SOFTWARE ENGINEERING

* Software entails writing a program using a programming language using rules or semantics
* A program is a subset of the software
* A program can be graduated to be a software if it has:

An associated documentation: to elevate a program into a software document include:

* Analysis specification documents (S.R.S) it must include:
* Format specifications including objectives
* Context diagram - represents all the entities that will be interacting with the software, listing each personal entity will be doing as they interact with the software
* Data flow diagrams
* Design documents
* Flowcharts
* E/R diagram
* Implementation documentation
* Source code comments (description of inputs and outputs)
* Gross reference listings
* Test plan documents
* Test data
* Operating procedures
* User manuals to assist the system users (beginner’s tutorials), reference guide
* Operational manuals – people required to do the installations & administering the admin

SOFTWARE

B. Engineering

A body of knowledge used when buildings things

It consists of:

1. Scheduling - sequencing of activities
2. Cost estimations – resources how much they cost e.g. materials and persons required
3. Creating models – design, build solution modules to know what to build exactly possible solutions
4. Choices – to be able to pick one of the solution modules
5. Implementation of one of the choices you have selected – testing whether you are still within what was required of the product
6. Communication – interactions among team members, interactions with the persons who have sponsored the project
7. Organization – to have an organized way of eventually delivering the project

Solutions must be – correct (having features that ensure correctness), safe (have safety features), cheap(costs), adaptable (supported in different platforms/environments), fast (procedures results within an acceptable time limit), modifiable (easy modified to suite different environments)

Engineering involves planning, understand present and future requirements of the problem you are trying to create the solutions

You must put a lot of thought process in the requirements

Steps in software engineering

1. Describe the problem trying to solve

The description includes:

* What kind of information when data is given
* What is required to be final products

1. Analyzing the problem by way of building the solution method
2. Select one of the methods from the main solutions methods created
3. Specify the solution methods – you must specify the specifications of the features

The objective of software engineering is producing solutions to problems by providing s good quality maintainable software within time and within the budget

A software engineer is expected to use computer science skills to create software products that are of practical use and of economic value.

Software engineers must be morally, ethical, legally responsible for the correctness suitable safety of your eventual product of service

Definition of software engineering

Is the principle whose aim is the production of quality software that is delivered all time within the budget and must satisfy the user requirements

The establishment and use of sound engineering response in order to obtain economically developed software that is seen to be reliable and works efficiently in all real machines is the application of a systematic disciplined quantifiable approach to the development, operation and maintenance of software

Is an engineering discipline that is concerned with all aspects of software production

Why we study software engineering

* the computers, information system & control systems are part of today’s life
* a software engineer will be required to build such systems to make people perform the duties easily

Duties of a software engineering

1. To be able to model and understand complex iterative systems which are part of features of today and to make the iterative systems to work.
2. To be able to access in the identification of approaches which can be used to improve information systems that are used by individuals.
3. To be able to merge the construction of information systems and their components.
4. To ensure that the procedures which need to be followed in the construction of new systems, the modification of the existing systems are in place.
5. Maintenance of operational systems.

Development process of coming up with a software product is referred to as software model process

They are development approaches used in software engineering to produce a software product or service

Each of the process modules/models have got a set of development practices which are performed by a software engineer in a predefined manner

Defined phases of software development

1. Requirements gathering stage – establishing what the user of your software would want to be included as part of the system
2. Requirements analysis – trying to understand and also document the requirements

Software process models

1. Structured approach
2. Object oriented approach

Defined phases of software development (software development life cycle)

1. Requirement elicitation/Requirements gathering/Data gathering

* Requirements is a description of a statement that gives the function and what the software is supposed to do. It also gives the description of the features to be included as part of the software and what conditions must be met/constraints of the system.

