# USER EXPERIENCE DESIGN

## 1. What is interactive design?

Interaction Design = art of defining the behaviour of products & systems that a user interacts with

- Facilitates **interaction between user & system**
- = designing interactive products to support the way people communicate and interact in their everyday and working lives (Sharp, Rogers and Preece, 2011)
- = the design of spaces for human communication and interaction (Winograd, 1997)
- = the structure and behaviour of interactive systems. Interaction Designers strive to create meaningful relationships between people and the products and services that they use, from computers to mobile devices to appliances and beyond (Interactive Design Association)

Characteristics of **good design**:

- Trustworthy
- Appropriate
- Smart
- Responsive
- Clever
- Playful
- Pleasurable

Take into account:

- Who are the **users**?
- What **activities** are being carried out?
- Where is the **interaction** taking place?
- Need to match the users' activities & needs

**Users' needs**:

- What are people good at?
- What might help them in how they currently do things?
- What might provide quality user experiences?
- What do people want?
- Get them involved
- **Understanding them**:
  - Use tried & tested user-centered methods
  - Discover hidden wants, needs & desires
    - What people do and why
    - interpretation

**Goals** interaction design:

- develop usable products (easy to learn, effective to use & enjoyable experience)
- involve users in the design process
- **aim:**
  - reduce negative aspects (frustration, annoyance, ...)
  - enhance positive aspects (enjoyment, engagement, ...)
  - develop interactive products that are easy, effective & pleasurable to use (users' perspective)
- **hurdles:**
  - developers see user experience as an obstacle
  - enterprise solutions dictating the user experience
  - lack of budget
  - lack of time
  - too little too late

**User Experience**:

- How a product behaves and is used by people in the real world
  - the way people feel about it and their pleasure and satisfaction when using it, looking at it, holding it, and opening or closing it
  - "every product that is used by someone has a user experience: newspapers, ketchup bottles, reclining armchairs, cardigan sweaters." (Garrett, 2003)
- **Cannot design** a user experience, only design for a user experience

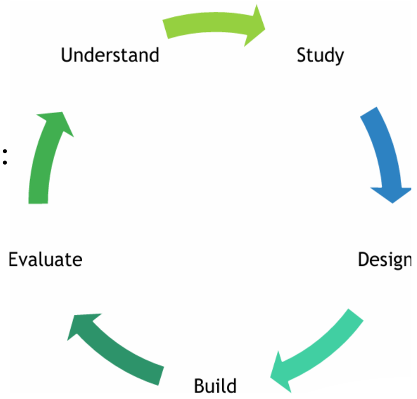
**Holistic experiences**:

- Sensual (sensory engagement)
- Emotional (anger, joy, ...)
- Compositional
- Spatio-temporal

**Process** of interaction design:

1. Establishing requirements
2. Developing alternatives
3. Prototyping
4. Evaluating

**Research & Design Cycle**:



**Characteristics** of interaction design:

- User involvement in development
- Usability & user experience goals need to be identified, documented & agreed at the beginning
- Iteration needed through the core activities

**Why?** Help designers:

- understand how to design interactive products that fit with what people want, need and may desire
- appreciate that one size does not fit all (e.g., teenagers are very different to grown-ups)
- identify any incorrect assumptions they may have about particular user groups (e.g., not all old people want or need big fonts)
- be aware of both people's sensitivities and their capabilities

**Cultural differences**:

- date: d/m/y or m/d/y
- colour: different meanings in different countries
- currency / measurement conversion (pounds, kg, miles, km, EUR, USD, ...)
- high-context ⇔ low-context

**Usability goals**:

- Effective to use
- Efficient to use
- Safe to use
- Have good utility
- Easy to learn
- Easy to remember how to use

**Whitney Quesenberry's 5 E's of Usability**:

- Effective
- Efficient
- Engaging
- Error Tolerant
- Easy to Learn
- Can help plan usability testing

**User experience ⬄ Usability**:

- User Experience = all aspects of the end-user's interaction with the company, its services, and its products
- Usability = a quality attribute that assesses how easy user interfaces are to use
  = method for improving ease-of-use during the design process

User experience **goals**:

- **Desirable aspects**:
  - satisfying
  - enjoyable
  - engaging
  - pleasurable
  - exciting
  - entertaining
  - emotionally fulfilling
  - motivating
  - challenging
  - enhancing sociability
  - supporting creativity
  - cognitively stimulating
  - fun
  - provocative
  - surprising
  - rewarding
  - helpful
- **Undesirable aspects:**
  - boring
  - frustrating
  - making one feel stupid
  - annoying
  - childish
  - cutesy
  - gimmicky
  - annoying
  - childish
  - unpleasant
  - patronizing

User aim: **task completion**

- assessment of user goals
- application of design principles to conceptual & physical designs
- evaluation of design alternatives
- selection and application of evaluation goals met

**Design principles**:

- Generalisable abstractions for thinking about different aspects of design
- The do's and don'ts of interaction design
- What to provide and what not to provide at the interface
- Derived from a mix of theory-based knowledge, experience and common sense

**Visibility**

- Problems arise when we can't see how to use a device
- Hiding functions can be advantageous in UI design
- Make visible only when needed

**Feedback**

- Sending information back to the user about what has been done
- Needs to be immediate and synchronised with user action
- Includes sound, highlighting, animation and combinations of these

**Constraints**

- Restricting the possible actions that can be performed
- Helps prevent user from selecting incorrect options
- Physical objects can be designed to constrain things

**Consistency**

- Design interfaces to have similar operations and use similar elements for similar tasks
- Main benefit is consistent interfaces are easier to learn and use
- Apply the Principle of Least Astonishment
- Aesthetic, Functional, Internal, External

- Inconsistency:
    - Contradiction: same options have different results
    - Irrecularity: different options to do the same thing

**Affordances**:

- Refers to an attribute of an object that allows people to know how to use it (e.g. a mouse button invites pushing, a door handle affords pulling)
- Norman (1988) used the term to discuss the design of everyday objects
- Since it has been much popularised in interaction design to discuss how to design interface objects (e.g. scrollbars to afford moving up and down, icons to afford clicking on)
- Interfaces are virtual and do not have affordances like physical objects
- Norman argues it does not make sense to talk about interfaces in terms of 'real' affordances
- Instead, interfaces are better conceptualized as 'perceived' affordances
    - Learned conventions of arbitrary mappings between action and effect at the interface
    - Some mappings are better than others
- **Virtual affordances**:
    - Web links : convention of mapping between action & effect
    - Tabbed dialogues
- **Physical affordances**:
    - Door handles
    - Cup holders

**Usability principles**

- Similar to design principles, except more prescriptive
- Used mainly as the basis for evaluating systems
- Provide a framework for heuristic evaluation

**Key points**:

- Interaction design is concerned with designing interactive products to support the way people communicate and interact in their everyday and working
- It is concerned with how to create quality user experiences
- It requires taking into account a number of interdependent factors, including context of use, type of activities, cultural differences, and user groups
- It is multidisciplinary, involving many inputs from wide-reaching disciplines and fields

## 2. The process of interaction design

A **process**

= a **goal**-directed problem solving activity informed by intended **use**, **target** domain, **materials**, **cost**, and **feasibility**

= a **creative** activity

= a **decision**-making activity to balance **trade-offs**

= **iterative** (one step forward, two steps back) (knowledge path continually moving forward)

- **4 approaches**:
    - **user centered** design (the user is the expert!)
    - **activity centered** design
    - **systems** design
    - **genius** design (not the best, designing for yourself)

**Involving users**:

- **Expectation management**
  - Realistic expectations
  - No surprises, no disappointments
  - Timely training
  - Communication, but no hype
- **Ownership**
  - Importance of involving users
  - Make the users active stakeholders
  - More likely to forgive or accept problems
  - Can make a big difference to acceptance and success of product
- **Degrees** of user involvement
  - Member of design team:
    - **Full time**: constant input, but lose touch with users
    - **Part time**: patchy input, and very stressful
    - **Short term**: inconsistent across project life
    - **Long term**: consistent, but lose touch with users
  - Newsletters and other dissemination devices
    - Reach **wider selection** of users
    - Need **communication** both ways
  - User involvement after product is released
  - Combination of these approaches

**User centered design** = UCD

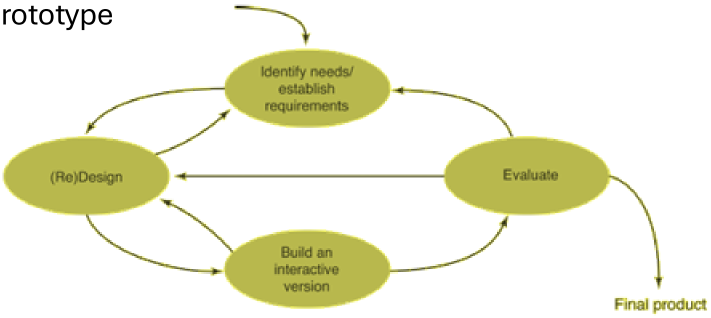
- Pioneered by: **Donald Norman**'s research laboratory at the University of California at San Diego.
- The objective: to develop a **design framework** that enables interaction designers to build more **usable** systems.
- ISO Standard: Human Centered Design Processes for Interactive Systems
- Design should emerge from the user's
  - **tasks**
  - **goals**
  - **environment**
- Focuses on human-centric issues
  - **cognition**
  - **perception**
  - **physical** attributes and conditions (user & environment)
- What?
  - **Early focus on users and tasks**: directly studying cognitive, behavioural & attitudinal characteristics
  - **Empirical measurement**: users' reactions and performance to scenarios, manuals, simulations & prototypes are observed, recorded and analysed
  - **Iterative design**: when problems are found in user testing, fix them and carry out more tests
- Involves:
  - User participation
  - Focus groups
  - Questionnaires
  - Ethnographic observations
  - Walkthroughs
  - Expert evaluations
  - Usability testing

**4 basic activities**:

1. **Establishing requirements**:
    - Who are the users?
    - What support do they need?
2. **Designing alternatives**:
    - there is no 1 solution + 2 designs needed
        a. conceptual design (what the system will do) (first step)
            - easier to edit (save time & cost)
            - can get customer confirmation
            - can come up with better ideas
        b. fysical design (how will the system do it)
3. **Prototyping**:
    - test alternatives
    - low fidelity & disposable prototypes
    - not wasting time designing a working prototype
    - (sketches, storyboards)
4. **Evaluating**

Interaction design **lifecycle model**:

- Exemplifies a user-centered design approach


Identify needs/ establish requirements
(Re)Design
Evaluate
Build an interactive version
Final product

**Practical issues**:

- Who are the **users**?
    - those who **interact**
    - those who **manage direct users**
    - those who **receive output** from the product
    - those who make the **purchasing decision**
    - those who use **competitor's products**
    - **4 categories of user**:
        - **primary**: frequent hands-on
        - **secondary**: occasional or via someone else
        - **tertiary** (indirect): affected by its introduction, or will influence its purchase
        - **facilitator**: person who maintains or develops the design
- What do we mean by '**needs**'?
    - Users rarely know what is possible
    - Users can't tell you what they 'need' to help them achieve their goals
    - Instead, look at **existing tasks**:
        - their context
        - what information do they require?
        - who collaborates to achieve the task?
        - why is the task achieved the way it is?
    - **Envisioned tasks**:
        - can be rooted in existing behaviour
        - can be described as future scenarios
- How to generate **alternatives**?
    - Humans stick to what they know works
    - But considering alternatives is important to '**break out of the box**'

    - Designers are trained to consider alternatives, software people generally are not
    - How do you generate alternatives?
        - **'Flair and creativity'**: research and synthesis
        - **Seek inspiration**: look at similar products or look at very different products
- How to **choose** among alternatives?
    - Evaluation with **users** or with peers (e.g. prototypes)
    - **Technical feasibility**: some not possible
    - **Quality thresholds**: Usability goals lead to usability criteria set early on and check regularly
        - safety: how safe?
        - utility: which functions are superfluous?
        - effectiveness: appropriate support? task coverage, information available
        - efficiency: performance measurements
    - **Generating prototypes**
- How to **integrate** interaction design activities with other **models**?

