happen that the structure of the database has to be changed, like adding new attributes or removing older ones. If those changes are updated automatically in the data dictionary by the DBMS, then the data dictionary is an active one.**
- It is also known as integrated data dictionary.**

Types of Data Dictionary

Passive:

- When the DBMS maintains the data dictionary separately and it has to be updated manually, then the data dictionary is an passive one.**
- It is also known as non-integrated data dictionary.**
- In this case, there is a chance of mismatch with the database objects and the data dictionary.**

Advantages of Data Dictionary

- *It gives the well structured and clear information about the database.*
- *One can analyze the requirement, any redundancy like duplicate columns, tables etc.*
- *It is very helpful for the administrator or any new DBA to understand the database. Since it has all the information about the database, DBA can easily able to track any chaos in the database.*

Advantages of Data Dictionary

- Since database is a very huge, and will have lots of tables, views, constraints, indexes etc, it will be difficult for anyone to remember. Data dictionary helps user by providing all the details in it.
- It is a valuable reference in any organization because it provides documentation.

System Analysis and Design

System Analysis and Design

- Cost-Benefit Analysis
- How to perform Cost-Benefit analysis

Cost-Benefit Analysis

- *It is a part of economical feasibility study.*
- *It is an analysis for estimating cost involved in the development of a project and the possible profit to be derived from it.*
- *It works on a simple rule, i.e., identifying the benefits from the proposal and associated costs, and subtracting the costs from benefits.*
- *It helps us to assign value to a proposal, calculate the future cash flow and measure the financial worthiness of it.*

How To Perform Cost-benefit Analysis

- Identify costs and benefits from the system*
- Assign Values to the costs and benefits*
- Determine the total cash flow*
- Determine the project's Economic value*
 - Return On Investment (ROI)*
 - Break-Even Point (BEP)*
 - Net Present Value (NPV)*

System Analysis and Design



System Analysis and Design

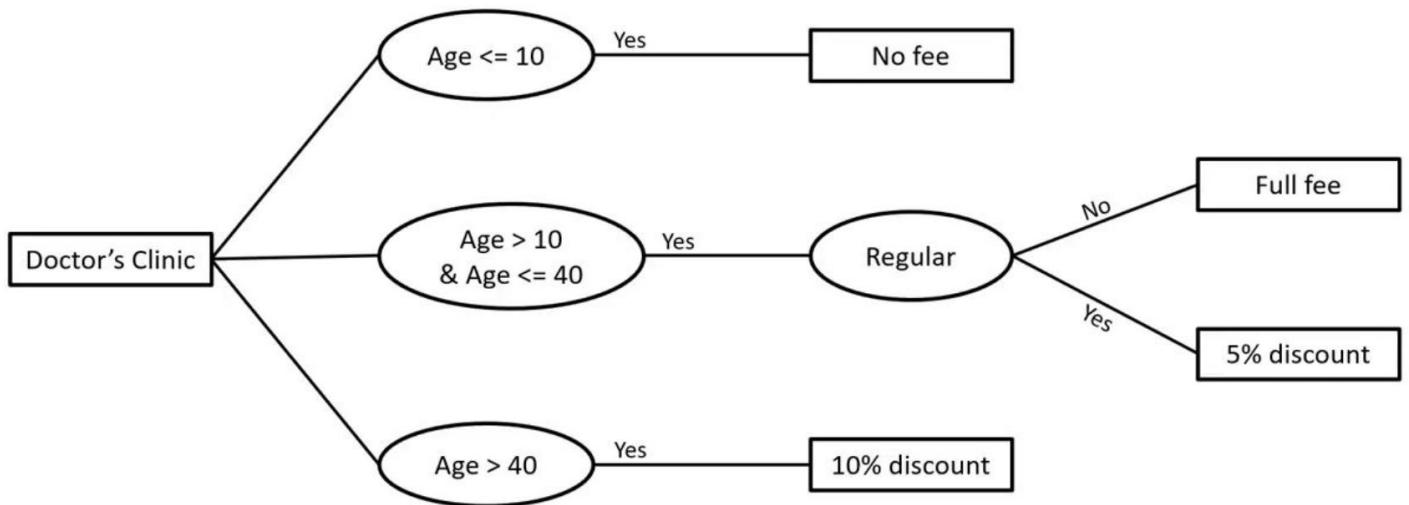
- **Decision Tree**
 - Examples
- **Decision Table**
 - Components of decision table
 - Examples



