**INTRODUCTION TO HCI (1/)**

**Human Computer Interaction (HCI) -** The study of the design, evaluation and implementation of interactive computing systems for human use.
A cross-disciplinary area (e.g., engineering, psychology, ergonomics, design) that deals with the theory, design, implementation, and evaluation of the ways that humans use and interact with computing devices.
**<u>3 parts:</u>**
the person, the computer, and the ways they work together.
**Components of HCI**
1. **User** - an individual user, a group of users working together. sight, hearing, touch relay information is vital. Different users form different conceptions about their interactions and have different ways of learning. Cultural and national differences play a part.
2. **Computer** – Any technology ranging from desktop computers, to large scale computer systems. Devices such as mobile phones or VCRs can also be considered to be "computers"
3. **Interaction** - HCI attempts to ensure that they both get on with each other and interact successfully. In order to achieve a usable system, you need to apply what you know about humans and computers, In real systems, the schedule and the budget are important and vital to find a balance between what would be ideal for the users and what is feasible in reality.

**<u>Goals of HCI</u>**
to produce usable and safe systems, as well as functional systems.
**<u>Producing computer systems with good usability:</u>**
1. understand the factors that determine how people use technology
2. develop tools and techniques to enable building suitable systems
3. achieve efficient, effective, and safe interaction
4. put people first

**Usability of HCI** - It is one of the key concepts in HCI. It is concerned with making systems easy to learn and use.
**<u>A usable system is:</u>**
1. easy to learn
2. easy to remember how to use
3. effective to use
4. efficient to use
5. safe to use
6. enjoyable to use

**<u>Factors in HCI</u>**
A. **Organisation Factors**
B. **Environmental Factors**
C. **Health and Safety Factors**
D. **The User**
E. **Comfort Factors**
F. **User Interface**
G. **Task Factors**
H. **Constraints**
I. **System Functionality**
J. **Productivity Factors**

**<u>Disciplines contributing to HCI</u>**
A. **Computer Science**
- technology software
- design, development & maintenance
- User Interface Management Systems (UIMS) & User Interface Development Environments (UIDE)
- prototyping tools
- graphics

B. **Cognitive Psychology**
- information processing
- capabilities limitations
- cooperative working
- performance prediction

C. **Social Psychology**
- Social & organizational structures

D. **Ergonomics/Human Factors**
- hardware design
- display readability

E. **Linguistics**
- Natural language interfaces

F. **Artificial Intelligence**
- Intelligent software

G. **Philosophy, Sociology & Anthropology Computer supported cooperative work (CSCW)**
H. **Engineering & Design**
- graphic design
- engineering principles

**Humans** are limited in their capacity to process information.

**<u>Information is received and responses given via:</u>**
1. visual channel
2. auditory channel
3. haptic channel
4. movement

**<u>VISUAL CHANNEL</u>**
**Human vision** is a highly complex activity with a range of physical and perceptual limitations. It is the primary source of information for the average person. *Vision begins with light*.
The **eye** is a mechanism for receiving light and transforming it into electrical energy.

**Light** is reflected from objects and their image is focused upside down on the back of the eye.

The **receptors** in the eye transform it into electrical signals then passed to the brain.

**<u>THE EYE</u>**
The **cornea and lens** at the front of the eye focus the light into a sharp image on the back of the eye, the **retina**.

The **retina** is light sensitive with two types of photoreceptor: **rods and**
**INTRODUCTION TO HCI (2/)**

**cones.**

**Rods** are highly sensitive to light and allow us to see under a low level Illumination. They are unable to resolve fine detail and are subject to light saturation.

*There are approximately 120 million rods per eye* mainly situated towards the edges of the retina.
*Rods dominate peripheral vision.*

**Cones** are the second type of receptor in the eye. They are less sensitive to light therefore they can tolerate more light.

*There are three types of cone*.
This allows color vision.
*There's approximately 6 million cones, mainly concentrated on the fovea.*

*There is one blind spot where the optic nerve enters the eye.*

The **retina** also has specialized nerve cells called ganglion cells.
**There are two types:**
1**. X-cells**, which are concentrated in the fovea and are responsible for the early detection of pattern
2. **Y-cells** which are more widely distributed in the retina and are responsible for the early detection of movement.

**DESIGN FOCUS -** Humans ability to read or distinguish falls off inversely as the distance from our point of focus increases

**VISUAL PERCEPTION -** How we perceive size and depth, brightness and color.

**Perceiving size and depth -** Visual angle is affected by both the size of the object and its distance from the eye.

**Perceiving brightness** - It is affected by luminance which is the amount of light emitted by an object.
**INTRODUCTION TO HCI (3/)**

**Brightness** is in fact a subjective reaction to levels of light.

**Contrast** is related to **luminance**: it is a function of the luminance of an object and the luminance of its background.

**Color** is usually regarded as being made up of three components: **hue, intensity and saturation.**
**Hue** is determined by the spectral wavelength of the light. Blues have short wavelengths, greens medium and reds long.
**Intensity** is the brightness of the color
**Saturation** is the amount of whiteness in the color.

*Humans can perceive in the region of 7 million different colors.*

**Visual processing** involves the transformation and interpretation of a complete image

*Expectations affect the way an image is perceived.*

*The ability to interpret and exploit our expectations can be used to resolve ambiguity.*

**Optical illusions** highlight the differences between the way things are and the way we perceive them

**READING**
**Stages in Reading process:**
1. the visual pattern of the word on the page is perceived.
2. It is decoded with reference to an internal representation of language.
3. The final stages of language processing include syntactic and semantic analysis and operate on phrases or sentences.

*Adults read approximately 250 words a minute*

*Words are scanned serially, character by character, words can be recognized as quickly as single characters*

**HEARING** – considered secondary to sight
- hearing begins with vibrations in the air or sound waves.
**3 sections:** outer ear, middle ear and inner ear.

**Pitch** is the frequency of the sound. low frequency produces a low pitch. High frequency, a high pitch.

**Loudness** is proportional to the amplitude of the sound; the *frequency remains constant.*

**Timbre** relates to the type of the sound: sounds may have the same pitch and loudness but be made by different instruments

*The human ear can hear frequencies from about 20 Hz to 15 kHz.* It can distinguish frequency changes at low frequencies but is less accurate at high frequencies.

The **auditory system** performs some filtering of the sounds received, allowing us to ignore background noise

*The ear can differentiate quite subtle sound changes and can recognize familiar sounds without concentrating*

**TOUCH** - "Haptic Perception". The third and last of the senses that to consider. It tells us when we touch something hot or cold, and can therefore act as a warning.

**3 types of sensory receptor:**
1. t**hermoreceptors** respond to heat and cold
2. **nociceptors** respond to intense pressure
3. **mechanoreceptors** respond to pressure.
**2 kinds of mechanoreceptor:**
1. **Rapidly adapting mechanoreceptors -** respond to immediate pressure as the skin is indented. It reacts more quickly with increased pressure. However, they stop responding if continuous pressure is applied.
• **Slowly adapting mechanoreceptors** - respond to continuously applied pressure.

A second aspect of haptic perception is **kinesthesis**: awareness of the position of the body and limbs.
**3 types:**
1. **rapidly adapting**, which respond
when a limb is moved in a particular direction
2. **slowly adapting**, which respond to both movement and static position

3. **positional receptors**, which only
respond when a limb is in a static position.

## MOVEMENT

**Motor control** is the way we move affects our interaction with computers.

**Movement time** is dependent largely on the physical characteristics of the subjects: their age and fitness.

**Reaction time** varies according to the sensory channel through which the stimulus is received.

A second measure of motor skill is **accuracy** and is dependent on the task and the user.

**Speed and accuracy of movement** are important considerations in the design of interactive systems

*Users will find it more difficult to manipulate small objects, targets should generally be as large as possible and the distance to be moved as small as possible.*

**HUMAN MEMORY** - Storing all our factual knowledge, our memory contains our knowledge of actions or procedures

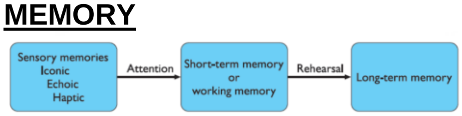
**Information is stored in memory:**

1. sensory memory
2. short-term (working) memory
3. long-term memory

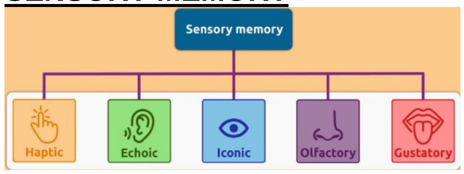
*It allows us to repeat actions, to use language, and to use new information received via our senses.*

*It also gives us our sense of identity, by preserving information from
our past experiences.*

## MODEL STRUCTURE OF MEMORY




## SENSORY MEMORY




The **sensory memories** act as buffers for stimuli received through the
senses.