## 3. Understanding & conceptualizing interaction

Assumption = taking something for granted when it needs further investigation

Claim = stating something to be true when it is still open to question

**Framework** for analysing the **problem space**:

- Are there **problems** with an **existing product** or **user experience**? If so, what are they?
- **Why** do you think there are problems?
- **How** do you think your proposed design ideas might **overcome** these?
- If you are designing for a new user experience, how do you think your proposed design ideas **support**, **change, or extend current ways of doing things**?

**Analysing** the **problem space**:

1. **Plan**: What do you want to learn about the problem
2. **Implement**: develop prototypes or other tools
3. **Measure**: Collect information
4. **Learn**: What have we learnt?

Problem statement = concise description of the problem that needs to be solved

- **no problem** ⇨ **no solution** + **no reason** for a company to exist.
- **Writing** a problem statement:
    - Background of a problem
    - People affected by the problem
    - Impact of the problem on the organization

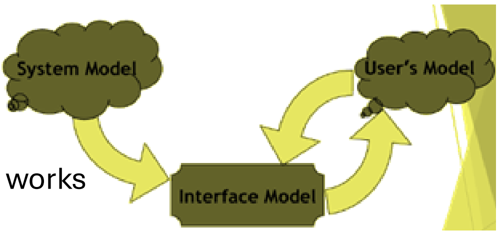
**Benefits of conceptualizing**:

- **Orientation**: enables design teams to ask specific questions about how the conceptual model will be understood
- **Open-minded**: prevents design teams from becoming narrowly focused early on
- **Common ground**: allows design teams to establish a set of commonly agreed terms

**Problem space** ⇨ **design space**:

- Having a good understanding of the problem space can help inform the design space
  - what kind of **interface**, **behavior**, **functionality** to provide
- Before deciding upon these it is important to develop a **conceptual model**
  - **Conceptual model** = high-level description of how a system is organized and operates
  - Enables designers to **straighten out their thinking** before they start laying out their widgets

**Models in UI design**:



- **System model** = how the system works
- **Interface model** = model the system presents to the user
- **User model** = conceptual model = how the user thinks the system works

**Components** of **conceptual model**:

- **Metaphors and analogies**: understand what a product is for and how to use it for an activity
- **Concepts** that people are exposed to through the product
  - task–domain objects
  - their attributes
  - operations
  - (e.g. saving, revisiting, organizing)
- **Relationship** and **mappings** between these concepts
- Many kinds and ways of **classifying them**
  - in terms of **core activities and objects**
  - in terms of **interface metaphors**

**Implications** for **interaction design**:

- Users' conceptual models are often **vague** and **incomplete**
- Interface should match the **understanding** the user already has
- Users prefer simple models (**threshold of indignation** = where user gives up on using the model)

**Formulating** a **conceptual model**:

- What will the **users** be **doing** when carrying out their tasks?
- How will the **system support** these?
- What kind of **interface metaphor**, if any, will be appropriate?
- What kinds of **interaction modes and styles** to use? always keep in mind when making design decisions how the user will understand the underlying conceptual model

**Activity** = describe the components of the conceptual model underlying most online shopping websites (**e.g.** shopping cart, proceed to check-out, 1-click, gift wrapping, cash till?, ...)

**Interface metaphors** = conceptual model instantiated at the interface

- **Conceptualizing** what we are doing
- **Visualizing** an operation
- Interface designed to be **similar to a physical entity** but also **has own properties**
  - e.g. desktop metaphor, search engine
  - Is a blend (Fauconnier & Turner, 2002)
- Can be based on **activity, object or a combination** of both
- Exploit user's **familiar knowledge**, helping them to understand 'the unfamiliar'
- Conjures up the essence of the **unfamiliar activity**, enabling users to leverage of this to understand more aspects of the unfamiliar functionality
- Examples:
  - Paintbrush tool in graphics application
  - Page in word processor
  - Shopping cart on e-commerce site
  - Bookmarks in browser

- **improve the design**, they can
  - put abstract concepts in concrete terms
  - create familiarity
  - trigger emotions
  - draw users' attention
  - motivate users to action

- **Benefits**:
  - Makes **learning new** systems easier
  - Helps users **understand the underlying conceptual** model
  - Can be very **innovative** and enable the realm of computers and their applications to be made more **accessible** to a greater diversity of users
- **Problems**:
  - **Break** conventional and cultural **rules** (e.g. recycle bin placed on desktop)
  - Can **constrain designers** in the way they conceptualize a problem space
  - **Conflict** with design **principles**
  - **Forces users** to only **understand** the system in terms of the metaphor
  - Designers can inadvertently use **bad existing designs** and transfer the bad parts over
  - **Limits designers' imagination** in coming up with new conceptual models
- **Coming up** with interface metaphors:
  - Functional definition (understand how the system works)
  - Identify the users' **problems**
  - Metaphor generation (which ones are implicit in problem description)
  - **Evaluate** the metaphors
- **Principles good** metaphor design:
  - Integration
  - Unpacking
  - Topology
  - Analysis
  - Design

**Interaction types**:

- **Instructing** = issuing **commands** and **selecting** options
  - Where users instruct a system and tell it what to do (e.g. tell the time, print a file, save a file)
  - Very **common** interaction type, underlying a range of devices and systems (e.g. word processors, vending machines)
  - Main benefit is that instructing supports quick and efficient interaction (good for repetitive kinds of actions performed on multiple objects)
- **Conversing** = interacting with a system as if having a **conversation**
  - Underlying model of having a conversation with another human
  - Range from simple voice recognition menu-driven systems to more complex 'natural language' dialogs
  - **E.g.** timetables, search engines, advice-giving systems, help systems, virtual agents, toys and pet robots designed to converse with you
  - **Pro**: allows users, especially novices and technophobes, to interact with the system in a way that is familiar (feels comfortable, at ease and less scared)
  - **Con**: misunderstandings can arise when the system does not know how to parse what the user says
- **Manipulating** = interacting with objects in a **virtual or physical** space by manipulating them
  - Involves **dragging, selecting, opening, closing and zooming** actions on virtual objects
  - Exploit's users' knowledge of how they move and manipulate in the **physical world**

    - Can involve actions using physical controllers (e.g. gaming systems) or air gestures to control the movements of an on screen avatar
    - **Tagged physical objects** (e.g. balls) that are manipulated in a physical world result in physical/digital events (e.g. animation)
    - **Direct manipulation**:
        - Shneiderman (1983) coined the term DM, came from his fascination with computer games at the time
        - captures the idea of "direct manipulation of the object of interest"
            - Continuous representation of objects and actions of interest
            - Physical actions and button pressing instead of issuing commands with complex syntax
            - Rapid reversible actions with immediate feedback on object of interest
        - **Benefits DM**:
            - Novices can learn the basic functionality quickly
            - Experienced users can work extremely rapidly to carry out a wide range of tasks, even defining new functions
            - Intermittent users can retain operational concepts over time
            - Error messages rarely needed
            - Users can immediately see if their actions are furthering their goals and if not do something else
        - **Disadvantages DM**:
            - Some people take the metaphor of direct manipulation too literally
            - Not all tasks can be described by objects & not all actions can be done directly
            - Some tasks are better achieved through delegating (e.g. spell checking)
            - Can become screen space 'gobblers'
            - Not suitable for small graphic displays.
            - Moving a mouse around the screen can be slower than pressing function keys to do same actions
            - May be more difficult to program
            - Compact notations may better suit expert users
- Exploring = **moving** through a virtual environment or a physical space
    - Physical environments with embedded **sensor technologies** (context aware)
- **Which interaction type = best?** Depends:
    - **Direct manipulation**:
        - good for 'doing' types of tasks
        - e.g. designing, drawing, flying, driving, sizing windows
    - **Instructing**:
        - good for repetitive tasks
        - e.g. spell checking, file management
    - **Conversing**:
        - good for children, computer phobic, disabled users and specialised applications
        - e.g. phone services
    - **Hybrid conceptual models**
        - good where different ways of carrying out the same actions is supported at the interface - but can take longer to learn

- **choosing** interaction type:
    - Need to determine **requirements** and user **needs**
    - Take **budget** and other constraints into account
    - Also will depend on **suitability of technology** for activity being supported

**Summary**:

- Important to have a good understanding of the problem space
- Fundamental aspect of interaction design is to develop a conceptual model
- Interaction modes and interface metaphors provide a structure for thinking about which kind of conceptual model to develop
- Interaction styles are specific kinds of interfaces that are instantiated as part of the conceptual model

## 4. Identifying needs & establishing requirements

**What**?

- Understand as much as possible about **users, task & context**
- Produce a stable **set of requirements**

**How**?

- **Data gathering** activities
- **Data analysis** activities
- Expression as '**requirements'**
- All of this is **iterative**

**Why**?

- **complimentary techniques** of data gathering have to be used to negate the limitations of certain techniques
    - interviews:
        - + pick up on non verbal cues
        - + flexibility in questions
        - + can ask more detail (follow-up questions)
        - - difficult analysis
        - - time consuming --> can't do many (smaller group of users reached)
    - questionnaires:
        - + broad audience
        - + easy to analyse
        - - no flexibility, stuck with certain questions
        - - less in depth
- Getting requirements right = crucial

**Establishing requirements:**

- What do users **want**? ≠ What do users '**need'**?
    - Requirements need clarification, refinement, completion, re-scoping
    - Input: requirements document (maybe)
    - Output: stable requirements
- Why '**establish'**?
    - Requirements arise from understanding users' needs
    - Requirements can be justified & related to data

**Kinds of requirements**:

- Requirements need to be:
    - **Clear**
    - **Specific**
    - **Unambiguous**
- **Functional**:
    - What the system should do
    - Historically the main focus of requirements activities
- **Non-functional**:
    - memory size, response time
- **Data:**
    - What kinds of data need to be stored?
    - How will they be stored (e.g. database)?
- **Environment / context of use**:
    - physical: dusty? noisy? vibration? light? heat? humidity? .... (e.g. Information Kiosk, ATM)
    - social: sharing of files, of displays, in paper, across great distances, work individually, privacy for clients
    - organisational: hierarchy, IT department's attitude and remit, user support, communications structure and infrastructure, availability of training
- **Usability**:
    - How well the system needs to work for the users
    - Learnability, throughput, flexibility, attitude
- **Users**: Who are they?
    - Characteristics: ability, background, attitude to computers
    - System use: novice, expert, casual, frequent
    - Novice: step-by-step (prompted), constrained, clear information
    - Expert: flexibility, access/power
    - Frequent: short cuts
    - Casual/infrequent: clear instructions (e.g. menu paths)
- **Unrequirements** = what not to include in the product
    - **Constraints** : generally imposed and immovable
    - **Exclusions** : approaches the business deliberately wants to avoid
- **Personas** = rich descriptions of the users
    - **Research based** documents that describe **typical users**
    - Capture user **characteristics**
    - Should **not be idealised** (don't need users that do everything perfectly the way you designed them → cannot improve your system this way)
    - Bring them to life with a name, characteristics, goals, personal background
    - Develop **multiple** personas
    - ≠ **market segments**
        - Segments cluster individuals into groups
        - Personas create individuals to represent groups
        - Easier to design for Joe, aged 30 than a group 78% male, aged 26-35
        - Marketing data useful starting point for personas
        - Need to add motivations and behaviours

- **Perspectives**:
  - Goal-directed
  - Role-based
  - Engaging
  - Fiction-based
- **Finding participants**: (need to be representative!)
  - Research recruiters
  - Existing customers
  - Friends, family, colleagues
  - Public recruitment
  - Incentives