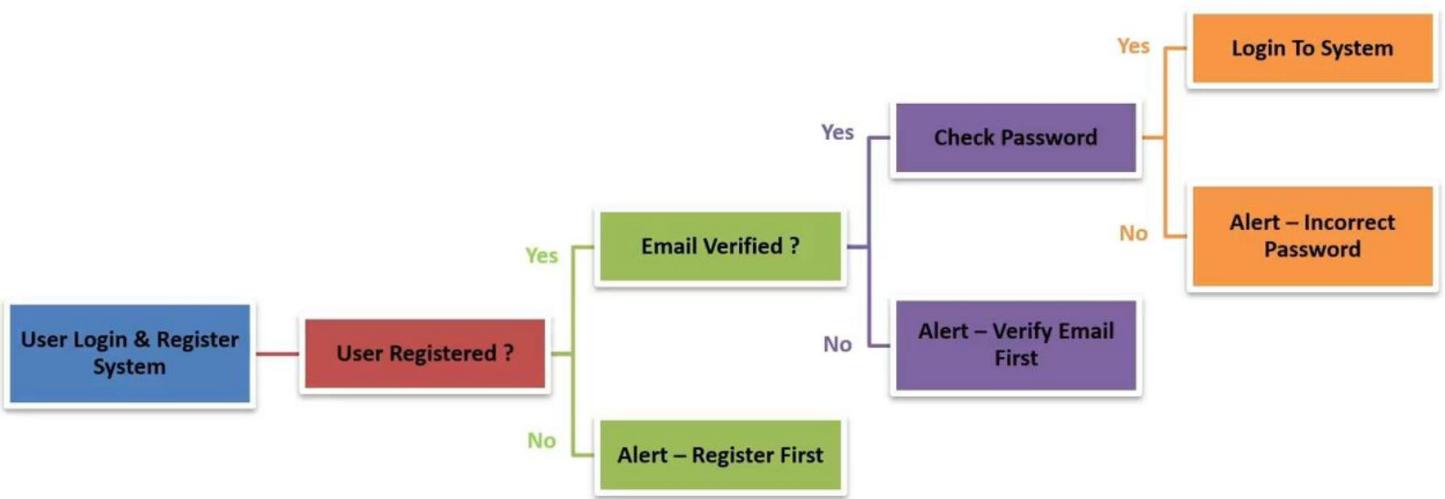
Decision Tree

- ***Decision Tree is a graph which uses a branching method in order to display all the possible outcomes of any decision.***
- ***It helps in processing logic involved in decision making and corresponding actions taken.***
- ***It is a diagram that shows conditions and their alternative actions within horizontal tree framework.***
- ***It helps analyst to consider the sequence of decisions and identifies the accurate decision that must be made.***

Decision Tree



Decision Tree



Decision Table

- *Decision Table is just a tabular representation of all conditions and actions.*
- *It is used for describing the complex problem in a precise manner which is easily understandable.*
- *We can derive decision table from decision tree.*
- *It helps testers to search the effects of combinations of different inputs and other software states that must correctly implement business rules.*

Decision Table

- ***Components of Decision Table:***

- ***Condition stub:*** It is the upper left quadrant which lists all the condition to be checked.
- ***Action Stub:*** It is the lower left quadrant which lists all the action to be carried out to meet such condition.
- ***Condition Entry:*** It is the upper right quadrant which provides answers to questions asked in condition stub quadrant.
- ***Action Entry:*** It is the lower right quadrant which provides the appropriate action resulting from the answers to the conditions in the condition entry quadrant.

Condition stub	Condition entry
Action stub	Action entry

Decision Table

User Login & Register System

Condition	Rule 1	Rule 2	Rule 3	Rule 4
Username	F	T	F	T
Password	F	F	T	T
Action	-	-	-	-
Output	Error	Error	Error	Login

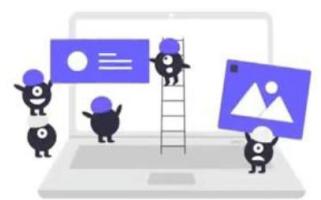
Decision Table

Doctor's Clinic

Condition	Rule 1	Rule 2	Rule 3	Rule 4
Age<=10	Yes	-	-	-
Age>10 & Age<=40	-	-	Yes	Yes
Age>40	-	Yes	-	-
Regular	-	-	Yes	No
Action				
Fee	No	Yes	Yes	Yes
Discount	-	10%	5%	No