1. Requirement analysis

* Software engineer attempting to understand/ document the requirement
* In the process of trying to understand the requirements, there will be a lot of interrogation.
* A document must be produced containing the functions features and conditions specifications written in the **software requirements specification**

1. Design

* Picking up for the specifications to build up the models of solutions and selecting one that should implement
* Involves actualizing the requirements by building solution models

1. Construction/implementation

* You write programs to tie together what you have designed , those a language to write a program to implement the design based on speifications

1. Testing
2. Deliver

* Moving the system from the lab to its intended environment

1. Maintenance

* Upgrading new requirements
* Correcting defects
* Adding new functionalities

## Feasibility study

Before any software is developed you must perform a feasibility study

The study involves:

Technical feasibility – attempting to ask yourself

* 1. Do we have the required technology to develop the system?
* Hence you must understand the features that are being requested from
* What are the conditions the system must meet?
  1. What are the technical constraints of the existing technology
* The database can hold what capacity?
  1. What are the user interface features of this technology?
  2. What is the existing web technology

Social feasibility

* Will the system be acceptable to the persons who will be directly/ indirectly be consumers of the system?
* What will be the effect of the system on potential users?
* What will be the new jobs descriptions of the system?
* How will the new tasks introduced be carried?

Economic feasibility

Trying to establish what will be the cost of developing the system and what are the potential benefits; tangible/intangible.

* Development time
* Material costs
* You have to list the cost of items
* List the tangible/intangible benefits

Market research

* Establishing similar software products/services that are available in the market which are being used for and if, their will my system penetrate?
* Does the company have the ability to build the system?
* Do you have the people to participate in building the system
* Programmer
* Analysts
* Testers
* Do you have the skills?
* Do you have the time?
* Do you have the money?

Feasibility takes the following options

1. Do nothing

* Maybe because the system is in place its okay
* There is no money/technology

1. Build a new system using the new language, database, technology to replace the old system
2. Change the software architecture to be able to meet the new functions to make it easier for the system to use
3. Buy a package (generic) software and customized it to suite the condition that you want

## Project scoping

* Providing the boundaries of your software
* The software will be covering a specific area of the organization
* Put a limit on how much you want to create as part of a system
* Must an organization chart of a system so as to be able to come up with the exact functionalities & conditions of a system

## Project management

* Trying to manage people in your project team, managing resources, managing the time
* Involves managing all resources with the purpose of delivering all project objectives

## Initiators of a new system

1. Direct users of the old system in terms of: complaints, feedback and general upgrades
2. The organization itself after performing strengths, weaknesses, opportunities, threats, is not faced by the organization

* Strengths include:
* Good educated work force
* Good working environment
* Good client base
* Skills
* Technology
* Good budget
* Weaknesses include:
* Customer complaints
* Poor distribution of information
* No morale for employees
* Outreach of market
* Opportunities include:
* New tech products
* New functionalities
* New trainers
* Direct user can complain:
* Poor output forms
* Design of information presented
* Poor structure of output form
* The content of what is to be presented
* General presentation e.g. font sizes, color etc.
* Change can be done to: screen icons, images etc.

1. Generators of the data or information going into the system

* They may also include the need to develop the new system
* The complaints include:
* Input forms
* Format of forms
* Design of users

Business requirements of an organization

A business requirement is a high-level statement that describes the opportunities that ab organization wants to realize by way of giving the goals of the organization, the objectives and describing the needs of the organization

In such a statement, the organization normally gives a statement of the problems which they want to solve so that they do remain in business

While performing the SWOT analysis, an organization may identify some weaknesses associated with the current technology used e.g. Operating system that no longer supports the features, the functions and conditions that are required in the business environment. The hardware can also be constrained because of hardware e.g. memory and processing time, software in place: such as features currently used that are not useful to the organization

1. Personnel

* Personnel may not be able to use the technology being currently used;
* E.g. processes, policies, procedures

Examples of business requirement statements

* The organization wants to upgrade the software/hardware
* Retaining the personnel

The business requirement gives an intention and must be supported by **business case**

**Business case** has the purpose of assisting the decision makers to ensure that the proposed initiative as per the business requirement will add value to the product or services of that organization

**Example of business cases**

1. Improved customer relations
2. Attracting more clients
3. Reduced processing time of information
4. Build reasonable security features so as to secure data
5. To monitor the usage of money or etc.
6. To reduce maintenance costs
7. Reaching out many people

