

# INTERACTION DESIGN

---

## Lecture 3

# Agenda

- What is ID?
- Needs & Requirements
- Interaction Models
- Interaction Styles

WHAT IS ID?

---

# What is Interaction Design?

- **Design** = “plan or scheme conceived in the mind and intended for subsequent execution” (Oxford English Dictionary)
- For execution we have to take into consideration the practical **constraints** like resources, materials, cost, and feasibility
- In interaction design: the product’s use and the target domain are the focus → **the development is determined by users, not by technical concerns**

# ID

- *Interaction Design involves four basic activities:*
  - Identifying **needs** and establishing **requirements**
  - Developing alternative **designs** that meet those requirements
  - **Building** interactive versions of the designs so that they can be communicated and assessed
  - **Evaluating what is being built throughout the process**
- ***Evaluating what has been built is the heart of Interaction Design***



# What is Interaction Design?

- Trade-offs, conflicting requirements - evaluation of alternative solutions
- “To get a good idea, get lots of ideas” (Marc Retting)
- Alternative designs must be evaluated by users – the design must be expressed in a form suitable for users to interact with
- The initial plan must be communicated in order to be discussed – **sketches, diagrams, prototypes**

# Case study – Electronic Calendar

- Design an electronic calendar or diary for yourself. Draw a sketch of the system outlining its functionality and its general look&feel (5 minutes)
- What did you do first?
- Did you have any particular artifacts or experience to base your design upon?
- What process did you go through?



# Four Basic Activities of Interaction Design

## 1. Identifying needs and establishing requirements

- Who are the target users and what kind of support an interactive product could usefully provide

## 2. Developing alternative designs that meet the requirements

- **Conceptual design:** what the product should **do, behave and look like**
- “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”
- **Physical design:** colors, sounds, images to use, menus design, icons design
- **Alternatives** must be provided at each step

## 3. Building interactive versions that can be assessed

- Users will best evaluate a design by interacting with it – NO SOFTWARE VERSION IS REQUIRED (paper-based prototypes are cheap and very effective)

## 4. Evaluating the prototypes

- Determining the usability and acceptability of the design with real users



# Key Characteristics of ID process

- **User focus**
- **Specific usability and user experience goals**
  - should be identified, clearly documented and agreed upon at the beginning of the project – they help designers in **choosing between different alternatives** and **check on progress** as the product is developed
- **Iteration**
  - allows the design to be refined based on **feedback** – designers never get the solution right the first time

# NEEDS & REQUIREMENTS

---

# Design & Requirements

- **Design** - creative activity that aims developing products that help users achieve their goals
- Developing a product starts with gaining some understanding of what is required
  - Where these requirements come from?
  - Who are the users?
  - Do the users know what they want? (innovative product)
  - Can we find the users?

# Who Are the Users?

- **Interaction design goal** – to **optimize** the interactions people have with computer-based systems:
  - **support needs**
  - **match wants** and
  - **extend capabilities**
- Identifying needs and establishing requirements – who are the users?
- User's definitions:
  1. People **who interact directly** with the product to achieve a task
  2. People **who manage direct users** , **people who receive products** from the system, people **who test** the system, people **who make the purchase** decision and people who **use competitive products**
- Three users categories:
  1. **Primary users** – frequent users of the system
  2. **Secondary users** – those who occasionally use the system or use the system through an intermediary
  3. **Tertiary users** – those affected by the introduction of the system or who will influence its purchase

# Stakeholders

- People who all have a stake in the development of a successful product - **stakeholders**
- “people or organizations who will be affected by the system and who **have a direct or indirect influence on the system requirements**”
- The group of stakeholders will be larger than the group of people you’d normally think of as users
  - the development team, managers, recipients of the product’s output, people who lose their jobs because of system’s use

# Case study - Stakeholders

- Stakeholders for the check-out system in a supermarket
  - Check-out operators
  - Supermarket owner
  - Warehouse staff
  - Supermarket suppliers
  - Customers