It exists for each sensory channel: **iconic memory** for visual stimuli, **echoic memory** for aural stimuli and **haptic memory** for touch.

**Attention** is the concentration of the mind on one out of a number of competing stimuli or thoughts.

*Information received by sensory memories is quickly passed into a more permanent memory store, or overwritten and lost.*

**Short-term memory** or working memory acts as a 'scratch-pad' for temporary recall of information.
- It can be accessed rapidly
- Decays rapidly. information can only
be held there temporarily, in the order of 200 ms.
- It has a limited capacity

**Patterns** can be useful as aids to memory

**Recency effect** is where recall of the last words presented is better than recall of those in the middle.

**Long Term Memory -** Main memory resource where we store factual information, experiential knowledge, procedural rules of behavior
- It is huge, if not unlimited in capacity.
- It has a relatively slow access time of
approximately a tenth of a second.
- Forgetting occurs more slowly in long-term memory.

**2 types of long-term memory:**

1. **Episodic memory** represents our memory of events and experiences in a serial form.

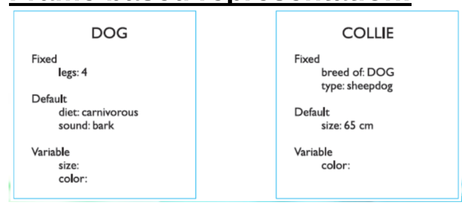
• **Semantic memory**, on the other hand, is a structured record of facts, concepts and skills that we have acquired.

Long-term memory may store information in a semantic network

*Items are associated to each other in classes and may inherit attributes from parent classes.*

**Frames and Scripts -** Models on how to organize information into data structures.

**Frame based representation:**




**Scripts** attempt to model the representation of stereotypical knowledge about situations. It represents this default or stereotypical information

**John took his dog to the surgery. After seeing the vet, he left.**

**ELEMENTS OF A SCRIPT:**

1. Entry conditions. Conditions that must be satisfied for the script to be activated.
2. Result. Conditions that will be true after the script is terminated.
3. Props. Objects involved in the events described in the script.
4. Roles. Actions performed by particular participants.
5. Scenes. The sequences of events that occur.
6. Tracks. A variation on the general pattern representing an alternative scenario.

**3 main activities related to long-term memory:**

1. Storage or remembering of information
2. Forgetting
3. Information retrieval

*Information from short-term memory is stored in long-term memory by rehearsal.*

According to the study of Ebbinghaus, the amount learned was directly proportional to the amount of time spent learning, which is also known as the **total time hypothesis.**

*Repetition is not enough to learn information if not meaningful and familiar.*

It is difficult to remember a set of words representing concepts than a set of words representing objects

**Information is processed and applied:**
1. reasoning
2. problem solving
3. skill acquisition
4. error

**Reasoning** is the process by which we use the knowledge we have to draw conclusions or infer something new about the domain of interest.

**Types of reasoning:**
1. **Deductive** - reasoning derives the logically necessary conclusion from the given premises.
2. **Inductive** - is generalizing from cases we have seen to infer information about cases we have not seen.
3. **Abductive** - reasons from a fact to the action or state that caused it.

**PROBLEM SOLVING -** It is the process of finding a solution to an unfamiliar task, using the knowledge we have.

It is characterized by the ability to adapt the information we have to deal with new situations.

**Views on how people solve problems:**
1. Gestalt Theory
2. Problem Space Theory
3. Analogy

**Gestalt Theory** - problem solving is both productive and reproductive
- productive problem solving involves insight and restructuring of the problem while reproductive solving problem draws on previous experiences.

**Problem Space Theory -** problem has an initial state and a goal state and people use the operators to move from the former to the latter.

**Heuristics** such as means-ends analysis are employed to select appropriate operators to

reach the goal.

**Means–ends analysis** the initial state is compared with the goal state and an operator chosen to reduce the difference between the two.

**Problem Space Theory** -
operates within the constraints of the human processing system such as the limited capacity of short-term memory
- their General Problem Solver model which is based on it, have largely been applied to problem solving in well-defined domains, for example solving puzzles.

**Analogy in Problem Solving**
**Analogy** is a cognitive process of transferring information or meaning from a particular subject to another, or a linguistic expression corresponding to such a process.

Done by mapping knowledge relating to a similar known domain to the new problem – called **analogical mapping.**

Similarities between the known domain and the new one are noted and operators from the known domain are transferred to the new one.

**PHYCHOLOGY AND THE DESIGN OF INTERACTIVE SYSTEM**
*recognition is easier than recall*

**Guidelines** - Human cognitive and perceptual process cannot be all directly applied to design due to being partial and simplistic.

**Models to support design -** Psychological theory has led to the development of analytic and predictive models of user behavior.

**Techniques for evaluation** - Psychology also provides a range of empirical techniques which we can employ to evaluate our designs and our systems.

*Human as an information processor, receiving inputs from the world*
*Information is received through the senses particularly*

*Information is stored in memory, either temporarily in sensory or working memory, or permanently in long-term memory.*

*Human perception and cognition are complex and sophisticated but they are not without their limitations.*

*An understanding of the capabilities and limitations of the human as information processor can help us to design interactive systems.*

*The principles, guidelines and models derived from cognitive psychology are invaluable tools for the designer of interactive systems.*

A **computer** system comprises various elements, each of which affects the user of the system.
- Input devices for interactive use, allowing text entry, drawing and selection from the screen:
− text entry: traditional keyboard, phone text entry, speech and handwriting
− pointing: principally the mouse, but also touchpad, stylus and others
− 3D interaction devices.

**Computer Standard Input Devices:**
1. Keyboard Devices
2. Point and Draw Devices
3. Digitizer
4. Data Scanning Devices
5. Microphone

6. Electronic cards Readers

**Output display devices for interactive use:**
− different types of screen mostly can be used to view images, video, or text.
− devices which can be utilized for the presentation of information in visual form or tactile form.
− common applications for electronic visual displays are television sets or computer monitors.
Standard Output Devices:
1. Monitor Screens
2. Printers
3. Plotters
4. Projectors

**Handwriting Recognition -** also known as handwritten text recognition, is the ability of a computer to receive and interpret intelligible handwritten input from sources
- a method of text entry.
- the most significant information in handwriting is not in the letter shape itself but in the stroke information – the way in which the letter is drawn.

**SPEECH RECOGNITION** - also known as automatic speech recognition (ASR), computer speech recognition, or speech-to-text, is a capability which enables a program to process human speech into a written format.
- commonly confused with voice recognition.
- focuses on the translation of speech from a verbal format to a text one
- considered to be one of the most complex areas of computer science – involving linguistics, mathematics and statistics.

**Speech Recognition most commonly used methods**:
**1. Natural language processing (NLP) -** Area of artificial intelligence which focuses on the interaction between humans and machines through language through speech and text.
**2. Hidden Markov Models (HMM) -** Utilized as sequence models within speech recognition, assigning labels to each unit
**3. N-grams** - the simplest type of language model (LM), which assigns probabilities to sentences or phrases.
**4. Neural networks** - Process training data by mimicking the interconnectivity of the human brain through layers of nodes.
5. **Speaker Diarization (SD)** - Algorithms identify and segment speech by speaker identity. This helps programs better distinguish individuals in a conversation and is applied at call centers distinguishing customers and sales agents.

**Display Devices -** majority of interactive computer systems would be unthinkable without some display screen.
**Types of electronic displays:**
**1. Cathode-Ray Tube(CRT) -** a technology which is used in traditional computer monitor and television.
- a type of vacuum tube that displays images when an electron beam collides on the radiant surface.
**2. Color CRT Monitor**
**3. Liquid crystal display(LCD) -** depends upon the light modulating properties of liquid crystals.
- used in watches and portable computers.
- requires an AC power supply so it is difficult to use it in circuits.
**4. Light Emitting Diode(LED) -** a device which emits when current passes through it. A semiconductor device.
- consumes more power compared to LCD.
- used on TV, smartphones, motors
- powerful in structure, so they are capable of withstanding mechanical pressure. LED also works at high temperatures.
**5. Direct View Storage Tubes(DVST) -** used to store the picture information as a charge distribution behind the phosphor-coated screen.
- common advantages are less time consuming, no refreshing required, high resolution and less cost.
**6. Plasma Display -** a type of flat panel display which uses tiny plasma cells. It is also known as the Gas-Discharge display.
**7. 3D Display -** also called stereoscope display technology. This is capable of bringing depth perception to the viewer.