* A business case captures the reasons for initiating a **software project**

Requirements

A requirement represents a description or a statement that a particular user of the system be it a direct or indirect user would like to see included as part of the system that is in use in an organization

This description includes:

* What the system should do (functions)
* The features to be contained in the system
* The conditions the system would be able to capture/the constraints to be contained in the system

Examples of functions as a requirement

Is a statement that describes what the system should do:

They include:

1. The system which displays the names of items in the store
2. Search for an item from a group of items
3. A system that can perform some computations
4. A system that integrates the use of social media
5. A system where the user will be emailed when an order has been purchased

In summary, the requirements can be a function

Example of features as a requirement;

1. A description of things that can be seen on the screen
2. They include:

* Icons and their descriptions
* Images on the screen

1. Screen forms – new generation programming

* They include input forms and output forms

1. Hyperlinks – navigating from one form to the other

FUNCTIONAL REQUIREMENTS

They relay directly to the functioning of a system hence represents the aspects of the system that however is using the system would recognize.

They include;

1. **Business rules**

* A business rule may form part of functional requirements because they represent the policies which organizations or individuals will use to guide the operations of the system.
* Examples:

1. The system should generate average marks of candidates who have at least done 6 subjects
2. The system should generate a receipt for each payment
3. When the customer purchases good worth more than 250,000 he should be given credit of rates 200%
4. **Authentication**

* Each user should authenticate by use of login details to use the system

1. **Authentication levels**

* The system should categorize users into various categories to access so that to assign them priviledges to be able to know what kind of data access, processes to execute and what operarions they should be allowed

1. **Audit tracking**

* Is a functional requirement that a system should be able to show access the system at what time and who did what so as to be able to look for the foot prints of the person who accesses the system

1. **External interface requirement**

* This is a requirement which follows all objects, images, forms that appear on the screen interaction where the user will be using to communicate with the system
* You have to state the description if objects/images

1. **Functions**

* What a system should be doing
* A function represents a set of functional requirements that describe what the system would do e.g. a system should generate reports, the system shall sort items into ascending/descending order
* The system shall search for items using primary keys
* The system shall compute

Functional requirements can further be categorized into the following

1. **User-level facility / facilitation**

* E.g. once the authorization is successful, system should display name of user

1. **System properties functional requirement**

* The system properties category is normally used to implement the security of the data being accessed and also the persons who access the data
* E.g. The system must ensure that personal information will be made available to only authorized persons

1. **Specific algorithms associated with other requirements e.g. Functions**

* E.g. you may require that once the system has performed some computations, if the result of some computations is <= or >=, can instruct it to do something

Non – functional requirements

They cover those aspects of a system that include

1. System performance
2. Quality issues of the system
3. How the system will interface with other systems
4. System security
5. Portability of the systems (if the system can be used in different platforms)
6. Reliability of the system
7. Availability of the system to the uses
8. Aspects which will take care of human factors (e.g. font size, color, scalability)

* Non-functional requirements therefore represent constraints on how the system will do what it claims it will do.
* They represent the environment of the system e.g.
* The system must be developed in java
* The user interface must follow the operating system guidelines
* The system must respond to the user request in 2 Nano seconds
* The system should be capable of integrating the social media tools which should be displayed in the system
* Only authorized users will be allowed to do some things

Requirement gathering/elicitation

* This is creating a comprehensive requirement list that will be used to build a proposed system
* You are finding out the information that will be required in order to build the system
* This information will come from many sources
* **They include**;
  + 1. **What kind of software us currently available including** -OS, languages, what kind of system is used currently, what kind of language was used to develop the old system, what are the current features, function, constraints, weaknesses of the current system e.g. input forms, equipment processes, environment, output
    2. **People using the current system**
* so as to give information of the current system and state how it should be improved
* what do you think should be improved?
* Includes drawing an organization chart to know what each section of the system deals with knowing what kind of data each section is dealing with
* You must interview the top management in the organization because they are the people who control the vision of the organization. They also know the functionalities of the system to be developed should have. They will also give resources in terms of: persons, resources and materials required
  + 1. **You will also interrogate the data in order to develop the system**
* What are the **data items** e.g. attributes, distinguishing keys, constraints
  + 1. **Documentation**
* Some documents in an organization can give you the requirements on what to develop the system
* They include:
* Minutes
* Email communication
* Software requirements specifications
* Design documentations
* Maintenance communications documentations
  + 1. **Procedures**
* The procedures in use in the current system should be identified so as to determine whether they are problems in them