**Data gatherings** (→ finding **requirements**)

- Need good data:
  - **sufficient**
  - **relevant**
  - **appropriate**
- Interviews:
  - **Props** (e.g. sample scenarios of use, prototypes) can be used in interviews
  - Have it in the **environment** where it is going to be used:
    - They are more at ease
    - They can directly show you things
    - They see the things around them that remind them of problems
  - Good for **exploring** issues
  - But are **time consuming** and may be **infeasible** to visit everyone
  - **Do not impose** ideas or opinions on your users
- Focus groups:
  - Group interviews
  - Good at gaining a **consensus view** and/or **highlighting areas** of conflict
  - But can be **dominated by individuals**
    - group pressure
    - conflicts between individuals with strong opinions and different ideas
    - seniority (senior person says he is right because he has more experience)
- Questionnaires:
  - Often used in conjunction with other techniques
  - Can give **quantitative** or **qualitative** data
  - Good for answering **specific questions** from a **large, dispersed group** of people
- Researching similar products:
  - Good for **prompting requirements**
- Direct observation:
  - Gain insights into stakeholders' tasks
  - Good for understanding the **nature** and **context** of the tasks
  - But, it requires **time** and **commitment** from a member of the design team, and it can result in a **huge amount of data**

    - Types:
        - Watching the users working
            - might misunderstand what they are doing
            - can make wrong assumptions
        - letting the users think out loud while using the product
            - try to understand their cognitive approach to the usage
- **Indirect observation**:
    - Not often used in requirements activity
    - Good for **logging current tasks**
- **Studying documentation**:
    - Never use on its own
    - Procedures and rules are often written down in manuals
    - Good source of data about the steps involved in an activity, & any regulations governing a task
    - Not to be used in isolation
    - Good for understanding legislation, and getting background information
    - No stakeholder time, which is a limiting factor on the other techniques
- **Card sorting**
    - Write down topics and ask users to give **group related items**
    - Can help understand how users connect different types of items
    - Examines how people sort topics
    - **Open sort** = free decisions (not forcing into categories
    - **closed sort** = we give categories and they put the items in those categories
    - **Group sort** = a lot of individuals sorting things on their own
    - **individual sort** = think aloud (why do they sort the way they do)
    - Templates available for analysis
    - Get an idea of what people will expect to see

**Problems with data gathering**:

- It is often nearly **impossible to know who** your "users" will be
    - (except in very narrowly defined situations, such as accounting software)
- **'Real' users**, not managers: traditionally a problem in software engineering, but better now
- **Contradicting/Conflicting** Requirements (users/ stakeholders have different priorities
- **Communication** Problems
- **Undocumented** Processes (tacit knowledge)
- Requirements management: **version control**, **ownership**
- **Stakeholder** Design
- **Bad Requirements**
- **Domain knowledge** distributed and implicit:
    - difficult to dig up and understand
    - knowledge articulation: how do you walk?
- **Availability** of key **people**
- Users often don't **know what they want**, or what they **need**
    - what most users say they want is often not very relevant for what they will actually find useful
- **Political** problems within the organisation
- **Dominance** of certain **stakeholders**
- Economic and business **environment changes**
- Balancing **functional and usability** demands

**Basic guidelines**:

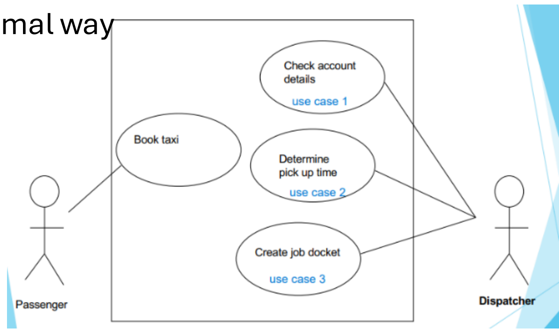
- Focus on identifying the **stakeholders' needs**
- **Involve** all the **stakeholder** groups
- **Involve multiple representatives** from each stakeholder group
- Use a **combination** of data **gathering techniques**
- Support the process with **props** (e.g. prototypes, task descriptions, …)
- Run a **pilot session**
- You will need to compromise on the data you collect and the analysis to be done, but before you can make sensible compromises, you need to know what you'd really like
- Consider carefully how to record the data

**Data interpretation & analysis**:

- Start soon after data gathering session
- Initial interpretation before deeper analysis
- Different approaches emphasize different elements

**Task descriptions**:

- Scenarios = informal narrative story, simple, 'natural', personal, not generalisable
- Use cases = UML technique for describing tasks in a very formal way
  - Also shows alternative courses
  - assume interaction with a system
  - assume detailed understanding of the interaction
  - use case diagram



**Summary**:

- Getting requirements right is crucial
- There are different kinds of requirement, each is significant for interaction design
- The most commonly-used techniques for data gathering are: questionnaires, interviews, focus groups, direct observation, studying documentation and researching similar products
- Scenarios can be used to articulate existing and envisioned work practices

## 5. Data Gathering

**Need for data**:

- **What** is the problem?
  - Being able to identify a problem correctly → easier to find a solution
  - Secondary / tertiary stakeholders to consider
- **Who** are the people involved? Who are you building the solution for?

**5 key issues**:

- **Setting goals**: Decide how to analyze data once collected
- **Identifying participants**: Decide who to gather data from
- **Relationship with participants**:
  - Clear and professional
  - Informed consent when appropriate
- Triangulation
  - = using multiple methods of gathering data
  - Look at data from more than one perspective
- **Pilot studies**:
  - = small trial of main study → make sure you get the information you want
  - + pilot of your analysis

**Data recording**:

- Notes, audio, video, photographs
- Notes + photographs
- Audio + photographs
- Video

**Observation**:

- at any stage in the development
- how do users approach the problem?
- types:
    - **Direct observation** in the **field**
        - Structuring frameworks
        - Degree of participation (insider or outsider)
    - **Direct observation** in **controlled environments**
        - So they do not get distracted
        - Usually you learn a bit less than in the field
    - **Indirect observation**: tracking users' activities
        - Diaries
        - Interaction logging (key strokes, clicks, ...) → often automated, but very informative
        - Web analytics
- Need for **structuring frameworks** to **guide observation**
    - The **person**. Who?
    - The **place**. Where?
    - The **thing**. What?

  Alternative: **The Goetz and LeCompte** (1984) **framework**
    - Who is present?
    - What is their role?
    - What is happening?
    - When does the activity occur?
    - Where is it happening?
    - Why is it happening?
    - How is the activity organized?

  Alternative: **Robson** (2011) **framework**
    - Space
    - Actors
    - Activities
    - Objects
    - Acts
    - Events
    - Time
    - Goals
    - Feelings
- **Inside observers**: participate & engage with the people they observe

  ⇔ **outside observers**: watch from far, don't engage

**Direct observation in controlled environments** = co-operative evaluation = verbal protocols = talk-aloud

- **Concurrent** = observing participants interacting with the system & think aloud at the same time
- **Retrospective** = they think in hindside aloud about their experience with the system
    - Can forget things
    - Can remember things wrong: why did they do what they did?
- **Cued retrospective** = recording what they do and listening/watching afterwards

**Verbal protocols**:

- Can contain users' spoken observations
- Can add to video information or do alone
- Gathering of cognitive information
- Get subjective opinion from comments and tone of voice
- **React** regarding the users: **tone of voice**, **body language**, ...
    - how certain are they?
    - Are they annoyed?

- **Think aloud protocol**: user says out loud what she is thinking when carrying out task
  - **added strain** on user - **divided attention**
  - need to **encourage user** to talk to find out why problems exist
    - 2 people working together who talk to each other
    - allow users to ask questions
    - Prompting
    - experienced user teaching less experienced user
- **Post-event protocol**:
  - obtained after task completed
  - user views video and provides commentary
  - use where tasks require careful concentration and are time-critical
  - only contain recalled information
- **constraints**:
  - artificial
  - labour intensive
  - timings affected so can't benchmark
  - no statistical information
- **benefits**:
  - pinpointing problems
  - why problem occurred
  - catching problems when occur
  - learning how users approach tasks
  - studying attitude
- someone **thinking aloud** about their opinion on a **task ≠ opinion** about **design**
- To get the feedback you want, use
  - **Instructive practice** = give instructions on what they should talk about and
    - **Explain** the technique
    - Have a practice that **demonstrates**
    - Have participants perform a safe, short task first so can help them with the technique
  - **Operant conditioning techniques**
    - Need to encourage the behaviour you want and discourage the behaviour you don't want (e.g. with your dog)
      - **Reinforcement**: reward good behaviour
      - **Extinction**: ignore bad behaviour

Think aloud **analysis**:

- **task-based**
  - look at how users approached task
  - where difficulties lie
  - what can be done
  - aim - identify major problems
- **steps**
  1. **Transcribe** protocols
  2. **Develop** coding scheme (depends on goals of the evaluation) → reliable, clear, usefull
  3. **Categorise** comments and actions (use quotes to illustrate)

     **Card sorting** = method for categorising where users put the items into actual or virtual piles = **pile sorting** = **free grouping** = **partitioning** (categorize objects, tasks, ...)

     - **When** to card sort?
       - When you want the users' mental model to drive the design
       - When you need feedback about the content, terminology, and organisation of the product
       - Can do it for entire sets of information or subsets

- **Types of information**:
  - Overall organisation of the content or tasks
  - Terminology used by users
  - Labels users apply to categorisations
  - Missing objects
  - Unnecessary objects

  (users don't always have optimal mental models, they need to be familiar with the domain of interest)
- **Advantages**:
  - Simple
  - Cheap
  - Quick to execute
  - Established
  - Involves users
  - Provides a good foundation
- **Disadvantages**:
  - Does not consider users' tasks
  - Results may vary
  - Analysis can be time consuming
  - May capture "surface" characteristics only
- **Activities**: (in advance)
  - Identify and define objects
  - Develop user profile
  - Recruit users
  - Prepare documentation
  - Logistics
  - Pilot
- **Types**:
  - **group sort** = multiple people sort individually
  - **individual sort** = 1 individual and facilitator to find out why they actually sort (more in depth for 1 person)
- **Materials required**:
  - 3x5 inch index cards
  - Printer labels
  - Stapler
  - Rubber band
  - Envelopes
  - workspace
- **Changes users can make to the cards**
  - Delete an object
  - Add a new object
  - Rename an object
  - Change a definition
  - Place an object in multiple groups
- **Who**?
  - **Participants**: should meet the user profile
  - **Facilitator**: provide instructions, distribute material, answer questions and collect materials
  - **Videographer**: generally for individual sorts
- **Conduction:**
  1. Welcome the participants
  2. Practice
  3. Card review and sorting
  4. Labeling groups

**Interpreting results**:

- Need to **tidy** up results
  - Items in multiple groups
  - Items in groups on their own
- Look at **changes** the participants made: Was there agreement?