# Electronic Calendar Case Study - Stakeholders

- Example:
- Calendar system:
  - initial user group – one member – you
- stakeholders : people you make appointments with, people whose birthdays you remember, companies that produce paper-based calendars, companies that produce electronic calendars too.
- You DON'T need to involve ALL stakeholders in your user-centered design approach, but you have to be aware of the impact of your product and to decide who should be involved and to what degree



# Needs

- When we try to identify needs, it's not about simply asking a question: "What do you need?"
- We have to understand:
  - the characteristics and the capabilities of the users
    - Size of hands may affect the size and positioning of input buttons
    - Motor abilities may affect the suitability of some input/output devices
    - Height – designing a physical kiosk
  - What they are trying to achieve
  - How they achieve it currently
  - Whether they would achieve their goals more effectively if they get different support type



# Needs

- If the product is an innovation – difficult to identify the users and the representative tasks
- Example: first microwave oven design – designers had to imagine who might want to use it and what they might want to do with it
- Usually, a good indication for future behavior is current or past behavior
- It is always useful to start by understanding similar behavior that is already established
- Example: introduction of cell phones – the model was the standard phone

# Generating Alternative Designs

- Common human tendency – to stick with something that we know works
- Good alternatives may never be considered
- Alternatives comes from designers creativity
- Innovations come from cross-fertilization of ideas from different applications, the evolution of an existing product (word processors) or copying of other similar product (is it legal?)
- “An expert is someone who gets reminded of just the right prior experience to help him in processing his current experiences” (Shank)

# Case study – Electronic calendar

- What did inspire you for your electronic calendar?



- What was innovative?



# Choosing between Alternatives

- 2 features categories:
  - **Externally visible** and measurable:
    - the system takes 4 seconds to respond
  - **Internal features**:
    - database design decisions determine the fact that the query takes 4 seconds to execute
- In interaction design we focus on the externally visible and measurable behavior, but internal decisions will also affect the external features

# Choosing between Alternatives

- Let the users and stakeholders interact with the product
- By discussing their preferences, experiences and suggestions you will chose the proper design
- The design should be available in a form that can be manipulated by real users

# Prototyping

- Used to overcome potential client misunderstandings
- Test technical feasibility of a suggested design
- Involves a limited version of a product to answer specific questions about the design
- Give a better impression of user experience



# Minimum Viable Product (MVP)

- a development technique in which a new product or website is developed with sufficient features to satisfy early adopters
- it is not about creating minimal products
- version of a new product which allows a team to **collect the maximum amount of validated learning about customers** with the least effort.
- the final, complete set of features is only designed and developed after considering feedback from the product's initial users.
- An MVP has three key characteristics:
  - It has **enough value** that people are willing to **use it or buy it** initially
  - It demonstrates **enough future benefit** to retain early adopters
  - It provides a **feedback loop** to guide future development