Techniques to be used to get information from above sources (requirement elicitation)

* Interviews
* Questionnaires

1. **Examining how the work is currently being done/observation**

* It pre-supposes that you have spoken to the top management to allow you to go to the offices of the persons you want to watch and recon what other are doing
* **You have to record;**

1. What are people involved in various sources by way of categories of persons e.g. cashiers, accountants
2. What are they doing?
3. Where are they doing it? e.g. on paper
4. What data are they dealing with?
5. How are they performing their work?
6. Why are they doing it?

* Observation method would require you identify the above and work environment; how people do their work currently

Organization chart

* Used to identify the sections/units of the organization
* Used to list the tasks and activities that are formed in those units including the data that are used there in
* Used to know the categories of persons working in the sections
* Used to identify the interrelationships amongst the units

Context Diagram

* This is a diagram that shows the entities/categories of persons or extreme entities that interact with an organization system
* From the organization chart you can construct a context diagram for each category of users list, what they do to interact with the system
* They are known as the business services. Each business service is recorded as an instruction for the user to the system. It also include the output of the system to the user(s)

DIAGRAM!!

Understanding and documenting requirements

Data analysis

* As part of requirements gathering, you need to identify the data items that are of interest for the section of the organization for which you are developing the system
* Name of the data item
* The characteristics of the data item such as :
* **attributes (what is to be stored about that data item)**
* Specifying the data types (integer, real, strings, double)
* Any relationships among the identified data items
* e.g. connection relationship – which is an association between two data items which have no similar attributes
* Category relationship – this is the association between two data items that have similar attributes
* Identify the constraints of each data item e.g. identify keys – individual occurrences of a specified record (primary keys, candidate keys)

DIAGRAM

* Data analysis will require you to draw a **E/R** diagram so as to map it to the model that assists us understanding the structure storing values.

Notations of E/R diagram

1. Rectangle

* Name of the data item
* Will represent a named attribute attached to the relevant data item
* You underline the attribute agreed to be designed the primary key
* Represent the connection relationship and it must be named

b. cardinalities – how many of these can associate with how many of the others give categories in form of sub items then also give their individual attributes

for each entity it will be mapped onto the table

e.g. course (set of attributes)

* After all this you know how to construct the table
* You go back to what was listed e.g. functions, features (user interface, deinterrogation, input forms), conditions

**Features –** input forms, output forms

e.g. requirement – I want a system that will enable us to enter student records.

* There will be an input form for entering student details with the following fields.
* There will be a command button which user will click on the save data
* **Requirement analysis involves building models to represent t your understanding of the listed requirements**

**The models may**

1. Read data relationships and associated constraints. This is covered in principles of data analysis when you will be required to produce the E/R diagram representation of the domain data. The E/R will then be mapped onto a set of relations (tables like) structures with name and a set of column names called **attributes.** This is known as the **database schema.** Each attribute will accept values belonging to a given data type. One or more of these values will have their values as the identifying values. That is their values will be used to distinguish individual records. Each record represents the instance of the data item.
2. The other models behavioral and here you will need to establish the following:
3. What would be the expected values of the item being represented
4. The steps that will move it to that expected state:

* For example, if a requirement recorded is ‘insert details of an item’
* You will need to establish
* What is used to capture the details

1. Will it be field after field?
2. Will the fields be in one form?
3. What is the field width of each field?
4. Do they have some alignment? If yes who decides?
5. How will you know the correct field values that have been entered? That is what will be the validation criteria?
6. What other features need be visible in the insertion form?