**Choosing combining techniques**, depends on:

- The focus of the study
- The participants involved
- The nature of the technique
- The resources available

**Summary**:

- Three main data gathering methods: interviews, questionnaires, observation
- Five key issues of data gathering: goals, choosing participants, triangulation, participant relationship, pilot
- Interviews may be structured, semi-structured or unstructured
- Questionnaires may be on paper, online or telephone
- Observation may be direct or indirect, in the field or in controlled setting
- Techniques can be combined depending on study focus, participants, nature of technique and available resources

## 6. Interfaces & interaction

**Interface types**:

- **1980s** interfaces
  - Command
  - WIMP/GUI
- **1990s** interfaces
  - Advanced graphical
    - Multimedia
    - virtual reality
    - information visualization)
  - Web
  - Speech (voice)
  - Pen, gesture, and touch
  - Appliance
- **2000s** interfaces
  - Mobile
  - Multimodal
  - virtual reality
  - shareable
  - tangible
  - augmented and mixed reality

**Command-based**

- = **abbreviations typed** in at the prompt to which the **system responds** (e.g. listing files)
- some **hard wired** at **keyboard** ⇔ **assigned** to **keys**
- + efficiënt, precise, fast
- - large overhead to learning set of commands
- UNIX, MS-DOS, Natural language (Star Trek, GenAI)
- **Research and design issues** :
  - Form, name types and structure are key research questions
  - Consistency is most important design principle (e.g. always use first letter of command for shortcuts)
  - Command interfaces popular for web scripting

WIMP = **Windows Icons Menus Pointing** device

- Xerox Star first WIMP → rise to GUIs
- Windows
  - = a means of sharing a device's graphical display resources among multiple applications
  - could be scrolled, stretched, overlapped, opened, closed, and moved around the screen
- Icons = represented applications, objects, commands & tools that were opened when clicked on
- Menus = list of commands or options from which to choose
- Pointing device = a mouse controlling the cursor as a point of entry to the windows, menus, and icons on the screen

GUI = Graphical User Interface

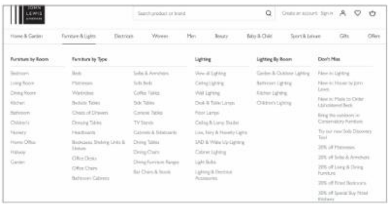
- **Same basic building blocks as WIMPs** but **more varied**
  - Color, 3D, sound, animation, GUIs
  - Many types of menus, icons, windows
- New graphical elements (e.g. toolbars, docks, rollovers)
- The first GUI-centric computer operating model was the Xerox 8010 in **1981**
- The **Macintosh** 128K was the first commercially successful personal computer to use Graphical User Interface

Windows

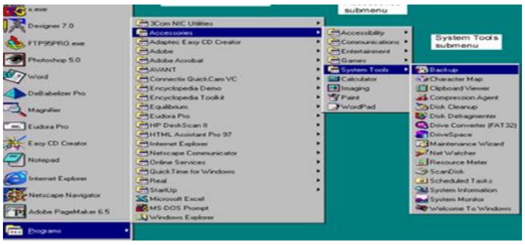
- were invented to **overcome physical constraints** of a computer display
- allow workstation's screen to be **divided** into areas which act like separate input & output channel
- **Scroll bars** within windows also enable **more information** to be viewed
- **Multiple windows** can make it **difficult** to **find desired** one u listing, iconising, shrinking are techniques that help
- Multiple windows can pose **management issues**
  - Managed by the user and by the system
  - Tiled, overlapping, Cascading
- Window Interfaces:
  - Multiple document interface (**MDI**)
    - Application-centric
    - Child windows residing within the primary window
  - Single document interface (**SDI**) = new window for each instance of an application document
  - Tabbed document interface = tabs used to switch between documents
  - Dialogue Boxes: modal/modeless
- **Research & design issues**:
  - Window management: enables users to move fluidly between different windows (and monitors)
  - How to switch attention between windows without getting distracted
  - Design principles of spacing, grouping, and simplicity should be used

## Menus:

- A number of **menu interface styles**
  - flat lists
  - drop-down
  - pop up
  - contextual
  - expanding ones (e.g., scrolling and cascading)
- Flat menus
  - good at displaying a small number of options at the same time and where the size of the display is small
  - but have to nest the lists of options within each other, requiring several steps to get to the list with the desired option
  - moving through previous screens can be tedious
- Expanding menus
  - Enables more options to be shown on a single screen than is possible with a single flat menu
  - More flexible navigation, allowing for selection of options to be done in the same window
  - Most popular are
    - Mega menus



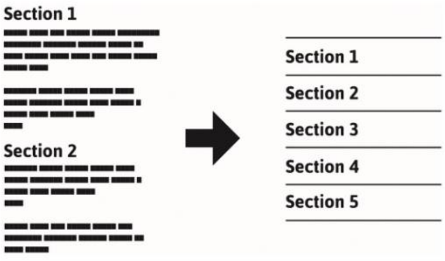
      - = expanding menu that allows you to see multiple options without the need to scroll
      - popular on websites
      - Problem: can accidentally open when hovering over



    - Cascading menus
      - = menus within menus
      - Used in e.g. documents
      - Problems:
        - Difficult to remember where things are
        - they require precise mouse control
        - can result in overshooting or selecting wrong options

    - Contextual menus
      - Provide access to often-used commands that make sense in the context of a current task
      - Appear when the user presses the Control key while clicking on an interface element (e.g., clicking on a photo in a website together with holding down the Control key results in options 'open it in a new window,' 'save it,' or 'copy it' )
      - Helps overcome some of the navigation problems associated with cascading menus

    - Collapsable menus
      - More options can be visible when selecting



- **Research & design issues**
    - What are best names/labels/phrases to use?
    - Placement in list is critical (Quit and save need to be far apart)
    - Many international guidelines exist emphasizing depth/breadth, structure and navigation
    - Menus for small displays
        - **Learnability** (no documentation available on how to use it)
        - **Less is more** (functions should be limited to essential ones)
        - **Sequence** by **frequency of use**
        - Allow users to deal with **interruptions** (e.g. getting calls on your phones)
        - **Simplicity**

Icons:

- Icons are assumed to be **easier to learn and remember** than commands
- Can be designed to be compact and variably positioned on a screen
- Now pervasive in every interface, e.g.:
    - represent desktop objects
    - tools (e.g. paintbrush)
    - applications (e.g. web browser)
    - operations (e.g. cut, paste, next, accept, change)
- Since the Xerox Star days icons have changed in their look and feel:
    - black and white -> color
    - shadowing
    - photorealistic images
    - 3D rendering
    - animation
- Many designed to be very **detailed** and **animated** making them both visually attractive and informative
- GUIs now highly inviting, emotionally appealing, and feel alive
- The mapping between the representation and underlying referent can be:
    - Metaphor
    - Direct mapping
    - Convention
- Most effective icons: **similar or direct mapping** ones
- Many operations are actions making it more difficult to represent them
    - use a combination of objects and symbols that capture the salient part of an action
- **Research & design issues**
    - There is a wealth of resources now so do not have to draw or invent new icons from scratch (guidelines, style guides, icon builders, libraries)
    - Text labels can be used alongside icons to help identification for small icon sets
    - For large icon sets (e.g. photo editing or word processing) use rollovers
    - Legibility: can user discriminate between icons?
    - Interpretation: can user understand what the icon is meant to convey?
    - **Icon should be**
        - Understandable
        - Familiar
        - Unambiguous
        - Memorable
        - Informative
        - Few
        - Distinct
        - Attractive
        - Legible
        - Compact
        - Coherent
        - Extensible

Multimedia:

- Combines different media within a single interface with various forms of interactivity
  - Graphics
  - Video
  - Animations
  - Text
  - Sound
- Users click on links in an image or text
  - another part of the program
  - an animation or a video clip is played
  - can return to where they were or move on to another place
- + Facilitates rapid access to multiple representations of information
  + Can provide better ways of presenting information than can any media alone
  + Can enable easier learning, better understanding, more engagement, and more pleasure
  + Can encourage users to explore different parts of a game or story
  + Reduce load on visual system
- - Tendency to play video clips and animations, while skimming through accompanying text or diagrams
- **Research & design issues**
  - How to design multimedia to help users explore, keep track of, and integrate the multiple representations
    - provide hands-on interactivities and simulations that the user has to complete to solve a task
    - Use '**dynalinking**,' where information depicted in one window explicitly changes in relation to what happens in another (Scaife and Rogers, 1996)
  - Several guidelines that recommend how to combine multiple media for different kinds of tasks

**Information visualization**:

- Computer-generated interactive graphics of complex data
- Amplify human cognition, enabling users to see patterns, trends, and anomalies in the visualization (Card et al, 1999)
- Aim is to enhance discovery, decision-making, and explanation of phenomena
- Techniques include:
  - 3D interactive maps that can be zoomed in and out of and which present data via webs, trees, clusters, scatterplot diagrams, and interconnected nodes
  - Dashboards
- **Research and design issues**
  - Whether to use animation and/or interactivity
  - What form of coding to use, e.g. color or text labels
  - Whether to use a 2D or 3D representational format
  - What forms of navigation, e.g. zooming or panning,
  - What kinds and how much additional information to provide, e.g. rollovers or tables of text
  - What navigational metaphor to use
  - Importing and cleaning data
  - Finding related information
  - Viewing large volumes of data
  - Integrating data mining
  - Integrating with analytical reasoning techniques
  - Collaborating with others

- **Principles of information-seeking**: overview first, zoom and filter, then details on demand
- **Dashboard** = a screenshot of data, updated over periods of time
  - Give a clear understanding of things to help establish a feeling of certainty
  - Give the resources to predict and plan for the future
  - Help to complete critical tasks in time to avoid last minute panic

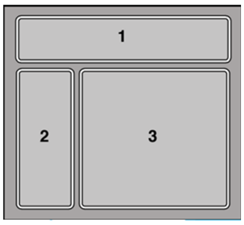
## Web

- Early websites were largely text-based, providing hyperlinks
- Concern was with how best to structure information at the interface to enable users to navigate and access it easily and quickly
- Nowadays, more emphasis on making pages distinctive, striking, and pleasurable
- **Usability ⇔ attractiveness**:
  - you can have both!
  - Vanilla or multi-flavor design? Ease of finding something ⇔ aesthetic and enjoyable experience
  - Web designers are "thinking great literature"
  - Users read the web like a "billboard going by at 60 miles an hour" (Krug, 2000)
  - Need to determine how to brand a web page to catch and keep 'eyeballs'
- **Research & design issues**
  - Need to consider how best to design, present, and structure information and system behavior
  - But also content and navigation are central
  - **Veen's design principles**

    (1) Where am I?

    (2) Where can I go?

    (3) What's here?



→ very recognisable

  - Top 10 mistakes:
    - Burying information too deep in a web site
    - Overloading pages
    - Awkward or confusing navigation
    - Putting information in unexpected places on the page
    - Not making links obvious and clear
    - Presenting information in bad tables
    - Making text so small users can't read it
    - Poor colour combinations for text
    - Using bad forms
    - Hiding (or not providing) features that could help users: Tullis (2005)
- **Web content presentation**:
  - Site-level issues: apparent throughout the entire site
  - Page-level issues: observed at an individual page level
  - "special" types of information: includes site maps, search functions, …
- **Guidelines**:
  - World Wide Web Consortium's Web Guidelines (www.w3.org)
  - Sun's Web Design Guide
  - The Research-Based Web Design & Usability Guidelines (usability.gov)
  - …

## Mobile

- = **handheld devices** intended to be used while on the move
- Have become pervasive, increasingly used in all aspects of everyday and working life
- Applications running on handhelds have greatly Mobile expanded, e.g.:
  - used in restaurants to take orders
  - car rentals to check in car returns
  - supermarkets for checking stock
  - in the streets for multi user gaming
  - in education to support life-long learning
- advent of the **iPhone app** = a whole new user experience that was designed primarily for people to enjoy
  - many apps not designed for any need, want or use but purely for idle moments to have some fun → no longer work-oriented & no longer just a phone (fun-oriented)
  - e.g.: iBeer developed by magician Steve Sheraton
  - ingenious use of the accelerometer that is inside the phone
- **QR-codes**:
  - menu's in restaurants are no longer printed → cost-saving
  - can compare products
  - find your missing dog (QR-code in collar)
  - ...
- **Mobile challenges**:
  - Small screens, small number of keys and restricted number of controls
  - Navigation difficulty
  - Lack of consistency in interfaces
  - Many smartphones now use multi-touch surface displays
  - Innovative physical designs including:
    - roller wheels
    - rocker dials
    - softkeys
    - up/down 'lips' on the face of phones
    - 2-way and 4 way directional keypads
    - silk-screened buttons
  - Usability and preference varies: depends on the dexterity and commitment of the user
- Special needs:
  - Braille phone
  - Phone for old people
- **Research & design issues**
  - Mobile interfaces can be tricky and cumbersome to use for those with poor manual dexterity or 'fat' fingers
  - Key concern is designing for small screen real estate and limited control space (e.g. mobile browsers allow users to view and navigate the internet, magazines etc., in a more streamlined way compared with PC web browsers)
  - Used by all sorts of people in all manner of physical and social contexts
  - Size of screen makes observation of user interaction difficult