# MVP: Dropbox

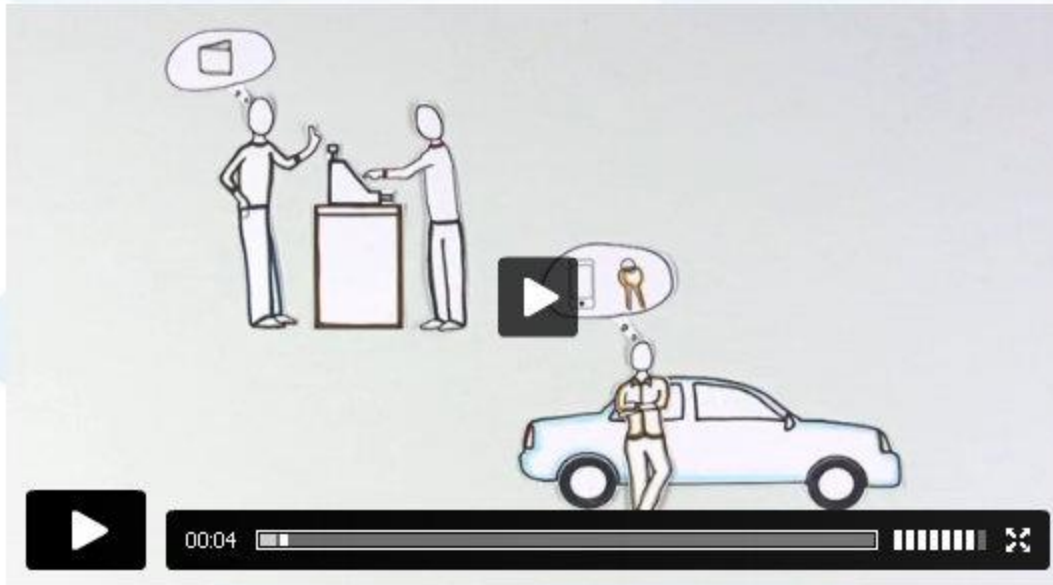
- 2010 - started with an explainer video – a 3m screencast published on Hacker news
- the screencast was enough to give the early adopters a hint of the product experience
- enough to get many smart people to give them “the same feedback as if putting a product in their hand”,
- New version of the video had the waiting list (of emails) jump from 5.000 to 75.000 in one single day.