* E.g. will you require to know who did the insertion?
* Do you require a data of insertion?
* Do you require authentication before you insert any values?
* What will be the connection between the form and the physical database? What is, what will be used to transfer the entered details to its datastore? E.g. supposing the data item is student with fields (Registration number, Name, Year of Study, Programme)

There will be one for to be used to insert the values say 3 in our case

Each time you enter the values, the form need be allowed to transfer the entered data into a physical record as shown in the table and leave the form blank

Establish the process (behavior that takes the data from the form to the table)

That is, find out if there is a need for a command to do this and establish the form of the command. In most cases the command is represented in form of command button to be labeled as – save, insert, add, delete

The naming must be uniform and agreed by the stakeholders

Then

If the command is agreed then you provide the instructions for the command

For example; **once the fields have been filled, the user clicks (double clicks)** the command button so that the fields are cleared and the data transferred to the database

1. **The process model –** the process model of the analysis captures the development of the algorithm and the recording of these steps

* Algorithms have a beginning, a set of instructions logically connected and an end
* At the end of the instructions a desired solution must be met**.** This is is for valid inputs
* **For example:** you will need instructions on how the transfer of data need be done from the screen from a database.
* Process models also captures: e.g. computations, comparisons, sorting, copying
* Each of these will require that you put down detailed instructions to be followed during the process
* For example: the requirement that the system should compute the average if the entered student marks
* You will need to provide a process model for this to represent your understanding of the requirement
* It can be something like this one;
* Compute average (the process model name)
* Get the student marks
* N/B: you must clear on how the marks are to be obtained by the system
* You may use an array of data structure and describe a looping instruction that will get the values into the system
* E.g. declare an array, say myArray{size} this size can be read or fixed at run time

Initialize loop variables say i>0

while( i<=arraySize)do

read(myarray[i])

++i

End(while)

* The instruction simply says your name, an array of some size and as you transfer the I = o you read in the values into the array

|  |
| --- |
| MyArray[0] |
|  |
|  |
|  |
| MyArray[signed] |

* Each use case will be associated with a category of actirs and this are the people who belong to a uniform category in the organization that are allowed to participate in executing the instruction associated with the use case.
* To be able to describe what happens in the analysis of the use-cases, you will be required to provide a **flow of events.**
* **`**

# Scenario modelling

* You use interactions to understand document requirements
* It’s expected that you have listed the use cases for all the actors describe in the problem description
* The interaction will between the authorized actor and the system
* During the interactions you may discover new requirements which must also be described
* During the interactions you may discover new requirements which must also be described
* Let us presuppose for purposes of our discussion that the following have been identified alongside their use cases

Clerk: presuppose that – log into system, Maintain student records

Teaching staff: presuppose that – log in, manage student marks

Scenario modelling requires to know the actors & what they will do

* Go to individual use cases & try to understand how the use cases can be actualized by each person
* Each use case must be numbered for purpose of analysis, and pick them one at a time
* E.g

1. Log in
   1. Provide a brief description of the use case which represents (task 1) this task describes how the user logs into the system before using the system. A user must be autorotated or authorized
   2. List the category of users authorized to use or access the task e.g. clerk, teaching, staff etc.
   3. Give the flow of events; what is it that must happen to actualize the task

* Basic flow of events (when things are okay) e.g. the task starts when the actor wants to log into the system
* The system will provide you with: log in screen with details asking you to log in the system {log in screen is new information and because it is part of interaction it will be put in the **external interface requirement**}

# External interface requirement

1. There will be a log in screen provided by the system
2. The screen will have in field such as: users, password and category
3. There will be an error message generated by the system when incorrectly logging in
4. There will be a screen of choices of delete or update of which the user will use
5. There will be a screen for entering student details with the following fields
6. Command button pressed by the user to transfer the form to the database

* The user will enter the details on the said screen
* The system validates the entered details

If successful; the system will take the user to the user profile screen

If not successful; the system will generate an error message

* 1. **Preconditions:** is there something that must be placed before you attempt to access the task associated with the use case e.g. all users must have a user account which means the system admin must have created your user account
  2. **Post conditions:** is what happens if your successful, you will get your corresponding screen