## Speech

- = where a **person talks with a system** that has a spoken language application (e.g., timetable, travel planner)
- Used most for inquiring about very specific information (e.g. flight times or to perform a transaction, e.g. buy a ticket)

- Also used by people with disabilities
  (e.g. speech recognition word processors, page scanners, web readers, home control systems)
- **Advantages**:
    - Faster to speak than write
- **Disadvantages**:
    - Bigger bandwidth needed
    - Requires knowledge of a language (steep learning curve)
    - Frustrating if forced to listen
- Better one? Depends on context & technology available
- **Call routing**
    - = caller-led speech where users state their needs in their own words (e.g. "I'm having problems with my voice mail")
    - Idea is they are automatically forwarded to the appropriate service
    - Sometimes people just want a human operator
    - **Format:**
        - Directed dialogs are where the system is in control of the conversation
        - Ask specific questions and require specific responses
        - More flexible systems allow the user to take the initiative:
            - e.g. "I'd like to go to Format Paris next Monday for two weeks."
        - More chance of error, since caller might assume that the system is like a human
        - Guided prompts can help callers back on track
            - e.g. "Sorry I did not get all that. Did you say you wanted to fly next Monday?"
        - Automatic speech recognition
            - Allows users to speak continuously into a microphone
            - Speech converted into text and displayed in real time
            - Require orientation for user to establish a profile
            - Typical applications:
                - Command and control
                - Dictation
                - simple speech APIs
                - transcription
                - keyword spotting
- **Research & design issues**
    - How to design systems that can keep conversation on track
        - help people navigate efficiently through a menu system
        - enable them to easily recover from errors
        - guide those who are vague or ambiguous in their requests for information or services
    - Type of voice actor (e.g. male, female, neutral, or dialect) → do people prefer to listen to and are more patient with a female or male voice, a northern or southern accent?
    - Vocabulary size increase leads to
        - Increased cost
        - Increased confusability
    - How to recognise words
    - Inherent ambiguities: I scream vs Ice cream

## Touch

- **Touch screens** = detect the presence and location of a person's touch on the display (e.g. walk up kiosks)
- Multi-touch support a range of more dynamic fingertip actions
  (e.g. swiping, flicking, pinching, pushing and tapping)
- Now used for many kinds of displays, such as Smartphones, tablets and tabletops
- **Research & design issues**
  - More fluid and direct styles of interaction involving freehand and pen-based gestures
  - Core design concerns include whether size, orientation, and shape of touch displays effect collaboration
  - Sequential targets are easier to hit
  - Much faster to scroll through wheels, carousels and bars of thumbnail images or lists of options by finger flicking
  - More cumbersome, error prone and slower to type using a virtual keyboard on a touch display than using a physical keyboard
  - Finger-flicking, stroking and touching a screen has resulted in new ways of consuming, reading, creating and searching digital content

## Virtual reality

- = **computer-generated graphical simulations** providing: "the illusion of participation in a synthetic environment rather than external observation of such an environment" (Gigante, 1993)
- provide new kinds of experience, enabling users to interact with objects and navigate in 3D space
- Create highly engaging user experiences
- **Pros**:
  - Can have a higher level of fidelity with objects they represent compared to multimedia
  - Induces a sense of presence where someone is totally engrossed by the experience : "a state of consciousness, the (psychological) sense of being in the virtual environment" (Slater and Wilbur, 1999)
  - Provides different viewpoints: 1st and 3rd person
- **Cons**:
  - Head-mounted displays are uncomfortable to wear, and can cause motion sickness and disorientation
  - Expensive equipment
  - Requires specialised knowledge to operate
- **Types**:
  - **Non-immersive** = screen based, pointer driven, 3D graphical representations
  - **Immersive** = create a sense of being in a world populated by virtual objects
- **Research & design issues**
  - Much research on how to design safe and realistic VRs to facilitate training
    - e.g. flying simulators
    - help people overcome phobias (e.g. spiders, talking in public)
  - Design issues
    - how best to navigate through them (e.g. first versus third person)
    - how to control interactions and movements (e.g. use of head & body movements)
    - how best to interact with information (e.g. use of keypads, pointing, joystick buttons)
    - level of realism to aim for to engender a sense of presence

## Augmented reality

- **augmented ⇔ mixed reality**:
    - **Augmented reality** - virtual representations are superimposed on physical devices and objects
    - **Mixed reality** - views of the real world are combined with views of a virtual environment
    - Many applications including medicine, games, flying, and everyday exploring
        - Medicine:
            - virtual objects (e.g. X-rays and scans) are overlaid on part of a patient's body
            - aid the physician's understanding of what is being examined or operated
        - Air traffic control:
            - dynamic information about aircraft overlaid on a video screen showing the real planes, etc. landing, taking off, and taxiing
            - Helps identify planes difficult to make out
- **'smart' augmented reality**: smartphone apps intended to guide people walking in a city
    - arrows and local information (e.g. nearest McDonalds) are overlaid on a picture of the street the person is walking in 'Smart' augmented reality?
    - Will this mean people spending most of their time glued to their smartphone rather than looking at the sights?
- **Research & design issues**
    - What kind of digital augmentation?
        - When and where in physical environent?
        - Needs to stand out but not distract from ongoing
        - Need to be able to align with real world objects
    - What kind of device/interface? Smartphone, head up display or other?
        - Will depend on task, users, context, cost, robustness, etc.
        - Mobile platforms taking over from PCs
        - Speech interfaces also being used much more for a variety of commercial services
        - Appliance and vehicle interfaces becoming more important

## Shareable

- Can support **more equitable participation** compared with groups using single PC
- Shareable interfaces are designed for **more than one person to use**
    - Provide multiple inputs & sometimes allow simultaneous input by co-located groups
    - simultaneously view the interactions and have same shared point of reference as others
- **Types of shareables**
    - Large wall displays: where people use their own pens or gestures
    - Interactive tabletops: where small groups interact with information using their fingertips υ (e.g. DiamondTouch, Smart Table and Surface)
    - Tangibles: where user interact with physically embedded artifacts
- **Advantages**:
    - Provide a large interactional space that can support flexible group working
    - Can be used by multiple users
        - can point to and touch information being displayed
        - Shared viewpoint

- **Research & design issues**
    - More fluid and direct styles of interaction involving freehand and pen-based gestures
    - Core design concerns include whether size, orientation, and shape of the display have an effect on collaboration
    - horizontal surfaces compared with vertical ones support more turn-taking and collaborative working in co-located groups
    - Providing larger-sized tabletops does not improve group working but encourages more division of labor

**Summary**:

- Many innovative interfaces have emerged post the WIMP/GUI era, including speech, wearable, mobile and tangible
- Many design and research questions need to be considered to decide which to use
- An important concern that underlies the design of any kind of interface is how information is represented to the user so they can carry out on-going activity or task

## 7. Design prototyping & construction

**What** is a **prototype**?

- Other design fields: = **a small-scale model**: e.g.
    - a miniature car
    - a miniature building or town
- Interaction design: e.g.
    - a series of screen sketches
    - a storyboard (= a cartoon-like series of scenes)
    - a PowerPoint slide show
    - a video simulating the use of a system
    - a lump of wood (e.g. Palm Pilot)
    - a cardboard mock-up
    - a piece of software with limited functionality written in the target language or in another language

**Why** prototype?

- Can **fail early** and **inexpensively**
- **Evaluation and feedback** are central to interaction design
- **Stakeholders** can see, hold, interact with a prototype more easily than a document or a drawing
- **Team members** can communicate effectively
- **Technically understand** the problem
- You can **test out ideas** for yourself
- Forces you to **focus on the core functionality**
- It encourages **reflection**: very important aspect of design
- **answer questions** & support designers in **choosing between alternatives**

**What** to prototype?

- Technical issues
- Work flow, task design
- Screen layouts and information display
- Difficult, controversial, critical areas

**How** to prototype? **7 step process**

1. **Identify** the problem (→ understand WHY people want something)
2. **Sketch** out ideas to solve the problem
3. **Share** your sketches with your team
4. **Revise** your sketches and prototype your epicentre
5. **Test** your prototype with potential users
6. **Interpret** your test results
7. **Build a new** prototype and keep testing

**Kinds of prototyping**

- **Low-fidelity prototyping** =
    - Uses a medium which is **unlike the final medium** (e.g. paper, cardboard)
    - Is **quick, cheap and easily changed**
    - Completely disposable
    - **Library**: pre-fabricated menus, buttons, dialogs, ...
    - Examples:
        - **sketches**
            - Don't be inhibited about drawing ability. Practice simple symbols
            - Actions & concepts are more difficult to draw that objects
        - **task sequences**
        - **'Post-it' notes**
        - Storyboards
            - used with **scenarios**, bringing more detail, and a chance to role play
            - = a **series of sketches** showing how a user might progress through a task using the device
            - Used early in design
            - Steps:
                1. Identify the **key tasks** & choose 1
                2. Produce **detailed design** of the task using text / diagrams
                   ⇨ consider alternative options & select those which deal with the task most effectively
                3. **Step through the design** to ensure it covers all issues
                4. **Sketch** the storyboard
                   ⇨ include interaction between people automatic & manual steps + interactions with the system
                5. **Repeat**!
        - Paper prototypes
            - Can use index cards
                - Each card represents one screen or part of screen
                - The user needs a button to go to the next screen (full control)
                - Often used in website development
            - Steps:
                1. **Sketch** main screens on large paper
                2. Create **interactions** using library
                3. **Assemble** components to enable users to perform critical tasks
                4. Be **productive**

- Wireframing = basic rendering of a page layout using (e.g. html)
    - This is more **high-fidelity**
    - to link sections of the screen to other screens
    - allows remote users to test the application and give feedback
    - Wireframe prototypes are recommended when the pages are dynamically generated
    - can be done using software like Adobe XD, Figma, Miro or even Paint
    - built for **review on the computer** ⇔ paper prototypes
    - Allow management and users to **understand how software will work** when something is clicked on in a way that paper prototypes do not
    - Rough out the form of the product Wireframing
        - Content
        - Functionality
        - Means of accessing or navigating to content and functionality
    - Can have annotations for non-obvious items → documents why control is there

- High-fidelity prototyping =
    - Uses **materials** that you would expect to be in the **final product**
    - Prototype looks more like the final system than a low fidelity version
    - For a high-fidelity software prototype common environments include Macromedia Director, Visual Basic, and Smalltalk
    - – danger that users think they have a full system
    - – **less efficient** than low-fidelity (time-consuming)
    - – ineffective for **proof of concept** (if it turns out no one likes it)
    - + useful for **selling ideas** to other people (it looks more exciting)
    - + better for **testing** and finding even more issues

**Compromises in prototyping**

- software-based prototyping: is a slow response? sketchy icons? limited functionality?
- mustn't be ignored: the product needs engineering
- **2 types**:
    - Horizontal = provide a wide range of functions, but with little detail
    - vertical = provide a lot of detail for only a few functions

## Construction

- = taking the prototypes (or learning from them) and **creating a whole**
- **Quality** must be attended to:
    - usability (of course)
    - reliability
    - robustness
    - maintainability
    - integrity
    - portability
    - efficiency
    - …
- Product must be engineered
    - Evolutionary prototyping = final version will evolve from the prototype (can be too forced → disastrous)
    - 'Throw-away' prototyping = not give in to pressure by using a disposable prototype to use it for a final version → throw them away instead

Conceptual design: (from **requirements design**)

- = "a description of the proposed system in terms of a set of integrated ideas and concepts about what it should do, behave and look like, that will be understandable by the users in the manner intended"
- Don't move to a solution too quickly. **Iterate**, iterate, iterate
- Consider **alternatives**: prototyping helps

Interface metaphors:

- = combining **familiar knowledge** with new knowledge in a way that will help the user understand the product
- **Three steps**:
  1. understand functionality
  2. identify potential problem areas
  3. generate metaphors
- **Evaluate metaphors**:
  - How much structure does it provide?
  - How much is relevant to the problem?
  - Is it easy to represent?
  - Will the audience understand it?
  - How extensible is it?