# MVP Dropbox

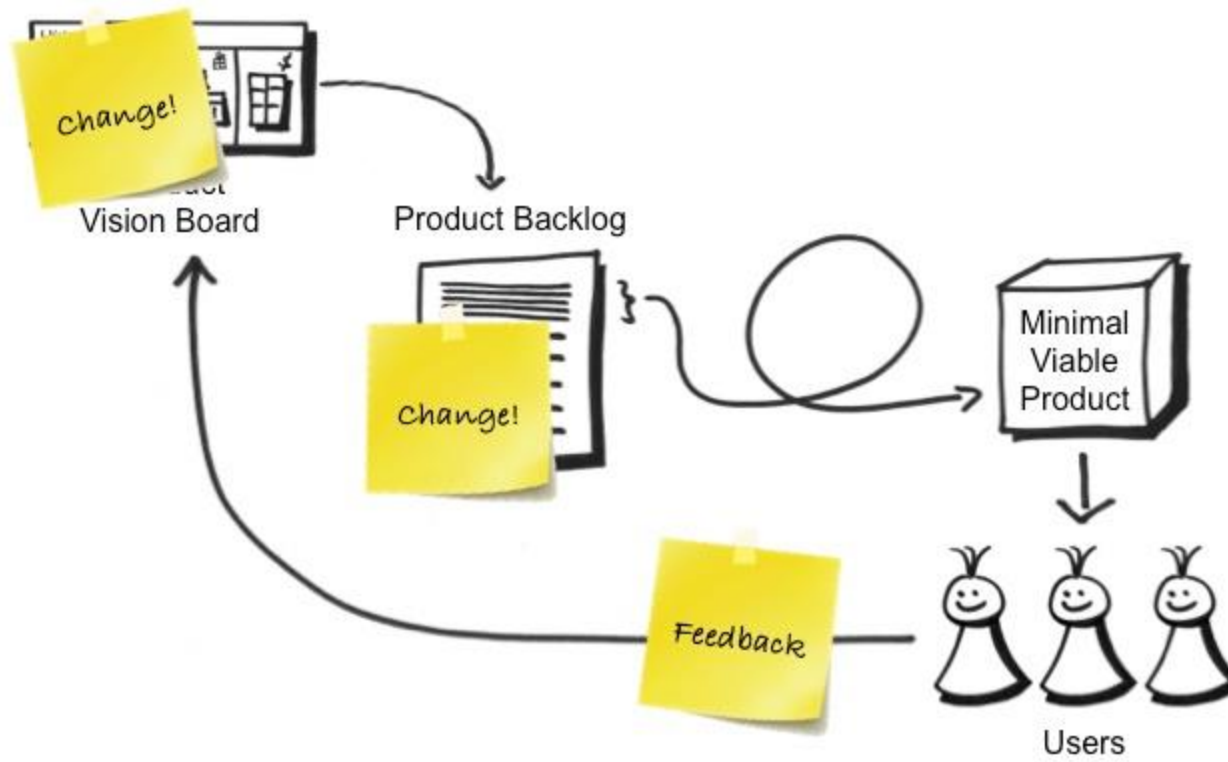


**Dropbox**



<https://www.youtube.com/watch?v=xy9nSnalvPc>

# MVP



# Choosing between Alternatives

- “quality” criterion
- Different understandings of what quality means
- Examples
  - You wait too long for a download – try a different site – you have applied a quality measure associated to the **time** taken for downloading a webpage
  - One cell phone makes it easy to perform a critical function than other that needs several steps – quality measure concerned to **efficiency**



# Choosing between Alternatives

- Different stakeholders – **different understandings** of “quality” and **different acceptance limits**
- Example:
  - All stakeholders agree upon criteria like:
    - “response time will be fast”
    - “the menu structure will be easy to use”
  - What each of these requirements means is likely to vary (“under a second”/“between 2-3 seconds”)
- It is important to **capture these views in an unambiguous language** because:
  - It helps **clarify expectations**
  - Provides a benchmark against the product can be **evaluated**
  - Gives a basis in **choosing** between alternatives



# Usability Engineering



- The process of writing down formal, verifiable and measurable usability criteria=**usability engineering**
- Involves specifying quantifiable measures of a product performance, documenting them in a **usability specification** and assessing the product against them
- This approach is used to make **changes to subsequent versions** of a product based on the results of usability tests

# Electronic Calendar - Case Study

- Suggest some usability criteria that you would use to determine the calendar's quality (quality must be measurable and specific)