**Virtual Reality Interaction -** an extension of how people interact with the computing systems using a graphical user interface (GUI).
- The user loses the ability to directly observe the input devices
- Employed in a number of industries and is powering many different types of interactions.
**Common Applications:**
1. Flight simulators
2. Architectural walk-throughs
3. Design - interference testing
4. Teleoperation of robots in dangerous (Chernobyl) or distant (Mars) locations
5. Medical X-ray
6. Remote surgery
7. Psychotherapy (e.g. fear of heights)
8. Interactive microscopy

**Virtual reality (VR)** systems and various forms of 3D visualization require you to navigate and interact in a three-dimensional space.
- Users need to navigate through these spaces and manipulate the virtual objects

**DEVICES FOR VR AND 3D INTERACTION**
**1. Cockpit and Virtual Controls**
- used by helicopter and aircraft pilots
- The user manipulates these virtual controls using an ordinary mouse
**2. 3D Mouse** – you can pick it up, move it in three dimensions, rotate the mouse and tip it forward and backward.

**3. Data Glove -** consisting of a lycra glove with optical fibers laid along the
fingers, it detects the joint angles of the fingers and thumb.
- It is very easy to use, and is potentially very powerful and expressive.

**4. VR Helmets** - have two purposes: (i) they display the 3D world to each eye and (ii) they allow the user's head position to be tracked.
• As the user's head moves around the user ought to see different parts of the scene.

**5. Full-body tracking -** made possible through the use of technological devices which are attached to the feet and the hands of the user, which
enables the tracking of their position in the space.

**6. VR Caves -** an immersive virtual reality environment where projectors
are directed to between three and six of the walls of a room-sized cube.
- user can look all around and see the
virtual world surrounding them.

**LIMITATIONS ON INTERACTIVE PERFORMANCE**:

**1. Computation bound**. The system should be designed so that long
delays are not in the middle of interaction

**2. Storage channel bound**. The speed of memory access can interfere with interactive performance. If there is plenty of raw computation power and the system is held up solely by memory

**3. Graphics bound**. The most common bottleneck for many modern
interfaces.

**4. Network Capacity**. Most computers are linked by networks and this can mean using shared files on a remote machine. When accessing such files, it can be the speed of the network rather than that of the memory which limits performance.

**NETWORK COMPUTING** -
increased computing power and memory enabled the people to have more extensive, faster and easier access to information
- Networks who operate over large distances, and the transmission of information may take some time, which affects the response time of the system and hence the nature of the activity.

**Processing speed is limited by various factors:** computation, memory access, graphics and network delays.

**Interaction models** help us to understand what is going on in the interaction between user and system.
• They address what the user wants and what the system does.
• **Ergonomics** looks at the physical characteristics of the interaction
& how it influences its effectiveness.
• The interaction takes place in social and organizational context that affects both user and system.

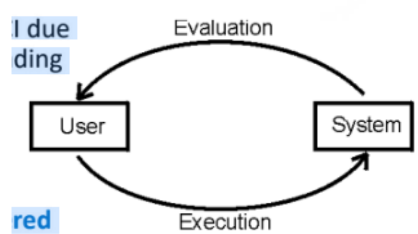
**TERMS OF INTERACTION:**
• **Domain** defines an area of expertise and knowledge in some real-world activity.
• **Tasks** are operations to manipulate the concepts of a domain.
• A **goal** is the desired output from a performed task.
• An **intention** is a specific action required to meet the goal.
• **Task analysis** involves the identification of the problem space.
• The System's language we will refer to as the core language and the User's language we will refer to as the task language.
• The **core language** describes computational attributes of the domain relevant to the System state, whereas the **task language** describes psychological attributes of the domain relevant to the User state.

**MODEL OF INTERACTION -** use of models help us to understand behavior and complex systems.
- can help us to understand exactly what is going on in the interaction
- provide frameworks to compare different interaction styles and to interaction problems.

**User** and **System** are complex and are very different from each other in terms of communication

**Donald Arthur Norman** - Director of The Design Lab at University of California, San Diego.
- best known for his books on design, especially The Design of Everyday Things.

**Norman's model of interaction** is the most influential in HCI
• Norman's 1988 book the Design of
Everyday Things is one of the first appearance of the phase "user centered design."
• the interaction as a cycle with two major components; execution and evaluation:




**the execution component can be further divided into:**
1. Establishing the goal
2. Forming the intention
3. Specifying the action sequence
4. Executing the action

**The evaluation component is divided into:**
1. Perceiving the system state
2. Interpreting the system state
3. Evaluating the system state with respect to the goals and intentions.

The **user** forms the execution and evaluation in a task domain using **task language** while the **system** responses to the user's action using
**core language**.

The **gulf of execution** is the difference between the user's formulation of the actions to reach the goal and the actions allowed by the system.

**Gregory D. Abowd** is a Professor in the College of Computing at Georgia Tech, USA. Best known for his work in ubiquitous computing, software engineering, and technologies for autism.

**Russel Beale** is a Professor at the School of Computer Science, University of Birmingham, UK. His interests include neural networks, agent technology, data-mining and highly configurable user interfaces.

Abowd and Beale introduced **interaction framework.**
The **interaction framework** is a more realistic description of interaction by including the system explicitly with four main components-- the System, the User, the Input and the Output.

4 main translation or mapping involved in the interaction:
1. Articulation. The user must map the goals to a sequence of actions.
2. Performance. The system (modeled as a finite state machine) interprets the user's actions and performs the correct internal change of state.
3. Presentation. The system presenting new internal state.
4. Observation. The user must correctly map the presentation to task language.

Ergonomics (or human factors) is traditionally the study of the physical characteristics of the interaction:
- how the controls are designed;
- the physical environment in which the interaction takes place;
- the layout and physical qualities of the screen.
• Focuses on user performance and how the interface enhances or detracts from it.
• Touch upon human psychology and system constraints.
• A large and established field, which is closely related from HCI.

**Glass Interfaces vs. Dials and Knobs**
• The traditional machine interface consists of dials and knobs directly wired or piped to the equipment. The interface must provide feedback at two levels:
(1) the user must receive immediate feedback, generated by the interface, that keystrokes and other actions have been received.
(2) the user's actions will have some effect on the equipment controlled by the interface and adequate monitoring must be provided for this.

• For a complex system, a glass interface can be both cheaper and more flexible, and it is easy to show the same information in multiple forms.
• a data value might be given both in a precise numeric field and also in a quick to assimilate graphical form.
• the information is not located in physical space and so context are missing and easy to get lost navigating complex menu systems.

**Interaction** can be seen as a dialog between the computer and the user.
**<u>Common interface styles including:</u>**
1. command line interface
2. Menu-based interface
3. natural language
4. question/answer and query dialog
5. form-fills and spreadsheets
6. WIMP (Windows, Icons, Menus and Pointers)
7. point and click
8. three-dimensional interfaces.
*WIMP interface is the most common and complex.*

**Command Line Interface**
• The first interactive dialog style to be commonly used and still widely used.
• The only way of communicating with the system in some applications
• Powerful in that they offer direct access to system functionality and can be combined to apply a number of tools to the same data.
• Commands must be remembered
• Ideal for expert users than for novices.

**<u>Command Line Interface</u>**
**Interaction Styles**

**Menus**
• The set of options available to the user is displayed on the screen, and selected using the mouse
• Options are visible and are less demanding of the user, relying on recognition rather than recall.
• Provided options still need to be meaningful and logically grouped
• The groupings and naming of menu options then provides the only cue for the user to find the required option.
**Natural Language**
• Considered to be the most attractive means of communicating with computers
• both of speech and written input, is the subject of much interest and research.
• The ambiguity of natural language makes it very difficult for a machine to understand.
**Question/answer and query dialog**
• is a simple mechanism for providinginput to an application in a domain.
• The user is asked a series of questions (mainly with yes/no responses, multiple choice, or codes) and so is led through the interaction step by step. An example is questionnaires.
• Interfaces are easy to learn and use, but are limited in functionality and power.
• Uses query languages to construct queries to retrieve information from a database.
• A specialized example is the web search engine.
**Form-fills and spreadsheets**
• These interfaces are used primarily for data entry
• The user is presented with a display resembling a paper form, with slots to fill in and is based upon actual form.
•

• Spreadsheets are an attractive medium for interaction: the user is free to manipulate values at will.