**Interaction types**:

- **Which** interaction type?
  - How the user invokes actions
  - Instructing, conversing, manipulating or exploring
- Most models include a **combination** of interface types
- Important to **avoid being unduly influenced** by **pre determined interface type**
- Do different interface types **provide insight**? (WIMP, shareable, augmented reality, ...)

**Expanding the conceptual model**

- What **functions** will the product perform?
  - What will the product do and what will the human do (task allocation)?
- How are the **functions related to each other**?
  - Sequential or parallel?
  - Categorisations (e.g. all actions related to telephone memory storage)
- What **information** needs to be available?
  - What data is required to perform the task?
  - How is this data to be transformed by the system?

Physical design:

- Considers more **concrete, detailed issues** of designing the interface
- Iteration between **physical and conceptual design** (e.g. a lot of information on a small screen)
- **Guidelines** for physical design
  - **Nielsen's heuristics !**
    - Visibility of system status
    - Error prevention
    - User control and freedom
    - Help and documentation
    - Flexibility and efficiency of use
    - Consistency and standards
    - Recognition rather than recall
    - Match between system and real world
    - Aesthetic and minimalist design
    - Help user recognise, diagnose & recover from errors

- **Shneiderman's eight golden rules !**
    - Strive for consistency
    - Offer informative feedback
    - Permit easy reversal of actions
    - Support internal locus of control
    - Enable frequent users to use shortcuts
    - Offer error prevention & simple error handling
    - Design dialogues to yield closure
    - Reduce short-term memory load
- **Styles guides**: commercial, corporate, decide 'look and feel' for your widgets prescribed (e.g. icons, toolbar)

**Using scenarios in conceptual design**

- Express proposed or imagined situations
- Used throughout design in various ways
    - scripts for user evaluation of prototypes u concrete examples of tasks
    - as a means of co-operation across professional boundaries
- Plus and minus scenarios to explore extreme cases

1. **Creating scenarios**:
    A. Set the scene
    B. Establish the goal or conflict
    C. Overcome crises along the way
    D. Achieve resolution
    E. Reach conclusion
2. Generate **storyboard** from scenario
3. Generate **card-based prototype**

**Summary**:

- Different kinds of prototyping are used for different purposes and at different stages
- Construction: the final product must be engineered appropriately
- Conceptual design (the first step of design)
- Consider interaction types and interface types to prompt creativity
- Storyboards can be generated from scenarios
- Card-based prototypes can be generated from storyboards
- Can use tools and methodologies to help in the design process

# 8. Introduction Evaluation

**Iterative design & evaluation** → a continuous process that examines:

- **Why**: to check that users can use the product and that they like it.
- **What**: a conceptual model, early prototypes of a new system and later, more complete prototypes.
- **Where**: in natural and laboratory settings.
- **When**: throughout design; finished products can be evaluated to collect information to inform new products.
- Designers need to check that they understand users' requirements.

Bruce Tognazzini: Why need to evaluate?

➔ "Iterative design, with its repeating cycle of design and testing, is the only validated methodology in existence that will consistently produce **successful results**. If you don't have **user-testing** as an integral part of your design process you are going to throw buckets of **money down the drain**."

**Types of evaluation**:

- **Formative** = evaluation before design is finalised in order to make improvements in the design
- **Summative** = evaluation of the completed design

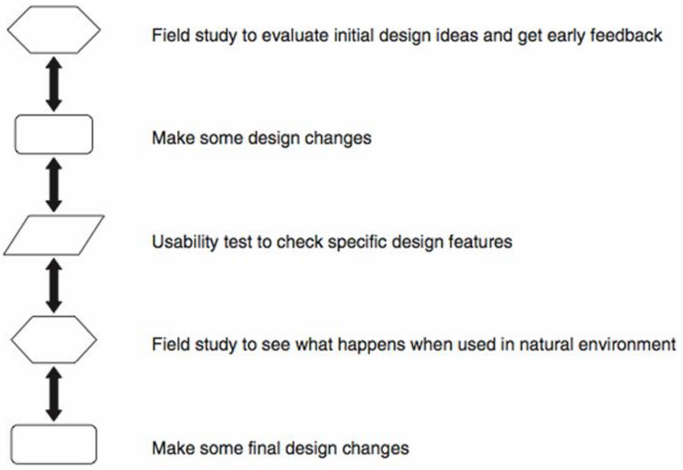
- **<u>Controlled settings</u>** <u>involving</u> users (e.g. usability testing & experiments in <u>laboratories</u> and living labs) → is design A faster to complete a task than design B?
    - **Usability lab**
        - Physical lab
        - Replicate the natural environment in the lab but with more control over it
        - <u>E.g.</u> build a plane in a lab when people use their product mostly on planes
    - **Living labs**
        - People's **use of technology** in their **everyday lives** can be evaluated in living labs.
        - Such evaluations are too difficult to do in a usability lab.
        - <u>E.g.</u> the Aware Home was embedded with a complex network of sensors and audio/video recording devices.

    ➔ + **Quantitative data** (easier to analyse)
- **<u>Natural settings</u>** <u>involving</u> users (e.g. <u>field studies</u> to see how the product is used in the real world) → little or no control over the user
    - **Field studies**: evaluate people in their natural surroundings
        - Help **identify opportunities** for new technology
        - Establish **requirements** for a new design
        - **Facilitate introduction** of new technology
        - **Uses** → Interviews, observation and logging

    ➔ - **Expensive & difficult** to conduct

    ➔ - **Qualitative data** (more difficult to analyse
- <u>Any</u> settings **<u>not involving users</u>** (e.g. <u>consultants</u> critique, predict, analyze & model aspects of the interface analytics)

➔ **Usability testing & field studies can complement**


Field study to evaluate initial design ideas and get early feedback
Make some design changes
Usability test to check specific design features
Field study to see what happens when used in natural environment
Make some final design changes

**<u>Evaluation case studies</u>**:

- <u>Experiment to investigate a computer game</u>
    - Challenge & engagement in a collaborative immersive game
    - Physiological measures were used.
    - Each person played once against a computer and once against another person
    - Players were more engaged when playing against another person than when playing against a computer.
    - What were the **precautionary measures** that the evaluators had to take?
        - Mitigate against extra stress due to sensors & wires → e.g. let them play without it first
        - They might be playing better the second time → e.g. half play against computer first, other half against people first (then effect will cancel each other out)
    - <u>Results</u>: playing against a friend is more challenging, less easy, more engaging, less frustrating, more fun, ...
- <u>In the wild field study of skiers</u>
    - In-situ user studies sampling experiences
    - Understanding **ordinary living** and designing technology that extends this
    - How long?
        - Depends
        - **Difficult to isolate specific effects** → what actually cause the issue?
    - **Control** moves to the **participant**

- Gathering **ethnographic data** at the royal highland show
    - Studies how users interact with technology in natural environment
    - Usually involves reflection
        - User interviews
        - Diaries
    - Used a chatbot to ask questions as participants wandered around the show → directed them to particular parts of the show & then asked questions about it
    - Aim: find out about their experience of & feelings about using the chatbot
    - Types of data:
        - Responses to list of questions (e.g. I learned something)
        - Additional open-ended comments & photo's they could upload
        - Interview data with researchers

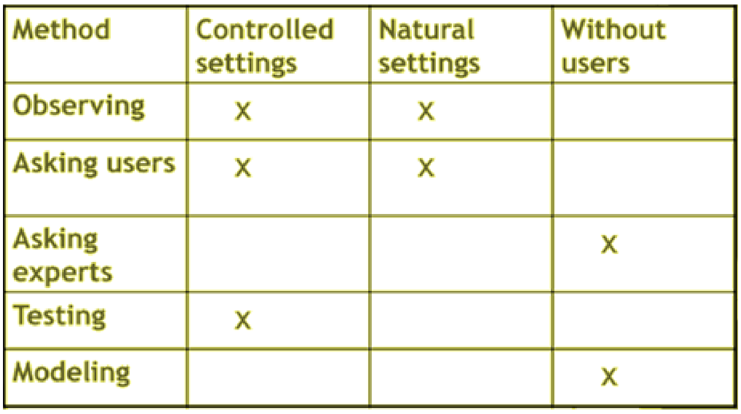
### Crowd sourcing

- = method of gathering people to do the testing with
- + Access to larger groups of individuals
- + faster & cheaper
- + varied user background
- – People do the minimum amount of work to gain the max amount of money
- – Need a way of verifying results (people might give randomized quick answers)
- – less interaction with participants
- – less focused user groups

### Gorilla usability testing

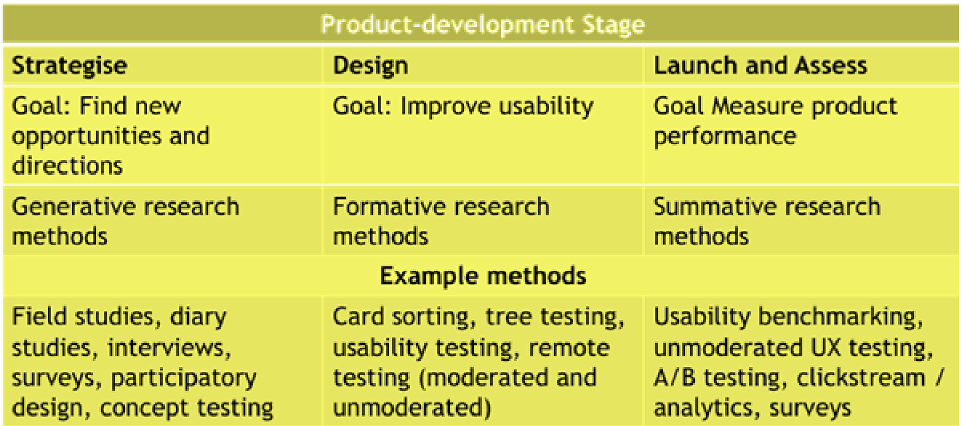
- = "the art of **pouncing on lone people** in cafes and public spaces, [then] quickly filming them whilst they use a website for a couple of minutes."
- Use when have **tight timeline**
- Can use **low fidelit**y techniques (e.g. sketching)
- Questions to get started
    - **What** shall we test?
    - **Where** will we test?
    - **With whom** will we test?
    - **How** will we test?
- Employing the technique
    - Beware the **implicit bias**
    - **Explain** what's going on
    - Be **ethical**
    - Make it **casual**
    - Be **participatory**
    - **Don't lead** participants (the product = tested, not them)
    - Be **observant** (body language, tone of voice)
    - **Capture** feedback (recording, writing, …)
    - Be conscious of **time** (don't waste their time)
- Issues:
    - Informing participants of **their rights** and getting their **consent**
        - Right to know the goal
        - They can leave when they want
    - Factors that influence **how the data is interpreted**
        - Reliability: can the study be replicated
        - Validity: is it measuring what you expected?
        - Biases: is the process creating biases
        - Scope: can the findings be generalized
        - Ecological validity: is the environment influencing the findings? i.e. Hawthorn effect
    - **Hawthorn effect** = we behave differently when we know we are being evaluated & measured

**Evaluation methods**:



| Method | Controlled settings | Natural settings | Without users |
|---|---|---|---|
| Observing | x | x | |
| Asking users | x | x | |
| Asking experts | | | x |
| Testing | x | | |
| Modeling | | | x |

**When to use which method**?



| Product-development Stage | | |
|---|---|---|
| **Strategise** | **Design** | **Launch and Assess** |
| Goal: Find new opportunities and directions | Goal: Improve usability | Goal Measure product performance |
| Generative research methods | Formative research methods | Summative research methods |
| **Example methods** | | |
| Field studies, diary studies, interviews, surveys, participatory design, concept testing | Card sorting, tree testing, usability testing, remote testing (moderated and unmoderated) | Usability benchmarking, unmoderated UX testing, A/B testing, clickstream / analytics, surveys |

**When is it time to stop testing**?

- **Every test** will reveal some **area for improvement**
- **Schedule and budget** constraints
- Tests should take as **little time as possible** while still **yielding useful** information

**Key points**:

- Evaluation & design are closely integrated in user-centered design.
- Some of the same techniques are used in evaluation as for establishing requirements but they are used differently (e.g. observation interviews & questionnaires).
- Three types of evaluation: laboratory based with users, in the field with users, studies that do not involve users
- The main methods are: observing, asking users, asking experts, user testing, inspection, and modeling users' task performance, analytics.
- Dealing with constraints is an important skill for evaluators to develop.