# INTERACTION MODELS

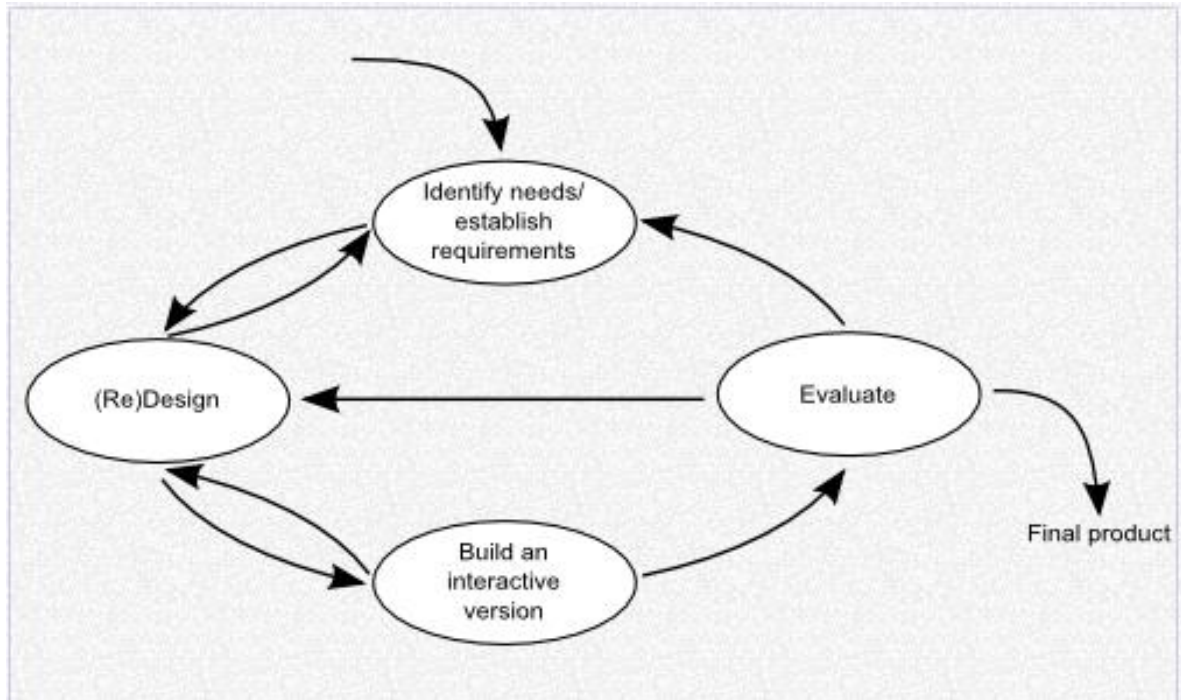
---

# Interaction Design Lifecycle Models

- **Lifecycle model** - a set of activities and how they are related
- May incorporate a description of **when** and **how** to move from one activity to the next and a description of the results for each activity
- Allow developers and managers to get an overall view of the development effort, deliverables specified, resources allocated, targets set
- Simplified version of reality
- Any organization that uses a lifecycle model will need to add specific details
- Example: Microsoft “sync and stabilize” process – wanted to maintain a small-team culture while also making possible the development of large products
  - Small development teams (3-8 members) or individual programmers working as a large team
  - Each team evolves their design
  - Synchronization is done frequently to avoid incompatibilities



# A simple lifecycle model for interaction design (Preece, Rogers, Sharp)

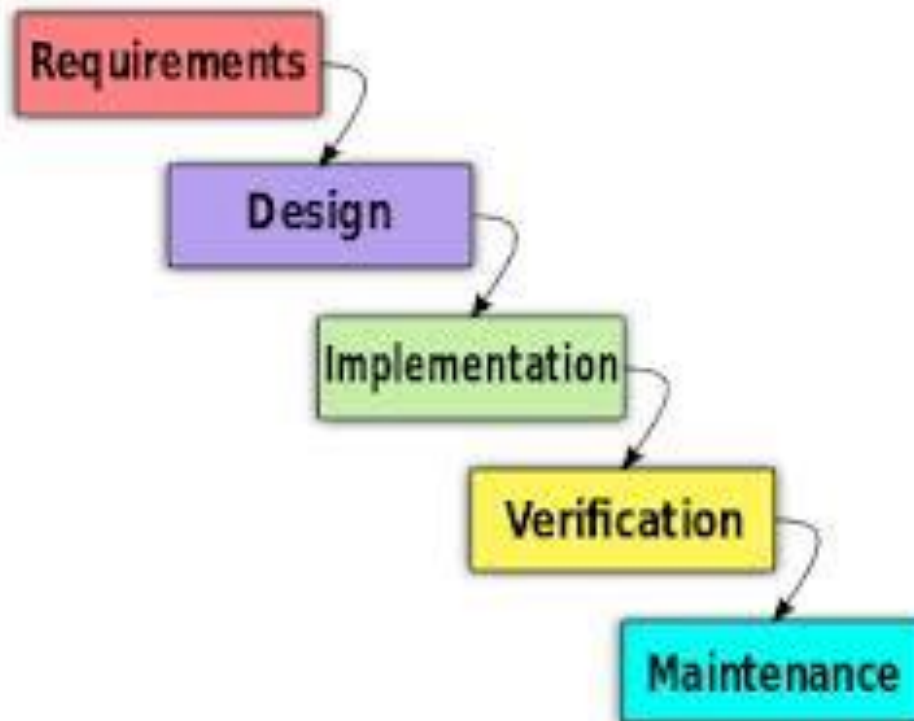


- The cycle is executed many times, depending on the available resources
- The model is prescriptive