**The WIMP Interface**

• often simply called windowing systems.

• Stands for windows, icons, menus and pointers (sometimes windows, icons, mice and pull-down menus), and is the default interface style for the majority of interactive computer systems in use today

• Examples include Microsoft Windows for IBM PC compatibles, MacOS for Apple Macintosh compatibles and various X Windows-based systems for UNIX.

**Point-and-click interfaces**

• In most multimedia systems and in web browsers, virtually all actions take only a single click of the mouse button.

• By pointing at a recognizable iconic button some action is performed.

• closely related to the WIMP style.

• has been popularized by world wide web pages, which incorporate all the above types of point-and-click navigation: *highlighted words, maps and iconic buttons.*

**Three-dimensional interfaces**

• VR is only part of a range of 3D techniques available to the interface designer.

• The simplest technique is where ordinary WIMP elements are given a 3D appearance using shading.

• Three-dimensional workspaces give you extra space, but in a more natural way than iconizing windows.

• the user can move about within a simulated 3D world.

**Design**

A simple definition is: "Achieving goals within constraints"

• **Goals:**

1. What is the purpose of the design we are intending to produce?
2. Who is it for?
3. Why do they want it?

• Constraints:

1. What materials must we use?
2. What standards must we adopt?
3. How much can it cost?

**Golden rule of Design:**

1. Understand your materials
- In HCI, the obvious materials are human and the computer
2. Understand computers
- Limitations, capacities, tools, platforms
3. Understand people
- Psychological, social aspects, human error

**The Process of Design:**

**1. Requirements** – what is wanted
- The first stage is establishing what exactly is needed.
- Find out what is currently happening.

**2. Analysis** - The results of observation and interview need to be ordered to bring out key issues and communicate with stages of design.
- Can be used both to represent the situation as it is and also the desired situation.

**3. Design** - The central stage when you move from what you want, to how to do it.
- input from theoretical work is most helpful, including cognitive models, organizational issues and understanding communication.

4. **Iteration and prototyping** - We need to evaluate a design to see how well it is working and where there can be improvements.

5. **Implementation and deployment** - Finally, when we are happy with our design, we need to create it and deploy it.
- It will involve writing code, making hardware, writing documentation and manuals.
- It is often better to have a product that is acceptable but on time and to cost than it is to have one that has perfect interaction but is late and over budget.

**User Focus**

• Know your users
- it is important to be aware that there is rarely one user of a system.
- Over time many people are affected directly or indirectly by a system and these people are called **stakeholders**.

**Scenarios**

• **Scenarios** are stories for design: rich stories of interaction.

• the simplest design representation, but one of the most flexible and powerful.

**Scenarios can be used to:**

**1. Communicate with others –** other designers, clients or users. It is easy to misunderstand each other whilst discussing abstract ideas.

2. **Validate other models –** A detailed scenario can be 'played' against various more formal representations such as task models or dialog and navigation models.

3. **Express dynamics –** Individual screen shots and pictures give you a sense of what a system would look like, but not how it behaves.

**Navigation design** is the discipline of creating, analyzing and implementing ways for users to navigate through a website or app.

**Navigation** plays an integral role in how users interact with and use your products. It is how your user can get from point A to point B and even point C in the least frustrating way possible.

**Levels of Interaction:**

• **Widgets**. will help you know how to use them for a particular selection or action.

• **Screens or windows**. You need to find things on the screen, understand the logical grouping of buttons.

• **Navigation within the application**. You need to be able to understand what will happen when a button is pressed

**• Environment**. The word processor has to read documents from disk. You swap between applications, perhaps cut and paste.

**Navigation Design**

**Local Structure** - looking from one screen or page out.

**Local navigation** is used to access lower levels in a structure, below the main navigation pages.

• The term “local” implies “within a given category.”
• On a given page, local navigation generally shows other options at the same level of a hierarchy

**Global Structure** - structure of site, movement between screens.
• The global or main navigation provides an overview and answers important questions users may have when first coming to a site
• The main navigation aids in orientation
• It allows people to switch topics. Users can get to other sections of a site efficiently, or they can reset their navigation path and start over
• It helps when users get interrupted while navigating and reminds visitors where they are in a site. **Main navigation** gives shape to a site. The main navigation defines the boundaries of the site itself.

• **Screen design** refers to the graphic design and layout of user interfaces on displays.
• It is a sub-area of user interface design but is limited to monitors and displays. the focus is on maximizing usability and user experience by making user interaction as simple and efficient as possible.
• The basic principles at the screen level reflect those in other areas of interaction design:
- Ask, Think, and Design.

**<u>Tools for Layout:</u>**
**• Groupings and structure**. If things logically belong together, then we should normally physically group them together.
• **Order of groups and items**. For data entry forms or dialog boxes we should also set up the order in which the tab key moves between fields.
**• Decorations**. Decorative features like font style, and text or background colors
• **Alignment**. For users who read text from left to right, lists of text items should normally be aligned to the left. Numbers aligned to the right.
• **White space**. Also known as “negative space,” is empty space around the content and functional elements of a page.
- let your design breathe by reducing the amount of text and functional elements that users see at once.

**<u>USER AND CONTROL</u>**
**Entering Information**
• The screen consists of places for the user to enter information or select options.
• Alignment is still important.
• Right-justified text for the field labels may be best
• a smaller font can be used for field labels and the labels placed just above and to the left of the field they refer to.
**Knowing what to do**
• Some elements of a screen are passive, simply giving you information; others are active, expecting you to fill them in, or do something to them.
• It is important that the labels and icons on menus are also clear.
**Affordance**
• Affordance is a relationship between a person and a physical or digital object.
• This means that users observe a UI and decide which actions are possible based on their expectations and previous experiences.
• What you must not do is depict a real-world object in a context where its normal affordances do not work!

**<u>APPROPRIATE APPEARANCE</u>**
**Presenting Information**
• The way of presenting information on screen depends on the kind of information and on the purpose for which it is being used.
• Different purposes require different representations.
• No matter how complex the data, the principle of matching presentation to purpose remains.
**Aesthetics and Utility**
• Remember that a pretty interface is not necessarily a good interface.
• Good graphic design and attractive displays can increase users’ satisfaction and thus improve productivity.
• Careful application of aesthetic concepts can also aid comprehensibility.
**Localization/Internationalization**
• The process of making software suitable for different languages and cultures.
• A different resource database is constructed for each language, and so the program can be customized to use

**<u>ITERATION AND PROTOTYPING</u>**
• **Iterative design and prototyping** is a repeating cycle of designing, prototyping, testing, and refining multiple “versions,” or iterations, of a product.
• **Formative evaluation** is the evaluation made if the created prototypes are acceptable and if there is room for improvement.
• **Summative evaluation** is performed at the end to verify whether the product is good enough.

Design in HCI is not just about creating devices or software, but instead is about the whole interaction between people, software and their environment.
• In the case of interaction design the goals are about improving some aspect of work, home or leisure using technology.

• The constraints remind us that the final design will inevitably involve trade-offs between different design issues.

• *A **good designer** understands the natural limitations of ordinary people.*
• ***Design process** starts with understanding the situation as it is and the requirements for change.*
• ***Hierarchy diagrams** can give a logical view of an application, which can be used to design menu or site structures.*
• *The **user dialog** focusses on the flow of user and system actions.*

• *In **screen design and layout**, visual tools that could help us to ensure that the physical structure of our screen emphasized the logical structure of the user interaction.*
• *Iteration is an essential part of virtually any interaction design process because we cannot get things right first time.*

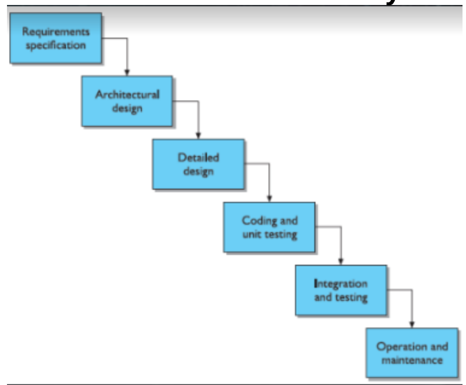
**Software development** is considered as an engineering discipline
• The **software life cycle** is an attempt to identify the activities that occur in software development.
**two main parties:**
1. the customer who requires the use of the product and;
2. the designer who must provide the product.