# 9. Evaluation Studies: controlled ⇔ natural settings

## Usability testing

- = recording **performance** of **typical users** doing **typical tasks**
- **Controlled** settings
- Users are **observed and timed**
- Data is **recorded** on video & **key presses** are logged
- The data is used to **calculate performance** times, and to **identify & explain errors**
- User satisfaction is evaluated using **questionnaires & interviews**
- **Field observations** may be used to provide **contextual understanding**

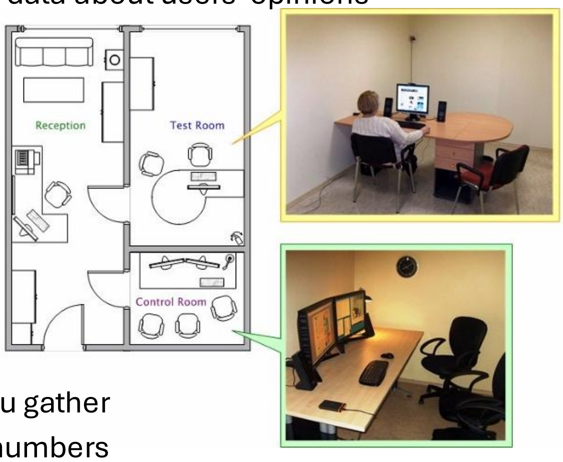
**Experiments ⇔ usability testing**:

- Experiments: test hypotheses to discover new knowledge by investigating the relationship between two or more things (i.e., variables)
- Usability testing = **applied experimentation** → can use experiments as part of usability testing
- **Developers** check that the **system is usable** by the **intended user population** for their tasks
- Experiments may also be done in usability testing

| **usability testing** | ⇔ **experiments for research** |
|---|---|
| Improve products | Discover knowledge |
| Few participants | Many participants |
| Results inform design | Results validated statistically |
| Usually not completely replicable | Must be replicable |
| Conditions controlled as much as possible | Strongly controlled conditions |
| Procedure planned | Experimental design |
| Results reported to developers | Scientific report to scientific community |

**Usability testing**

- Goals & questions focus on how well users perform tasks with the product.
- Comparison of products or prototypes common.
- Focus is on time to complete task & number & type of errors.
- Data collected by video & interaction logging.
- Testing is central.
- User satisfaction questionnaires & interviews provide data about users' opinions
- Usability lab →
  - **Minimum equipment** required
  - **Mobile equipment** for field use (e.g. mobile eye tracker)



- Remote usability testing
  - **Moderated**: moderater present
    - + more flexible in terms of time
    - - not same level of communication
  - **Unmoderated**: analytics based on the data you gather
    - + can be synchronous → much higher numbers
    - - can't see or talk to participants
    - - scenarios need to be straight forward

**Testing conditions**:

- Usability lab or other controlled space.
- Emphasis on:
  - selecting representative users
  - developing representative tasks
- Testing conditions
- 5-10 users typically selected.
- Tasks usually last no more than 30 minutes.
- The test conditions should be the same for every participant.
- Informed consent form explains procedures and deals with ethical issues.

When to conduct usability tests with **high numbers**?

- When testing **large, complex systems**
- When testing systems or features that **require strong confirmation** of high usability
- When **management** need to be **convinced** by numbers

**Data types**:

- Time to complete a task.
- Time to complete a task after a specified time away from the product.
- Number and type of errors per task.
- Number of errors per unit of time.
- Number of navigations to online help or manuals.
- Number of users making a particular error.
- Number of users completing task successfully

**Usability engineering orientation**

- Aim: improvement with each version.
- Current level of performance.
- Minimum acceptable level of performance.
- Target level of performance

**How many participants = enough**?

- The number is a practical issue.
- **Depends** on:
  - schedule for testing
  - availability of participants
  - cost of running tests
- Typically 5-10 participants
- Some experts argue that testing should continue **until no new insights are gained**

**Field studies**:

- Field studies are done in natural settings.
- in the wild" is a term for prototypes being used freely in natural settings
- Aim to understand what users do naturally and how technology impacts them.
- Field studies are used in product design to:
  - identify opportunities for new technology
  - determine design requirements
  - decide how best to introduce new technology
  - evaluate technology in use

**Data & collection analysis**:

- **Observation & interviews**
  - Notes, pictures, recordings
  - Video
  - Logging
- **Analysis**:
  - Categorized
  - Categories can be provided by theory
    - **Grounded theory** = theory is grounded in the data
    - **Activity theory** = practical theories in the world

**Data presentation**:

- Aim: show how the products are being appropriated and integrated into their surroundings
- Typical **presentation forms** include: vignettes, excerpts, critical incidents, patterns, and narratives

**Eye tracking experiment**: Combined quantitative and qualitative methods

- **Eye tracking**
  - Tracks where the individual is looking on screen
  - Fixation = when eye is focused on area of interest
  - Saccade = movement from one area of interest to another
- **Cued retrospective think aloud**
  - used to understand participants' thought process
  - used animation of eye tracking
- **heat maps**: showed interest areas on which attention was focussed

**Key points**:

- Usability testing is done in controlled conditions.
- Usability testing is an adapted form of experimentation.
- Experiments aim: test hypotheses by manipulating certain variables while keeping others constant.
- Field studies are done in natural environments.
- "In the wild" is a recent term for studies in which a prototype is freely used in a natural setting.
- Typically observation and interviews are used to collect field studies data.
- Data is usually presented as anecdotes, excerpts, critical incidents, patterns and narratives.

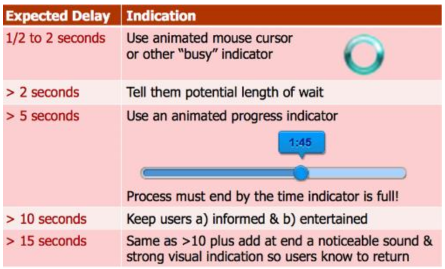
# 10. Analytical evaluation

## Inspections

- Several kinds
- Experts use their **knowledge of users & technology** to **review software usability**.
- **Expert critiques** (crits) can be formal or informal reports.
- **Heuristic evaluation** = a review guided by a set of heuristics.
- **Walkthroughs** = stepping through **a pre-planned scenario** noting potential problems.

## Heuristic evaluation

- Developed by **Jacob Nielsen** in the early **1990s**
- Based on heuristics distilled from an **empirical analysis** of 249 usability problems.
- These heuristics have been **revised for current technology**.
- Heuristics developed for mobile devices, wearables, virtual worlds, …
- **Design guidelines** form a basis for developing heuristics.
- **Nielsen's original heuristics**
  - **Visibility of system status** →
    System should always keep users informed, with appropriate feedback at reasonable times (e.g. 30% loaded)



| Expected Delay | Indication |
|---|---|
| 1/2 to 2 seconds | Use animated mouse cursor or other "busy" indicator |
| > 2 seconds | Tell them potential length of wait |
| > 5 seconds | Use an animated progress indicator<br>1:45<br>Process must end by the time indicator is full! |
| > 10 seconds | Keep users a) informed & b) entertained |
| > 15 seconds | Same as >10 plus add at end a noticeable sound & strong visual indication so users know to return |

  - **Match system & real world** →
    The system should speak the user's language, with familiar words. Information should appear in natural and logical order.

  - **User control and freedom**
    Users often choose functions by mistake and need a clearly marked "emergency exit."
    - Support undo and redo.
    - Select 'ok' when finished doing something (like reading the text / inserting something)

- **Consistency and standards**

  Users should not have to wonder whether different words/actions mean the same thing. Follow platform conventions.
  - Email & password for logins
  - Word, powerpoint & excell have same layout
  - Arrow up = higher, arrow down = lower
- **Error prevention**

  Better than good error messages is a careful design that prevents the problem in the 1st place
  - At least 8 characters + symbols in passwords
  - Can't open something because not supported → give message to prevent & explain
  - Secondary (don't save) & primary actions (save)
- **Recognition rather than recall**

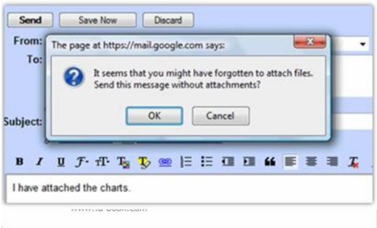
  Minimize the user's memory load by making/actions, options visible. The user shouldn't have to remember from one dialog to next.
  - Recently viewed items & featured recommendations
  - Visible which font you're currently using (e.g. when you want to change font)
- **Flexibility and efficiency of use**

  Accelerators (unseen by novice users) often speed up interaction for expert users. Allow users to tailor frequent actions
  - E.g. ctrl X, ctrl V, ...
- **Aesthetic and minimalist design**

  Interfaces shouldn't contain irrelevant information. Every unit of info competes for attention & diminishes relative visibility.
- **Help users recognize, diagnose, recover from errors**

  Error messages in plain language, precisely indicate problem, suggest solution.
- **Help and documentation**

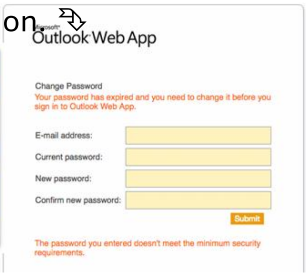
  Best to not need documentation but when necessary, should be easy to search, focused on user tasks, and list concrete steps.



The page at https://mail.google.com says:

It seems that you might have forgotten to attach files. Send this message without attachments?

I have attached the charts.



Change Password

Your password has expired and you need to change it before you sign in to Outlook Web App.

E-mail address:

Current password:

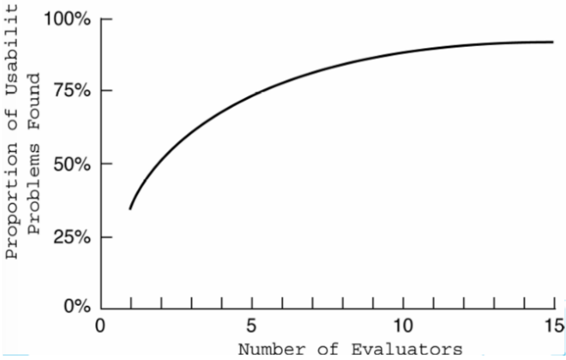
New password:

Confirm new password:

The password you entered doesn't meet the minimum security requirements.