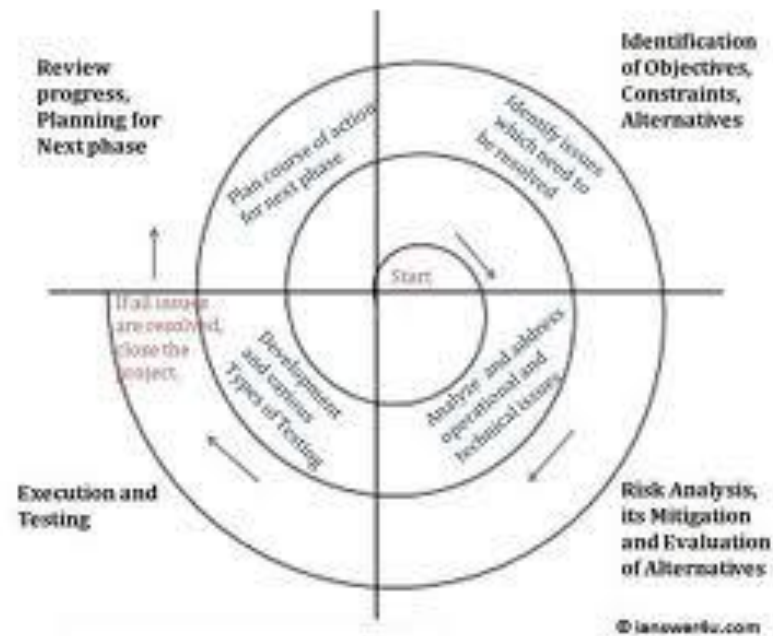
# Discussion

- Try to identify how the evolution of software engineering lifecycle models influences the user-centered design (interaction design)