## WATERFALL MODEL
• The graphical representation is reminiscent of a waterfall, in which each activity naturally leads into the next.
• The analogy of the waterfall is not completely faithful to the real relationship between these activities, but it provides a good starting point for discussing the logical flow of activity.
• It is also referred to as a linear-sequential life cycle model.
• It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.
• The Waterfall model is the earliest SDLC approach that was used for software development.

### Requirements specification
• The aim of the requirement analysis and specification phase is to understand the exact requirements of the customer and document them properly. This phase consists of two different activities.
 Requirement gathering and analysis: Firstly all the requirements regarding the software are gathered from the customer and then the gathered requirements are analyzed.
oThe goal of the analysis part is to remove incompleteness and inconsistencies.

 Requirement specification: These analyzed requirements are documented in a software requirement specification (SRS) document.
* SRS document serves as a contract between the development team and customers.

**The activities in the waterfall model of the software life cycle**




### Architectural Design
• While the requirements specification concentrates on what the system is supposed to do, this next activities concentrate on how the system provides the services expected from it.
• Decomposed the system into components that can either be brought in from existing software products or be developed from scratch independently.
• Describes the interdependencies between separate components and the sharing of resources that will arise between components.

### Detailed Design
• A refinement of the component description provided by the architectural design.
• The goal of this phase is to convert the requirements acquired in the SRS into a format that can be coded in a programming language.
• It includes high-level and detailed design as well as the overall software architecture.
• A Software Design Document is used to document all of this effort (SDD)

### Coding and unit testing
• The detailed design for a component of the system should be in such a form that it is possible to implement it in some executable programming language.
• In the coding phase software design is translated into source code using any suitable programming language. Thus each designed module is coded.
• The aim of the unit testing phase is to check whether each module is working properly or not.

### Integration and testing
• Integration of different modules are undertaken soon after they have been coded and unit tested.
• Integration of various modules is carried out incrementally over a number of steps.
• During each integration step, previously planned modules are added to the partially integrated system and the resultant system is tested.
• Finally, after all the modules have been successfully integrated and tested, the full working system is obtained and system testing is carried out on this.

• System testing consists of three different kinds of testing activities as described below:
**Alpha testing**: Alpha testing is the system testing performed by the development team.
**Beta testing**: Beta testing is the system testing performed by a friendly set of customers.
**Acceptance testing**: After the software has been delivered, the customer performed acceptance testing to determine whether to accept the delivered software or reject it.

### Maintenance
• After product release, all work on the system is considered under the category of maintenance, until such time as a new version of the product demands a total redesign or the product is phased out entirely.
• Maintenance is the most important phase of a software life cycle.

The effort spent on maintenance is 60% of the total effort spent to develop a full software.

• There are basically **three types of maintenance :**

**Corrective Maintenance**: This type of maintenance is carried out to correct errors that were not discovered during the product development phase.

**Perfective Maintenance**: This type of maintenance is carried out to enhance the functionalities of the system based on the customer's request.

**Adaptive Maintenance**: Adaptive maintenance is usually required for porting the software to work in a new environment such as working on a new computer platform or with a new operating system.

**Advantages of Classical Waterfall Model**

This model is very simple and is easy to understand.

Phases in this model are processed one at a time.

Each stage in the model is clearly defined.

This model has very clear and well-understood milestones.

Process, actions and results are very well documented.

Reinforces good habits: define-before- design, design-before-code.

This model works well for smaller projects and projects where requirements are well understood.

**ITERATIVE DESIGN AND PROTOTYPING**

• It is a purposeful design process which tries to overcome the inherent problems of incomplete requirements specification by cycling through several designs, incrementally improving upon the final product with each pass.

• On the technical side, iterative design is described by the use of prototypes, artifacts that simulate or animate some but not all features of the intended system.

**Techniques for Prototyping**

Some of the techniques that are available for producing rapid prototypes:

• **Storyboards**. Probably the simplest notion of a prototype is the storyboard, which is a graphical depiction of the outward appearance of the intended system, without any accompanying system functionality.

It communicates a story through images displayed in a sequence of panels that chronologically maps the story's main events.

**Components of a storyboard:**

• 1. **Scenario**

Storyboards are based on a scenario or a user story.

The persona or role that corresponds to that scenario is clearly specified at the top of the storyboard.

A short text description of the scenario is also included. The description of the scenario or story is clear enough that a team member or stakeholder could understand what is depicted before looking at the visuals.

For example: Corporate buyer, James, needs to replenish office supplies.

• **2. Visuals**

Each step in the scenario is represented visually in a sequence. The steps can be sketches, illustrations, or photos.

Depending on the purpose of the storyboard and on its audience, these images can be quick, low-fidelity drawings or elaborate, high-fidelity artifacts.

Images include details relevant to the story, such as what the user's environment looks like, speech bubbles with quotes from the user, or a sketch of the screen that the user is interacting with.

**• 3. Captions**

Each visual has a corresponding caption.

The caption describes the user's actions, environment, emotional state, device, and so on.

Because the image is the primary content in a storyboard, captions are concise and don't typically exceed two bullet points.

**How to create a storyboard**

Effective storyboarding follows 6 key high-level steps:

**• 1. Gather your data**

First, determine which data will be used in your storyboard — user interviews, usability tests, or site metrics.

It is possible to do a storyboard without real data if you haven't collected data yet or you want to use storyboards as a form of ideation.

• **2. Choose your fidelity level**

Keep in mind the goal and the audience of your artifact.

Use sketches to quickly draw a sequence or communicate a scene to your team during a brainstorming meeting.

Start by having a discussion of the timeline and steps the user will take.

The goal is to form a shared understanding, rather than a polished, refined artifact.

**3. Define the basics**

Define the persona and the scenario or user story represented.

The scenario should be specific and should correspond to a single user path, so that your storyboard doesn't split into multiple directions.

For complex, multipath scenarios, maintain a 1-to-1 rule — one storyboard per one path that the user takes. You'll end up with several storyboards, each outlining a different user path.

oIt's not a flowchart, it's a storyline.

**4. Plan out steps**

Start by writing out the steps and connecting them with arrows before going straight to the storyboard template.

Next, add the emotional state as an icon to each step, as seen below.

This technique will help you start to visualize what each visual frame will include.

5**. Create visuals and add captions**

Illustrate what you're trying to convey.

Add captions as bullet points underneath the visuals to describe

additional context that is not understood at first glance.
Your storyboard should be in an easy-to-modify format, so that you can make changes in further iterations if necessary.

**6. Distribute and iterate**

Distribute your storyboard to your audience, whether it's your internal team or the stakeholders of the project and ask for feedback.
If necessary, iterate over some of these steps to improve the artifact.

**Techniques for Prototyping**

Some of the techniques that are available for producing rapid prototypes:

**• Limited functionality simulations.**

More functionality must be built into the prototype to demonstrate the work that the application will accomplish.
Programming support for simulations means a designer can rapidly build graphical and textual interaction objects and attach some behavior to those objects, which mimics the system's functionality.

**DESIGN RATIONALE -** • A design rationale is the explicit listing of decisions made during a design process, and the reasons why those decisions were made.
• Its primary goal is to support designers by providing a means to record and communicate the argumentation and reasoning behind the design process.
• In computer system, design rationale is the information that explains why a computer system is the way it is, including its structural or architectural description and its functional or behavioral description.
• It relates to an activity of both reflection (doing design rationale) and documentation (creating a design rationale) that occurs throughout the entire life cycle.

Design Rationale helps ensure that UX designers are being intentional with all of their actions and decisions, which can further increase the potential of product success.
• By articulating design rationale, designers are aiming to convince team members (PMs, developers, other designers, etc.), stakeholders and clients to accept her designs or solutions and foster agreement.
• That's why people say a good designer must be a good communicator.