1. What are the things that may cause the failure of a given task (alternative flow of events)?

e.g. wrong password, wrong user id, wrong category, you don’t have a user account

## example 2

1. Maintain user records

* Give a description of a task – allows actors or authorized users to maintain the student records e.g add new details, update, delete
* Actors/users – associate a given task with appropriate actors
* Basic flow of events; tasks will start when the clerk wants to update, delete, student records from the system

1. The system will request the user to specify what maintenance activity to do, the user must have a way of providing the list of choices from which the user will select
2. Once the user selects the choices , presuppose;

* Add a student record;

1. Will enable a user to add a student’s record to the system
2. The system provides a screen for entering new student details

* The user fills the forms then press the save key so that the information given is transferred to the system
* In summary all tasks must be listed, users must be listed, any new information discovered must be listed in the section of external interface requirement

# Documenting requirements

* After analyzing requirements, the next stage is to document them in a document that has the force of law
* It will represent what has been agreed between the developer, the financier and other stakeholders. This will be the specifications of the software to the developer (features, conditions and functions)
* It will be known as the software requirement specifications document (S.R.S)
* The SRS document has a structure and a format and depending on the standard you are using, this must be followed
* The one that is popular is the one developed by the IEEE of ACM.
* This SRS can be in the following forms:

1. Can be in a natural language –
2. Graphical or diagrammatic representation –
3. Executable prototypes – small programs to represent the main program
4. Mathematical representation –

* Each requirement that is required in the document must be consistent, correct, non-ambiguous and must be something that can be modified

## Contents of an S.R.S

* + - 1. Introduction – this is the section required to give a description of the overall software requirements for the proposed system
* The attribute of the system in terms of functional requirements
* The conditions in which the system is required to meet
* Must also include the scope of the system (boundaries)
  1. The purpose of the S.R.S document – the document will provide a description of the capabilities that will be provided by the proposed software and must be qualified
* Name of the system
* The S.R.S will contain the various constraints in which the system is required to abide by
* It will contain the variety of staff who took part in developing the system such as the developers, designers, coders, testers and the DBA.
  1. The product features where you will be required to give out a brief description of the tasks in which your system will be performing and the benefits that can be obtained by using this system e.g. Student registration, reporting subsection, performance of the students
  2. Contents of S.R.S. – includes descriptions, acronyms, abbreviations
  3. Reference – any documentation that should be referred to used when developing the system
  4. What the rest of the document should contain – system requirements

1. General description – this gives the overall description of the system, the weaknesses of the system and a proposed solution to those weaknesses. In the proposed system you should put a description in such a way that the actors doing the analysis can be associated with the tasks they are doing, it should also give a brief description of potential valid inputs and a brief of expected outputs
   1. Product perspective – you must state if the system is stand alone, networked, web-based or mobile app
   2. .1. you must include the screen features that will be required for the system
   3. Product functions - Must give a highlight of the description of the system functions the proposed system will be dealing with e.g. manage subjects, manage student, manage marks, manage user account, search, sorting, generate
   4. User characteristics – this describes the persons using the system, this is used to identify the proficiency in computing, testing the computer literacy level of the person who is supposed to be using this system
   5. Constraints in which the system will be subjected to such as the security features of the system, the policies of the system
   6. You have to give the assumptions and dependencies that were identified in development of the system such as the necessary hardware that must be purchased by the company that will support the system
2. Specific requirements – this is where the details of requirements which will enable the designers
   1. The external interface requirements - includes all those features that will be part of the screen
      1. User interface – provide a list of screens to be provided including those which were discovered during analysis and those ones which were specifically mentioned by the stake holders e.g.
3. Log in screen – allows the user to log into the system and a brief description on how the users will interact with it
4. Subject information screen – describe the fields which it will have and how the users will use them
5. Hyperlinks – which will be connecting you to another document or screen but must be described
   * 1. Hardware interface- for example if the system has to interface with the printers
     2. Software interface – database system, MS Word
     3. Communication interface – give what kind of communication interface will be required e.g. this kind of ports
     4. Performance
     5. Logical database requirements – draws the Entity Relationship diagram, database schema, data dictionary
     6. Analysis model – what kind of modelling techniques did you do

e.g. objects, use cases, context diagram, scenario modeling

### SYSTEM DESIGN

Is the process of translating the SRS into the model that will be implemented

You deal with

1. User interface design
2. Data design
3. System architecture (process design which is an algorithm used to implement modules(writing programs))