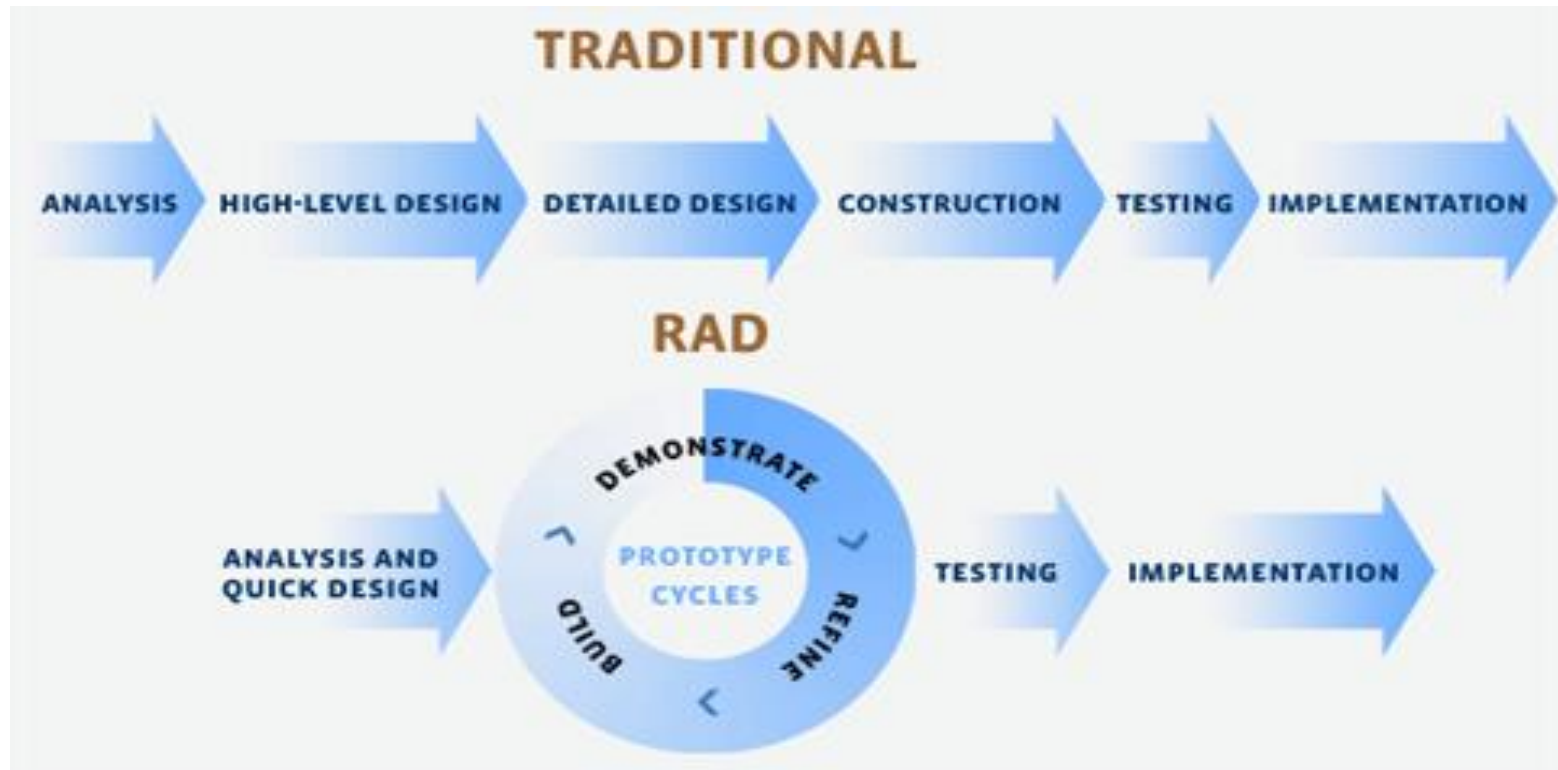
# Discussion- Waterfall Model



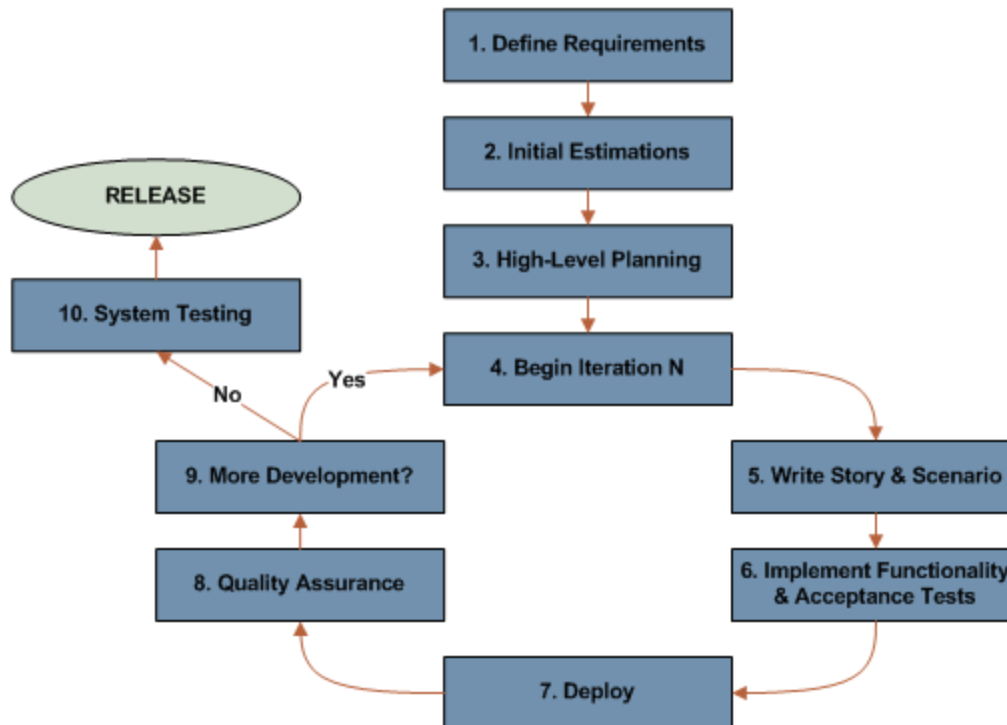
# Discussion- Spiral Model