**Design Space Analysis** is an approach to representing design rationale.
• The design space is initially structured by a set of questions representing the major issues of the design.
• It uses a semiformal notation, called QOC (Questions, Options, and Criteria), to represent the design space around an artifact.
• The main constituents of QOC are:
Questions identifying key design issues;
Options providing possible answers to the Questions, and;
Criteria for assessing and comparing the Options.

**QOC notation** - The option which is favorably assessed in terms of a criterion is linked with a solid line
• Whereas negative links have a dashed line.
• The most favorable option is boxed in the diagram.

A **psychological design rationale** proceeds by having the designers of the system record what they believe are the tasks that the system should support and then building the system to support the tasks.
• The designers suggest scenarios for the tasks which will be used to observe new users of the system.
• It aims to make explicit the consequences of a design for the user, given an understanding of what tasks he intends to perform.
• Previously, these psychological consequences were left implicit in the design, though designers would make informal claims about their systems.
• For example, that it is more 'natural' for the user, or easier to learn.

**The first step in the psychological design rationale is to:**

identify the tasks that the proposed system will address and;
to characterize those tasks by questions that the user tries to answer in accomplishing them.

• By documenting the psychological design rationale, we become more aware of the natural evolution of user tasks and the artifact, taking advantage of how consequences of one design can be used to improve later designs.

**• The first step in the psychological design rationale is to:**

identify the tasks that the proposed system will address and;
to characterize those tasks by questions that the user tries to answer in accomplishing them.

• By documenting the psychological design rationale, we become more aware of the natural evolution of user tasks and the artifact, taking advantage of how consequences of one design can be used to improve later designs.

*• Software engineering and the design process relate to interactive system design.*
*• The software engineering life cycle aims to structure design in order to increase the reliability of the design process.*
*• Usability engineering encourages incorporating explicit usability goals within the design process, providing a means by which the*

*product's usability can be judged.*
• *The design process is composed of a series of decisions, which pare down the vast set of potential systems to the one that is actually delivered to the customer.*

• Designing for maximum usability is the goal of interactive systems design.
• Abstract principles offer a way of understanding usability in a more general sense, especially if we can express them within some coherent catalog.
• Design rules in the form of standards and guidelines provide direction for design, in both general and more concrete terms, in order to enhance the interactive properties of the system.
• The essential characteristics of good design are often summarized through 'golden rules' or heuristics.
• Design patterns provide a potentially generative approach to capturing and reusing design knowledge.

**DESIGN STANDARDS**
• **Standards** - something established by authority, custom, or general consent as a model, example, or point of reference.
• Standards for interactive system design are usually set by national or international bodies to ensure compliance with a set of design rules by a large community.
• Standards can apply specifically to either the hardware or the software used to build the interactive system.
• Differing characteristics between hardware and software that affect its design standards:
Underlying theory. Standards for hardware are based on an understanding of physiology or ergonomics/human factors.
Change. Hardware is more difficult and expensive to change than software, which is usually designed to be very flexible.

**Standards institutions /organizations:**
is an organization whose primary function is developing, coordinating, promulgating, revising, amending, reissuing, interpreting, or otherwise contributing to the usefulness of technical standards to those who employ them.

**• Purpose:**
the purpose of developing and adhering to standards is to ensure minimum performance, meet safety requirements, make sure that the product/system/process is consistent and repeatable, and provide for interfacing with other standard- compliant equipment (ensure compatibility).

**The standards organizations for Data Communication and Software Engineering:**
• **International Organization for Standardization (ISO)** creates set of rules and standards for graphics , document exchange , data communication etc.
• **Consultative Committee for International Telephony and Telegraphy (CCITT)** is now standard organization for the United States. CCITT developer's recommended set of rules and standards for telephone and telegraph communication.
• **American National Standard Institute (ANSI)** is primary organization for fostering the development of technology standards in the United States and providing various set of rules and standard for Data Communication.

• **Institute of Electrical and Electronic Engineering (IEEE)** It is US based professional organization of electronic , computer and communication engineering. It provides various set of rules and standard in communication and networking field.
• **Electronic Industries Association(EIA)** This organization establish and recommends industrial standards. EIA has developed the **RS(Recommended Standards**) series of Standards for data and telecommunication.
• **British Standards Institution (BSI)** is the national standards body of the United Kingdom. BSI produces technical standards on a wide range of products and services and also supplies certification and standards-related services to businesses.

**One component of the ISO standard 9241, pertaining to usability specification:**
• **Usability**. The effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments.
• **Effectiveness**. The accuracy and completeness with which specified users can achieve specified goals in particular environments.
• **Efficiency**. The resources expended in relation to the accuracy and completeness of goals achieved.
• **Satisfaction**. The comfort and acceptability of the work system to its users and other people affected by its use.

**DESIGN GUIDELINES**
• are sets of recommendations on how to apply design principles to provide a positive user experience.
• used by designers to judge how to adopt principles such as intuitiveness, learnability, efficiency and consistency so they can create compelling designs and meet and exceed user needs.
• rules of thumb to create work which never frustrates users; such as catering to users who have a wide range of disabilities.
• **Design guidelines fall into several groups, including these:**
1. **Style –** e.g., brand logos, colors
2. **Layout –** e.g., grid or list structure

3. **User interface (UI)** components – e.g., menus, buttons
4. **Text –** e.g., font, tone, labels/fields
5. **Accessibility** – e.g., Aria markup for disabled users
6. **Design Patterns –** e.g., forms

**Shneiderman's 8 Golden Rules of Interface Design**

• **Ben Shneiderman** (born August 21, 1947) is an American computer scientist and professor at the University of Maryland Human-Computer Interaction Lab.

• His work is comparable to other contemporary design thinkers like Don Norman and Jakob Nielsen.
• In his popular book "Designing the User Interface: Strategies for Effective Human-Computer Interaction", Shneiderman reveals his eight golden rules of interface design.

1. **Strive for consistency**

By utilizing familiar icons, colors, menu hierarchy, call-to-actions, and user flows when designing similar situations and sequence of actions.

Standardizing the way information is conveyed ensures users are able to apply knowledge from one click to another; without the need to learn new representations for the same actions.

Consistency plays an important role by helping users become familiar with the digital landscape of your product so they can achieve their goals more easily.

**"Consistency" and "Perceived Stability"** are woven into the design of Apple's Mac OS.

The Mac OS Menu Bar is designed to contain consistent graphicelements regardless of whether it's a version from the 1980's or the 2010's.

2. Enable frequent users to use shortcuts

With increased use comes the demand for quicker methods of completing tasks.

For example, both Windows and Mac provide users with keyboard shortcuts for copying and pasting, so as the user becomes more experienced, they can navigate and operate the user interface more quickly and effortlessly.

3. Offer informative feedback

The user should know where they are at and what is going on at all times. For every action there should be appropriate, human-readable feedback within a reasonable amount of time.

A good example of applying this would be to indicate to the user where they are at in the process when working through a multi-page questionnaire.

A bad example we often see is when an error message shows an error-code instead of a human-readable and meaningful message.

Offer **informative feedback.** Poorly designed error messages often show an error-code that does not mean anything to the user. As a good designer you should always seek to give human-readable and meaningful feedback.

4. Design dialogue to yield closure

Don't keep your users guessing. Tell them what their action has led them to.

For example, users would appreciate a "Thank You" message and a proof of purchase receipt when they've completed an online purchase.

5**. Offer simple error handling**

No one likes to be told they're wrong, especially your users. Systems should be designed to be as fool-proof as possible, but when unavoidable errors occur, ensure users are provided with simple, intuitive step-by-step instructions to solve the problem as quickly and painlessly as possible.

For example, flag the text fields where the users

A gentle **error message** is shown explaining to the user what was happening and why it was happening. It even goes further to reassure the user, telling them that they are in control by explaining that this is due to their own security preference choices.

6. Permit easy reversal of actions
Designers should aim to offer users obvious ways to reverse their actions. These reversals should be permitted at various points whether it occurs after a single action, a data entry or a whole sequence of actions.

As Shneiderman states in his book:
"This feature relieves anxiety, since the user knows that errors can be undone; it thus encourages exploration of unfamiliar options."

When users make an error in providing information during the installation process, they are allowed to go back to the previous step instead of being “punished” by having to start over, or the user can undo a previous action quickly and easily.

7. Support internal locus of control
Allow your users to be the initiators of actions. Give users the sense that they are in full control of events occurring in the digital space.