## Discount evaluation

- = heuristic evaluation where **5 evaluators** are used
- Empirical evidence suggests that on average 5 evaluators identify 75-80% of usability problems.



**Stages** of heuristic evaluation:

1. **Briefing session** to tell experts what to do.
2. **Evaluation** period of 1-2 hours in which:
   - Each expert works separately
   - Take one pass to get a feel for the product
   - Take a second pass to focus on specific Stages of heuristic evaluation features.
3. **Severity rating** decided for each problem
   - **0** - don't agree that this is a usability problem Severity Ratings
   - **1** - cosmetic problem
   - **2** - minor usability problem
   - **3** - major usability problem; important to fix
   - **4** - usability catastrophe; imperative to fix

4. **Aggregation** and **prioritisation** of **problems** as a group
5. **Debriefing session** where **outcome** is discussed with the design team.

**Advantages**:

- Few ethical & practical issues to consider because users not involved
- Fast
- Identify the most obvious issues
- Very effective at early stages
- Best for optimising workflow, larger, more obvious parts of design and overall usability
- Best experts have knowledge of application domain & users

**Disadvantages**

- Can be difficult & expensive to find experts
- Important problems may get missed
- Many trivial problems are often identified
- Experts have biases
- No user involvement

**Cognitive walkthrough**

- Focus on **ease of learning**.
- Designer presents an aspect of the **design & usage scenarios**.
- Expert is **told** the **assumptions** about user population, context of use, task details.
- **One or more experts** walk through the design prototype with the scenario.
- Experts are guided by **3 questions**:
    - Will the correct action be sufficiently evident to the user?
    - Will the user notice that the correct action is available?
    - Will the user associate and interpret the response from the action correctly?
- As the experts work through the scenario, they **note problems**.

**Pluralistic walkthrough**

- **Variation** on the cognitive walkthrough theme.
- Performed by a **carefully managed team**.
- The panel of experts begins by **working separately**.
- Then there is **managed discussion** that leads to agreed **decisions**.
- The approach lends itself well to **participatory design**.

**Analytics**

- A method for evaluating **user traffic** through a system or part of a system
- Many examples including Google Analytics, Visistat (shown here)
- Times of day & visitor IP addresses
- Social Action Analysis: **Perer & Shneiderman**, 2008
    - = a system and framework for social network analysis (SNA) that integrates statistical methods with interactive visualizations to support exploratory data analysis
    - designed to balance systematic procedures with flexible exploration ⇨ allowing users to move beyond opportunistic techniques and gain deeper insights from complex network data
    - A key aspect: the **integration of statistical and visualization tools**, including interactive elements like filtering and ranking, to make complex data more comprehensible.

# 11. Cognitive aspects

**Why** need to **understand users**?

- Interacting with technology is cognitive
- Need to take into account cognitive processes involved and **cognitive limitations** of users
- Provides **knowledge** about what users can and cannot be expected to do
- Identifies and explains the **nature and causes** of problems users encounter
- Supply theories, modelling tools, guidance and methods that can lead to the design of better interactive products

**Cognitive processes**

- **Attention**
- **Perception and recognition** ➔ The 3 most important to user experience design
- **Memory**
- Learning
- Reading, speaking and listening
- Problem-solving, planning, reasoning and decision-making

Attention:

- = **selecting** things to **concentrate** on at a point in time from the mass of **stimuli** around us
- Allows us to **focus on information** that is relevant to what we are doing
- Involves **audio** and/or **visual** senses
- **Selective attention** = selects **different parts** of the environment in turn
- **Focused and divided attention** = enables us to be selective in terms of **the mass of competing stimuli** but limits our ability to keep track of all events
- Information at the interface should be **structured** to **capture users' attention** (e.g. use perceptual boundaries (windows), colour, reverse video, sound and flashing lights)

  **Don't use something that moves continuously**: it can blink to get the attention, but stop right after because it is distracting otherwise
- **2 main types**:
  - **Controlled processing** = **reflective cognition**
    - Use for **difficult tasks**
    - e.g. trying to make decisions → doesn't happen automatically
    - **Heavy demands** on attention
    - Creates new **ideas & creativity**
  - **Automatic processing** = **experimental cognition**
    - Use for **easy tasks**
    - e.g. reading → something we learned to do without thinking too much
    - Little or **no demand** on attention
    - As you **practice** it becomes more automatic

  ➔ Both are essential for everyday life
- **Factors affecting** attention
  - **Stress**
  - **Vigilance**
  - **Mental workload**
  - **Visual search** = our ability to find items in a visual scene
    - Logic of where something could be
    - brighter / bigger objects

- **Design implications** for attention
  - Make **information salient** when it needs attending to
  - Use techniques that make things **stand out** like colour, ordering, spacing, underlining, sequencing and animation
  - **Avoid cluttering** the interface with too much information
  - Avoid using too much because the software allows it

## Perception

- How we **extract meaning** (and hence recognition and understanding) from the world
- Obvious implication is to design representations that are **readily perceivable**, e.g.
  - Text should be **legible**
  - Icons should be easy to **distinguish** and read
  - E.g. text & speech should be **aligned** (⇔ a series where the subtitles are lagging behind)
- We **see what we expect to see**: perception is influenced by our goals
- We don't see irrelevant items
- We fill in the gaps based on our understanding and comprehension
- **Design implications**
  - **Icons** should enable users to readily distinguish their meaning
  - **Bordering and spacing** are effective visual ways of **grouping** information
  - Sounds should be **audible and distinguishable**
  - **Speech output** should enable users to distinguish between the set of spoken **words**
  - **Text** should be legible and distinguishable from the **background**
  - **Tactile feedback** should allow users to recognize and distinguish **different meanings**

## Memory

- Involves **1) encoding** and **2) retrieving knowledge**
- **Processing** in memory: we don't remember everything - involves **filtering and processing** what is attended to
  1. **Encoding** = first stage of memory
     - determines which information is attended to in the environment and how it is interpreted
  2. Processing in memory
     - The **more attention** paid to something...
     - The more it is **processed** in terms of thinking about it and comparing it with other knowledge...
     - The more likely it is to be **remembered**
- **Context** is important in affecting our memory (i.e. where, when)
  - Context affects the **extent** to which information can be subsequently **retrieved**
  - Sometimes it can be difficult for people to **recall information** that was **encoded in a different context**: "You meet your dentist in the local bookshop. It takes a couple of seconds to recognise her as you are used to seeing her in a different context"
- We **recognize** things **much better** ⇔ being able to **recall things**
  - Command-based interfaces require users to recall from memory a name from a **possible set of 100s**
  - GUIs provide **visually-based options** that users need only browse through until they recognize one
  - Web browsers, MP3 players, etc., provide lists of visited URLs, song titles etc., that **support recognition memory**

- **Personal information management** = growing problem for many users
    - vast numbers of documents, images, music files, video clips, emails, attachments, bookmarks,...
        - → **where and how to save them** all, then remembering what they were called and where to find them again
        - **naming** most common means of encoding them
        - but can be difficult to remember, especially when have 1000s and 1000s
    - How might such a process be facilitated taking into account people's memory abilities?
    - Memory involves **2 processes**
        - **recall-directed** = use clues that you can recall to get as close as possible to what we are looking for
        - **recognition-based scanning** = once recall failed we start looking through everything
    - File management systems should be designed to optimize both kinds of memory processes
        - e.g. Search box and history list
    - Help users encode files in richer ways
        - Provide them with ways of saving files using colour, flagging, image, flexible text, time stamping, ...
- **Design implications**
    - **Don't overload users' memories** with complicated procedures for carrying out tasks
    - Design interfaces that **promote recognition** rather than recall
    - Provide **meaningful defaults and intuitive shortcuts**
    - Provide users with **various ways of encoding information** to help them remember (e.g. categories, color, flagging, time stamping)
    - Present information in an **organised, structured, familiar and meaningful** way
    - Give the user **control over the pace** of information presentation
    - Close **proximity** for
        - Items to be compared
        - Required information for a task
    - **Location** of important items
    - **Distinctiveness**

## Learning

- How to learn to use a computer-based application
- Using a computer-based application to **understand a given topic**
- People find it hard to learn by following instructions in a manual: **prefer to learn by doing**
- **Design implications**
    - Design interfaces that **encourage exploration**
    - Design interfaces that **constrain and guide learners**
    - **Dynamically linking** concepts and representations can facilitate the learning of complex material

## Reading, speaking & listening

- The **ease** with which people can read, listen, or speak
    - Many prefer **listening to reading**
        - Reading: can be **quicker** than speaking or listening
        - Listening: requires **less cognitive effort** than reading or speaking
        - Dyslexics have difficulties understanding and recognizing written words

- **Applications**:
    - Speech-recognition systems allow users to interact with them by using spoken commands (e.g. Google Voice Search app Applications)
    - Speech-output systems use artificially generated speech (e.g. written-text-to-speech systems for the blind)
    - Natural-language systems enable users to type in questions and give text based responses (e.g. Ask search engine)
- **Design implications**
    - Speech-based menus and instructions should be **short**
    - Accentuate the **intonation** of **artificially generated speech** voices: they are harder to understand than human voices
    - Provide **opportunities** for making text large on a screen

## Problem solving, planning, reasoning and decision making

- All involves reflective cognition (e.g. thinking about what to do, what the options are, and the consequences)
- Often involves conscious processes, discussion with others (or oneself), and the use of artifacts Problem solving, planning, reasoning and decision making (e.g. maps, books, pen and paper)
- May involve working through different scenarios and deciding which is best option
- **Design implications**
    - Provide additional information/functions for users who wish to understand more about how to carry out an activity more effectively
    - Use simple computational aids to support rapid decision-making and planning for users on the move

## Mental models

- Users develop an **understanding of a system** through **learning** about and **using** it
- **Knowledge** is sometimes described as a mental model:
    - How to use the system (what to do next)
    - What to do with unfamiliar systems or unexpected situations (how the system works)
    - People make inferences using mental models of how to carry out tasks
- **Craik** (1943): mental models = "internal constructions of some aspect of the external world enabling predictions to be made"
- Involves **unconscious** (cannot manipulate) **and conscious** (deliberate) processes
- **images and analogies** are activated
- **Deep ⇔ shallow models** (e.g. how to drive a car and how it works)
- **Characteristics**:
    - Unscientific
    - Constantly evolving
    - If, then, else rules in exploration
- **2 types**
    - **Functional mental model** = knowing what to do but not why
    - **Structural mental model** = understanding the components and their relationships → allow us to problem solve

- Many people have **erroneous mental models** (**Kempton**, 1996) e.g.:
  - You arrive home to a cold house. Turn heat on fully or to desired temperature. Which heats up the house quicker?
  - General valve theory, where 'more is more' principle is generalised to different settings (e.g. car accelerator, gas cooker, tap, radio volume)
  - Thermostats based on model of on-off switch model
- Ideal: **mental model of user = conceptual model of designer**
- Mental models **(+)**
  - include what a person **thinks is true**, not necessarily what is actually true
  - are **similar in structure** to the thing or concept they represent
  - allow a person to **predict the results** of his actions
  - are **simpler** than the thing or concept they represent
  - only **enough information** to allow accurate predictions.
- Mental models **(-)**
  - Can be **wrong**
  - Can be **used improperly**
  - Require **correct and timely information**
  - Require **assessment**
- **Documenting** mental models, key parts:
  - An **image** (needed if the mental model is of a physical thing)
  - A **script** (needed if the mental model has a process)
  - A set of **related mental models**
  - A controlled **vocabulary**
  - A set of **assumptions**

## Errors

- "Human errors" often are "design errors"
- Oversights
  - Lack of attention
  - Unintentional, unconscious
  - Changes during routine operations
- Errors
  - Exhaustion
  - Conscious
  - Knowledge lack