System Analysis and Design

System Analysis and Design



System Analysis and Design

- Software testing***
- Need of software testing***
- Importance of software testing***

Software Testing

- ***Software testing is a method to check whether the actual software product fulfils the expected requirements and to ensure that software product is defect free.***
- ***It involves execution of software components using manual or automated tools to evaluate whether it is bug free or not.***



Software Testing

- *The primary purpose of software testing is to identify errors, gaps or missing requirements in comparison to actual requirements.*
- *The testing team executes the software with an intention of finding error in it and a successful test is one that finds errors.*



Need & Importance of Software Testing

- Find bugs or errors in the developed software.*
- Resolve bugs before delivery of software.*
- Save your money for the long term as if bugs are caught in the earlier stage of software testing then it costs less to fix.*
- Security and satisfaction after deployment.*

System Analysis and Design



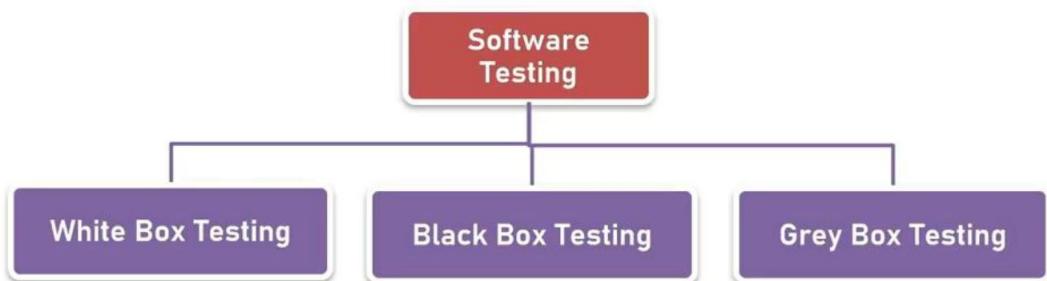
System Analysis and Design

□ Types of Software Testing

- White Box Testing
- Black Box Testing
- Grey Box Testing



System Analysis and Design



White Box Testing

- *It is a software testing technique in which internal structure, design and coding of software are tested.*
- *It is done to verify the flow of input-output in the software and to improve design, usability and security.*
- *In this testing, programming skills are required to design test cases.*
- *In white box testing, code is visible to the testers so it is also called clear box testing, open box testing, transparent box testing, and glass box testing.*



White Box Testing

- *It focuses on checking of predefined inputs against expected and desired outputs.*
- *In white box testing various tests are performed, like:*
 - *Path testing*
 - *Loop testing*
 - *Condition testing*
 - *Memory consumption testing*
 - *Performance Testing*



Black Box Testing

- *It is a software testing technique in which the functionalities of software are tested without having knowledge of internal code structure, implementation details and internal paths.*
- *It is entirely based on software requirements and specifications and mainly focuses on input and output of software.*



Black Box Testing

- *For Example, we use an Operating System, a website, or our own custom application and we can perform black box testing by just focusing on the inputs and outputs without knowing the internal code / implementation.*
- *It is also known as behavioral testing or specification based testing.*



Grey Box Testing

- *It is combination of black box and white box testing.*
- *In this technique, the tester tests the software with partial knowledge of the internal working structure.*
- *This type of testing is mainly used in web-based applications as if any error is caught by the tester, then it is resolved in real-time.*
- *It helps to test both presentation layer as well as internal coding structure of a software.*



System Analysis and Design

System Analysis and Design

Types of software testing

Functional Testing

- Unit testing
- Integration testing
- System testing
- Regression testing
- User acceptance testing

Non-Functional Testing

- Performance testing: load, stress, scalability etc.
- Usability testing
- Compatibility testing

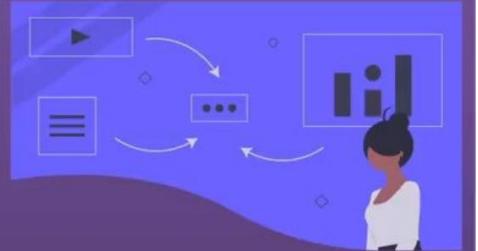
Functional Testing

- *It is a type of black box testing in which the testers identify what an application is supposed to do and runs a series of tests to make sure that the software performs as expected.*
- *The tester will check all the components systematically against requirement specifications. It involves testing of user interface, APIs, Database, Security, Client/Server communication etc.*