 Earn their trust as you design the system to behave as they expect.

The user is able to Quit or Force Quit a program if it crashes.

8. Reduce short-term memory load
Human attention is limited and we are only capable of maintaining around five items in our short-term memory at one time.
 Therefore, interfaces should be as simple as possible with proper information hierarchy, and choosing recognition over recall.
 Recognizing something is always easier than recall because recognition involves perceiving cues that help us reach into our vast memory and allowing relevant information to surface.

As humans are only capable of retaining 5 items in our short term memory at one time, the Apple iPhone has stuck with allowing only 4 app icons to sit in the main menu area at the bottom of the screen, regardless of whether it’s the iOS 11 or the iOS 16.

**Norman’s Seven (7) Principles DESIGN GUIDELINES**

• **Donald Arthur Norman** (born December 25, 1935) is an American researcher, professor, and author. Norman is the director of The Design Lab at University of California, San Diego.
• Norman's main idea is that devices,
things, computers, and interfaces should be functional, easy to use, and intuitive.
• His idea is that there are two gulfs
to avoid: the gulf of execution and the gulf of evaluation.

**Norman's main idea** is that devices, things, computers, and interfaces should be functional, easy to use, and intuitive
His idea is that there are two gulfs to avoid: the gulf of execution and the gulf of evaluation.
 This gulf is small where there are only a few roadblocks (like when you're deleting a photo). It's much larger when there are lots of roadblocks, like having lots of fields in a contact form.
 The gulf of evaluation is when a user is expecting feedback from a system, and the system either doesn’t provide the feedback at all or, alternatively, doesn't give the feedback the user is expecting.

1. Discoverability
 Norman describes good discoverability as: “it is possible to determine what actions are possible and the current state of the device”.
 Whenever we engage with an everyday thing such as a TV remote control, or a product like a website or an application, we figure out where and how to perform various functions.
 Through good discoverability, we can consider the different options and choose the one that should work to meet our goal.
 Clear focal points (calls to action, images, and headers); visual hierarchy (content structured in order of priority); and obvious navigation systems all constitute good discoverability and understanding within a design.

This landing page uses focal points, such as calls to action, images and headers; clear navigation; a visual hierarchy, and spacing to make it easier for a user to figure out what actions are possible.

2. Feedback
 Norman describes feedback as, “some way of letting you know that the system is working on your request”.
 When we interact with an everyday thing or product, we need something to communicate the result of our action: feedback. Without any immediate response, we are left wondering if our performed action has had any impact.
 It must be immediate, informative, planned (in an unobtrusive manner), and prioritized.
 Important information should be immediate and clear.
For example, when something goes wrong, an error dialogue forces our attention to critical feedback on what has just happened and what to do next. Subtle feedback, however, can inform us that our action has worked without interrupting our flow.

Feedback is used to update the user on the status of their request or transaction.

3. Conceptual model
 Norman explains, “the design projects all the information needed to create a good conceptual model of the system, leading to an understanding and a feeling of control”.
 A conceptual model is a simple and useful explanation of how something works. For example, a website or application onboarding

experience demonstrates how to use the product or service.

 Other examples of a conceptual model can be found in packaging, instruction manuals, and iconography. Digital interfaces particularly use visual metaphors to help us create an understanding of what can be done; for example, the trash can icon depicts the idea of deleting unwanted files.

 When a conceptual model is not clear, however, then it could go against our mental model (our expectation of how something should work). For example, there are some USB connectors that can only be inserted one way which can conflict with our expectations of the design being reversible.

4. Affordance

 Norman states, "The term affordance refers to the relationship between a physical object and a person". For example, a door handle or a pull chain is shaped like what it controls.

 Affordance is the perceived action and actual properties of an object that help us determine its operation.

 It should be noted that affordance relies on knowledge in the head (what we already understand) and cultural relevance.
Without these properties, then the desired action of an object will be harder to perceive.

 Take flat design, for example, novice users may not immediately understand that certain visual elements can be manipulated.

The thermostat knob is shaped like what it can do i.e., turn to increase the temperature and the default icons on the iPhone dock.

5. Signifiers

 Norman explains, "The term signifier refers to any mark or sound, any perceivable indicator that communicates appropriate behaviour to a person". For example, a button label tells us exactly the kind of action the corresponding control does.

 While signifiers provide clarity on where to perform an action, they also rely on cultural understanding.

 For instance, when Amazon launched its website in India, users mistook the search icon for a ping pong bat. This proves that signifiers must hold relevant meaning so they can be immediately recognised by the target user.

Signifiers, such as buttons and visual cues, tell you where different actions can take place within the design

6. Mapping

 Norman explains, "when mapping uses spatial correspondence between the layout of the controls and the devices being controlled, it is easy to determine how to use them".

 Mapping is the relationship between controls and the effect they have on the world. For example, the arrow buttons on a claw machine correspond to the movements of the toy grabbing mechanical arm.

 Natural mapping enables us to subconsciously change the effects of our world, such as flicking the appropriate light switches.

 When the mapping of something becomes unnatural, then the design becomes more counterintuitive.

 if a natural mapping is not possible, use signifiers to clarify where the relationships exist.

The colour picker depicts the relationship between the controls and the corresponding elements they affect

7. Constraints

 Norman explains, "physical, logical, semantic, and cultural constraints guide actions and ease interpretation".

 Constraints, however, can restrict the kind of interactions that can take place and therefore help reduce the amount of information we process.

 As described by Norman, there are different forms of constraints including physical, semantic, cultural, and logical; each of which can help focus our attention on an important task and help reduce the chance of human error.

 Physical constraints restrict the possible operations; for example, a mouse cursor cannot be moved outside the screen. Semantic limitations are clues to where actions can be performed; for instance, a socket has meaning to where the plug pins can be inserted. Cultural restrictions are social conventions, such as waiting in a queue to be served. And finally, logical constraints help determine the alternatives; for example, scrolling is how we know how to see the rest of the screen.

A complex process for creating and managing an event has been Simplified through the use of small steps which provide the user with just enough information to complete one task at a time.

**Norman's seven fundamental design principles** can help the user determine the answers to their questions; whether they are using an everyday thing or a product. In summary, here are the principles we observed:

 Discoverability makes it easier to understand where to perform actions.

 Feedback communicates the response to our actions.

 Conceptual models are a simple explanation of how something works.
 Affordance is the perceived action of an object.
 Signifiers tell us exactly where to perform an action.
 Mapping is the relationship between the controls and the effect they have.
 Constraints help restrict the kind of interactions that can take place.
A designer can apply each principle as a special strategy to communicate with the user. When all principles are done well, then the conceptual model, in particular, can fit with the user's mental model, thus enabling them to use the product correctly and effectively.

• Describes a recurring problem together with a proven
solution.
• An approach to reusing knowledge about successful design solutions.
• A pattern is an invariant solution to a recurrent
problem within a specific context.
• Examples:
Light on Two Sides of Every Room (architecture)
Go back to a safe place (HCI)

• An interaction design (ID) pattern is a general repeatable solution to a commonly-occurring usability problem in interface design or interaction design.
• Usually consists of the following elements:
Problem: Problems are related to the usage of the system and are relevant to the user or any other stakeholder that is interested in usability.
Use when: a situation (in terms of the tasks, the users
and the context of use) giving rise to a usability problem. This section extends the plain problem-solutions dichotomy by describing situations in which the problems occur.

• Interaction Design Pattern elements:
Principle: a pattern is usually based on one or more
ergonomic principles such as user guidance, or consistency, or error management.
Solution: a proven solution to the problem. A solution
describes only the core of the problem, and the designer has the freedom to implement it in many ways. Other patterns may be needed to solve sub problems.
Why: How and why the pattern actually works, including an analysis of how it may affect certain attributes of usability.

• Interaction Design Pattern elements:
Examples: Each example shows how the pattern has been successfully applied in a real life system. This is often accompanied by a screenshot and a short description.
Implementation: Some patterns provide implementation details.

Principle: Error Management (Safety) (Norman, 1988)

Solution: Maintain a list of user actions and allow users to reverse selected actions.
Each 'action' the user does is recorded and added to a list.
This list then becomes the 'history of user actions' and users
can reverse actions from the last done action to the first one recorded. This is also called a Linear Multi-level Undo.