* Formulating of various modules that will represent instructions that will tell the compiler what to do

1. **User interface design**

* It represents that part of the system where the users will be interacting with your system
* Is part which you have to do it right
* The perception which the users will have to the system will be affected by the interface design
* Will be implementing part of srs of the features which will be included in the system
* They include: output screens, input screen forms
* The user interface design refers to those part of the computer and its software that a computer user will use to hear, see, touch or even talk to the computer
* It is therefore a set of all the things that allow you and the computer to communicate
* This particular one, to be able to design a user interface, the computer scientist (designer)must have some elements of PHIL 104 because you are dealing with human beings (HCL which is **HUMAN COMPUTER INTERACTION**)
* The HCL describes the relationship between computers and the people who use them to perform their daily routines. It deals with the needs of the users and human characteristics of this people and how they can react as a group or individuals to the responses of the computer (any images, behavior, output of the computer as they interact with these machines)
* **some of the things to consider when designing hcl include;**

1. **usability –** any user has the right to be in control to feel he or she is in control of both the hardware and software and the computer is simply a tool that is to assist them to do what they want to do and is not the computer to control the user
2. **user control –** the user has got the right to demand and generate desired responses from the system
3. **compliance –** it represents the fact that the user has got the right to have the system that performs what was promised and what was promised is what is in the SRS

**PRINCIPLES WHEN PERFORMING A HUMAN COMPUTER INTERFACE (user centered designed principles)**

* **understand the business activities your trying to automate**
* must have been given as the business functions or services
* you must relate the business functions to ‘how does the system support this functions’
* the overall objective therefore is to design an interface that will assist the users to perform the services
* for each business function, you make a list on the important features as given in your srs that will be part of how that feature should be supported
* **maximize graphics effectiveness**
* this is because the graphical user interface is easy to learn and they can relate easily to how a user naturally thinks
* if in case the uses supposed to perform a function of making a selection from the list of options, those options should be put in a graphical form and in such a way that the user can be invited to naturally select from the options
* if a user is supposed to fill a screen form, that form should be designed in a graphical way and the user will be invited to fill the form in the most natural manner possible from top to bottom
* if there there are any terminologies supposed to be used as part of the graphics, they should relate as much as possible to the language that is used in that business area
* **use models and prototypes**
* the models are the same kinds of designs (drawing diagrams) to represent the design of the interface
* the use of models is to increase the feedback so that you can refine your design
* **focus on usability issues**
* the interface is expected to include all the tasks that are supposed to exist during a normal interaction between the user and the machine
* **these include; - screen forms ,images, command buttons, icons, hyperlinks**
* all these must be designed with focus on how best the user can make use of these items on the screen
* usability therefore requires that anything on the screen should be readable to the general users i.e. font size, coloration background, text that can be easily read, sizes of the images should be visible, navigation form one part to another

the command buttons must have the right naming which reflect the business services, languages used in the area and uniformity e.g. don’t use both close, quit and exit

* you must be consistent with the naming, If there are many choices to be made, in a given form,
* you must group the choices and create a hierarchy so that you will be moving from one level to another will be introducing finer details. Don’t include all the choices on the screen form
* **User interface should enable users to provide you with feedback**
* **Design an interface that is very transparent** and it should improve the efficiency of the user. Efficiently should be seen as

1. **Should reduce the introduction of errors** when the user is interacting with the system e.g. make sure typing is little by using drop down lists. If in case typing must be done the UI must capture any possibility of errors getting into the system by making sure all the fields are validated t allow any values that don’t belong to the field
2. **User interface should be easy to use**. It should be guiding the user on what is supposed to be done. It means you design the interface (design/writing commands, actions, responses that are consistent and being able to guide the user on how to do some of the things. The interface should be easy to use and learn)