Functional Testing

- *The testing can be done either manually or using automation.*
- *Functional testing is executed before non-functional testing.*
- *Functional testing basically involves (Levels of functional testing):*
 - *Unit testing*
 - *Integration testing*
 - *System testing*
 - *Regression testing*
 - *User acceptance testing*



Functional Testing – Unit Testing

- ***Unit Testing:***
 - *In this testing, the tester tests every individual unit or module of the software.*
 - *It is carried out during the coding phase of software development.*
 - *It is typically done by the developer.*
 - *It is done to ensure each module is working correctly.*



Functional Testing – Integration Testing

- **Integration Testing:**
 - *It is the second level of functional testing and is done after the unit testing.*
 - *During the development of the software, two or more modules are integrated with each other to work together as a group. So, in integration testing we test this group of modules as a whole.*
 - *It is done to test the data flow between dependent modules or interaction between two features / modules.*



Functional Testing – System Testing

System Testing:

- It is the third level of functional testing and is done after integration testing.**
- In system testing, the tester validates the complete and fully integrated software product.**
- The purpose of a system test is to evaluate the end-to-end system specifications.**



Functional Testing – User Acceptance Testing

□ User Acceptance Testing:

- It is the last level of functional testing.***
- It verifies whether the software is fulfilling the initial goals according to the requirements.***
- It performed by the end user or the client to verify / accept the software before the deployment of software for production.***



Non-Functional Testing

- *In this testing, the tester tests the software for non-functional aspects like security, efficiency, usability, flexibility, portability and much more.*
- *It is performed to optimize the software for client / users before production.*
- *It helps to improve the product behavior and technology used.*



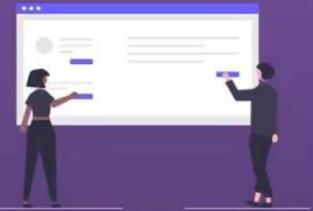
Non-Functional Testing – Performance Testing

- *Performance testing refers to look out performance symptoms and issues of the software like slow response, long load time, easy overloading, poor scalability, high CPU and disk usage etc.*
- *We also test the software with work load in order to measure performance as the work load increases.*



Non-Functional Testing – Usability Testing

- *Usability Testing is also known as User Experience Testing and is a method to measure how easy and user-friendly a software is.*
- *A few number of target users use the software to expose usability defects.*
- *It mainly focuses on user's ease of using the software, flexibility of software and ability of software to meet its objective.*



Non-Functional Testing – Compatibility Testing

- *In Compatibility testing, we check the functionality of software on different hardware platforms, networks, and browsers to find out whether it is working as expected or not.*
- *The software must be adaptive for all kind of the users irrespective of what device they use, that is what we test in compatibility testing.*



System Analysis and Design

System Analysis and Design

- **Software design**
- **Objective of software design**
- **Software design principles**
 - *Problem partitioning / Decomposition*
 - *Abstraction*
 - *Modularity*
 - *Top-down & bottom-up approach*



System Analysis and Design

- **Software design**
- **Objective of software design**
- **Software design principles**
 - *Problem partitioning / Decomposition*
 - *Abstraction*
 - *Modularity*
 - *Top-down & bottom-up approach*



Software Design

- ***Software design is the process of designing the elements of a software such as the architecture, modules and components, different interfaces of those components and the data that goes into it.***
- ***In Software designing we transform user requirements into some suitable form, which helps the programmer in software coding and implementation.***
- ***In software design, the development is divided into several sub-activities, which coordinate with each other to achieve the main objective of software development.***



Software Design Principles

- *There are some principles for good system design:*
 - *Problem partitioning / Decomposition: It refers to break down a complex system into components or smaller sub-system.*
 - *Abstraction: It means to describe the external behavior of that component without bothering with the internal details that produce the behavior.*



Software Design Principles

- *There are some principles for good system design:*
 - *Modularity: It means the division of software into separate modules which are differently addressed and are integrated later on in to obtain the complete functional software.*
 - *Top-down & bottom-up approach: The top-down approach starts with the identification of the main components and then decomposing them into more detailed sub-components. The bottom-up approach begins with the lower details and moves towards upward to form one main component.*