Why Offering the possibility to always undo actions gives users a comforting feeling. It helps the users feel that they are in
control of the interaction rather than the other way around.
They can explore, make mistakes and easily go some steps
back, which facilitates learning the application's functionality.
It also often eliminates the need for annoying warning
messages since most actions will not be permanent

**WHAT IS EVALUATION?**
• Evaluation is an integral part of the design process
and should take place throughout the design life
cycle.
• An evaluation is a systematic assessment of how
well a project or program is meeting established
goals and objectives.
• Evaluations involve collecting and analyzing data to
inform specific evaluation questions related to
project impacts and performance.
• Analyzing the design and testing the system to
ensure that the expectations and requirements of
the users are met.

**GOALS OF EVALUATION**
• Evaluation has three main goals:
to assess the extent and accessibility of the
system's functionality;
to assess users' experience of the interaction;
to identify any specific problems with the
system.

**EVALUATION THROUGH EXPERT ANALYSIS**
•It depends upon the designer, or a human factors
expert, taking the design and assessing the impact
that it will have upon a typical user.
•It can identify any areas that are likely to cause
difficulties because they violate known cognitive
principles, or ignore accepted empirical results.
•It can be used at any stage in the development
process.
• Relatively cheap, since they do not require user
involvement.

**COGNITIVE WALKTHROUGH**
•It is an attempt to introduce psychological theory into
the informal and subjective walkthrough technique.
• Similar to code walkthrough familiar in software

engineering wherein the sequence represents a
segment of the program code that is stepped through
by the reviewers to check certain characteristics.
(e.g. coding style, variable naming conventions,
procedure calls, etc.)

•In the cognitive walkthrough, the sequence of actions refers to the steps that an interface will require a user to perform in order to accomplish some known task.

• Four questions to be used by an assessor during a
cognitive walkthrough:
1. Will the user try and achieve the right outcome?
(e.g. new smartphone)
2. Will the user notice that the correct action is
available to them? (e.g. TV/DVD remotes)
3. Will the user associate the correct action with the
outcome they expect to achieve? (e.g. Ctrl+Alt+Del)
4. If the correct action is performed, will the user see
that progress is being made towards their intended
outcome? (e.g. feedback mechanism)

• Here's a cognitive walkthrough of the steps involved in editing an image in
**Adobe Photoshop:**
1. Open Photoshop and select the "File" menu. Choose "Open" and navigate to the location of the image you want to edit.
2. Once the image is open, you can begin editing it. There are many tools
and features available in Photoshop for editing images, such as the
"Lasso" tool for selecting specific areas of the image, the "Brush" tool
for painting or drawing on the image, and various filters for altering
the colors and appearance of the image.
3. To make basic adjustments to the image, you can use the "Adjustments" panel on the right side of the screen. This panel contains options for adjusting the brightness, contrast, and color balance of the image.
4. If you want to make more advanced edits to the image, you can use
the "Layers" panel to create separate layers for different parts of the
image. This allows you to make changes to one layer without affecting
the others.
5. When you're happy with your edits, you can save the edited version of
the image by selecting "File" and then "Save As". Choose a location to
save the edited image and click "Save".

**HEURISTIC EVALUATION**
• A heuristic is a guideline or general principle or rule
of thumb that can guide a design decision or be used
to critique a decision that has already been made.
• Heuristic evaluation is a process where experts use
rules of thumb to measure the usability of user
interfaces in independent walkthroughs and report issues.

• Evaluators use established heuristics (e.g., Nielsen-Molich's) and reveal insights that can help design

teams enhance product usability from early in
development.

**NIELSEN'S TEN HEURISTIC**:
1. Visibility of system status.
• Always keep users informed about what is going on,
through appropriate feedback within reasonable time.
For example, if a system operation will take some time,
give an indication of how long and how much is
complete.

2. Match between system and the real world.
• The system should speak the user's language, with
words, phrases and concepts familiar to the user, rather
than system-oriented terms. Follow real-world conventions, making information appear in natural and logical order.

3. User control and freedom.
• Users often choose system functions by mistake and
need a clearly marked 'emergency exit' to leave the
unwanted state without having to go through an
extended dialog. Support undo and redo.

4. Consistency and standards.
• Users should not have to wonder whether words,
situations or actions mean the same thing in different
contexts. Follow platform conventions and accepted standards.

:
5. Error prevention.
• Make it difficult to make errors. Even better than good
error messages is a careful design that prevents a
problem from occurring in the first place.

6. Recognition rather than recall.
• Make objects, actions and options visible. The user
should not have to remember information from one part
of the dialog to another. Instructions for use of the
system should be visible or easily retrievable whenever
appropriate.

7. Flexibility and efficiency of use.
• Allow users to tailor frequent actions. Accelerators –
unseen by the novice user – may often speed up the
interaction for the expert user to such an extent that the

system can cater to both inexperienced and experienced users.

8. Aesthetic and minimalist design.
• Dialogs should not contain information that is irrelevant or rarely needed. Every extra unit of information in a dialog competes with the relevant units of information and diminishes their relative visibility.

9. Help users recognize, diagnose and recover from errors.
• Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
10.Help and documentation.
• Few systems can be used with no instructions so it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large. Once each evaluator has completed their separate assessment, all of the problems are collected and the mean severity ratings calculated. The design team will then determine the ones that are the most important and will receive attention first.

**EVALUATION THROUGH USER PARTICIPATION**
• User participation in evaluation tends to occur in the later stages of development when there is at least a working prototype of the system in place.
• These include empirical or experimental methods, observational methods, query techniques, and methods that use physiological monitoring.
• There are two distinct evaluation styles:
• Laboratory studies : In the first type of evaluation studies, users are taken out of their normal work environment to take part in controlled tests, often in a specialist usability laboratory.
• Field studies : The second type of evaluation takes the designer or evaluator out into the user's work environment in order to observe the system in action

**EMPIRICAL METHODS: EXPERIMENTAL EVALUATION**
•User participation in evaluation tends to occur in the later stages of development when there is at least a working prototype of the system in place.
• controlled evaluation of specific aspects of interactive behaviour
• evaluator chooses hypothesis to be tested
• a number of experimental conditions are considered which differ only in the value of some controlled variable.
• changes in behavioural measure are attributed to different conditions

Experimental Factors:
• Subjects
– who – representative, sufficient sample
• Variables
– things to modify and measure
• Hypothesis
– what you'd like to show
• Experimental design
– how you are going to do it

**OBSERVATIONAL TECHNIQUES**
• A popular way to gather information about actual use of a system is to observe users interacting with it and the evaluator watches and records the users' actions.
• Simple observation is seldom sufficient to determine how well the system meets the users' requirements since it does not always give insight into the their decision processes or attitude.

**QUERY TECHNIQUES**
• An evaluation techniques that relies on asking the user about the interface directly and useful in eliciting detail of the user's view of a system.
• The advantage of such methods is that they get the user's viewpoint directly and may reveal issues that have not been considered by the designer.
• The information gained is necessarily subjective, and may be a 'rationalized' account of events rather than a wholly accurate one.
• There are two main types of query technique: interviews and questionnaires.

**EVALUATION THRU MONITORING PHYSIOLOGICAL RESPONSE**
• An objective usability testing that enables monitoring physiological aspects of computer use.
•It allows the evaluators to not only to see more clearly exactly what users do when they interact with computers, but also to measure how they feel.
• The two areas receiving the most attention to date are eye tracking and physiological measurement.

**FACTORS DISTINGUISHING EVALUATION TECHNIQUES**
•the stage in the cycle at which the evaluation is carried out
•the style of evaluation
•the level of subjectivity or objectivity of the technique
•the type of measures provided
•the information provided
•the immediacy of the response
•the level of interference implied
•the resources required.

SUMMARY
• Evaluation is an integral part of the design process and should take place throughout the design life cycle.
•Its aim is to test the functionality and usability of the design and to identify and rectify any problems.
•It can also try to determine the user's attitude and response to the system.
•It can take place in a specialist laboratory or in the user's workplace, and may or may not involve active participation on the part of the user.

**SUMMARY**

- *A design can be evaluated before any implementation work has started, to minimize the cost of early design errors.*
- *Most techniques for evaluation at this stage are analytic and involve using an expert to assess the design against cognitive and usability principles.*
- *The choice of evaluation method is largely dependent on what is required of the evaluation.*
- *Evaluation methods vary in the stage at which they are commonly used and where they can be used.*
- *Some require more resources in terms of time, equipment and expertise than others.*