* To be able to make your own system easy to learn, you make sure that the user controls the comma**nd** buttons in the interface, icons/images in the user interface are those that the user in that area understand and will be able to associate with their daily activities
* on screen interactions should be clear, logical and consistent
* you have to show all the commands in a list if those commands are supposed to be used either collectively or individually
* the screen forms should be designed in such a way that the navigations from one part to another should be easy
* if the user is entering data on a screen form, it should be easy to provide flexibility
* the user interface must enhance user productivity so that when the user is using your system he should be faster
* therefore, the system should make sure all tasks to be performed are in an organized manner to make sure commands put in the appropriate area of the screen, the functions to be performed by the user are grouped in an organized way so that the user should reach those functions easily. This organization should resemble the business functions the organization had been using before the new system. This will enhance user productivity
* **make it easy for users to correct errors**
* **e.g.** online help to recover from errors
* the designer should include confirmation messages to avoid destroying what you have done
* the designer should also undo functions if in case things go wrong
* **create ab attractive layout of your user interface**
* get average feedback from the general stakeholders
* have appropriate colors of the background, any highlighting colors should be also clear
* limit animations in your user interface
* use hyperlinks to get to relate topics/documents/systems to whatever that is in use
* display the titles, messages and any command buttons in a very consistent way for all your screen forms
* **use consistent terminologies**

# user interface control features

* they are used for screen designs
* most programming languages have got templates which are normally used to produce screen designs, which you can customize to suite your design
* the designs should be in line with what was captures in specifications
* the following are some of the control features

MENU BAR – it displays options where these options are designed to belong to some hierarchy where you move from the topmost while decomposing the generic tasks into subtasks

* The menu bar may contain icons/command buttons that are used to represent short cuts for executing the commands associated with the software

COMMAND BUTTON - used to initiate for purposes of initiating some actions, this is normally used if you want the user to initiate some action, you will require that user to interact with a command button

* Any command button that is used as part of screen size must be given an appropriate name
* A name that is meaningful and in line with the terminologies that are used in that business area
* If the command button is clicked upon by the user, that click interaction will initiate an action associated with that command
* This is used in many programming languages

DIALOG BOX – used for purposes to allow the user to enter the information about a task he or she is about to form

* You use a dialog box as a design of input form to allow the user to enter information about a task he or she will perform

TEXTBOX - Screen design to display images, messages or provide a user with a place where he or she will enter some text

* It is normally a small box
* In some cases, these textboxes have associated name this there are called **fields**
* If a data field is used it should be in line as part of the properties of the associated data item

LISTBOX – is used as a screen design to display a list of choices that the user can select from

* Normally used when you want to minimize typing errors
* You choose one value from the list

OPTION BUTTON (radio button) – used as a screen design to represent a set of options which could be used by the user depending on the purposes of the action, each option is represented once and the user is supposed to select/tick one of them

* If it is used appropriately the radio buttons will introduce user friendliness to the system or add ease of use of the system

CHECKBOX – used to select more than one item on a list

CALENEDER CONTROL TO – allows the user select the date for its real value

Screen designs require the use of these control features and in addition should be used along side your individual sense of beauty, the taste of graphics so that your user interface will be attractive to the user

You should design screen that are beautiful easy to use and workable

Should also include a feedback mechanism which would encourage the user to always feel is in communication with the system

The opening screen which is normally known as the switchboard should be designed in such a way that the user would be able to know as part of his or her own first interaction, the kind of generative tasks your system would be dealing with

Should also recall or remember that the user interface should allow the users who understand the business domain to be able to work using the system without having to require to read manuals not having several training sessions

The system should;

1. Will require an efficient searching facility
2. Should be able to retrieve information you need from the system fast and timely
3. Should allow you to generate reports easily
4. Should be able to customize the output to suite the needs of the user
5. Should be capable of performing computations, and these computations should not be the worry of the user
6. There are tips and techniques that should be used in design of system design