# Discussion- RAD Model

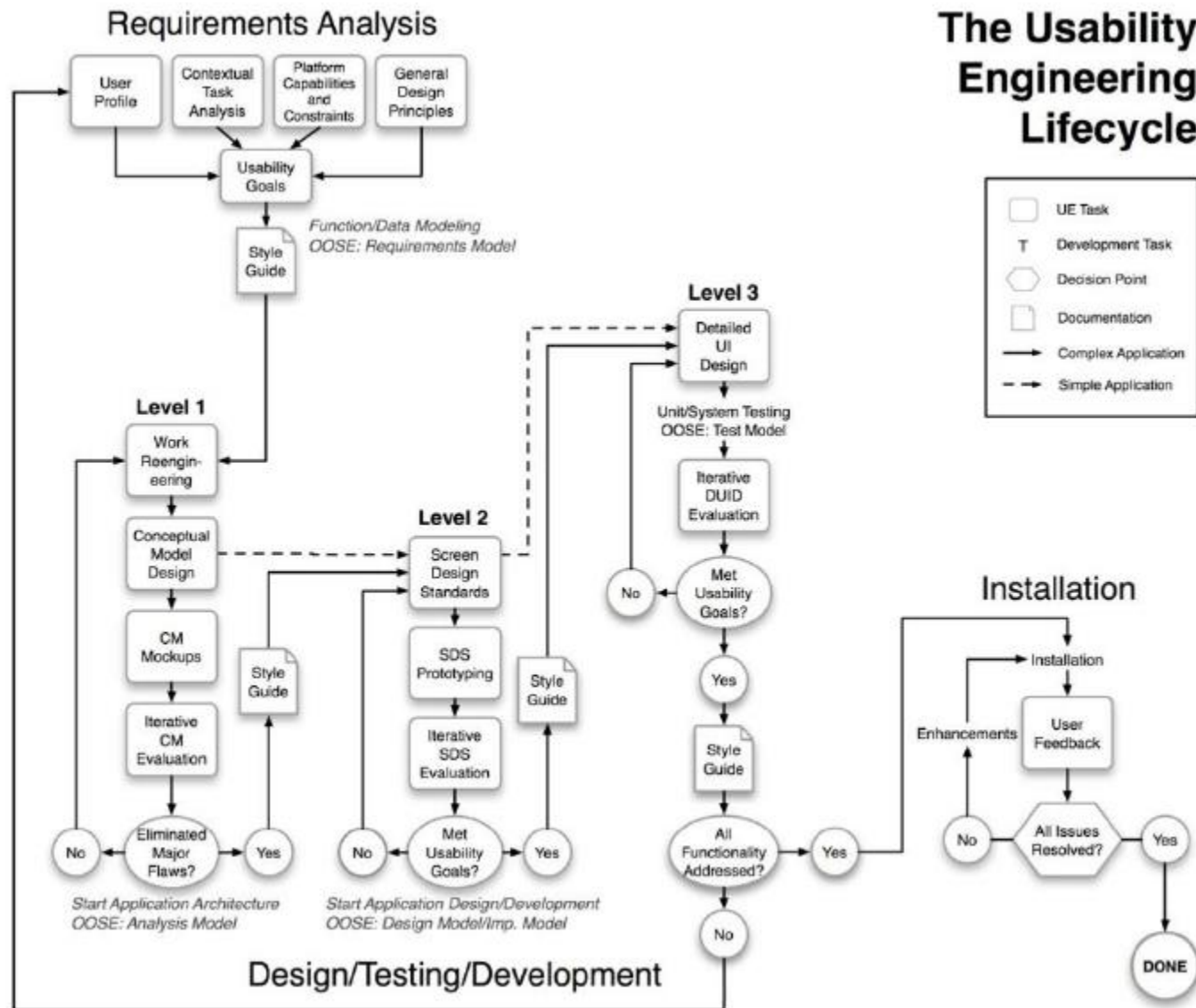


# Discussion- Agile Model



# Usability engineering lifecycle

## (Methodology)

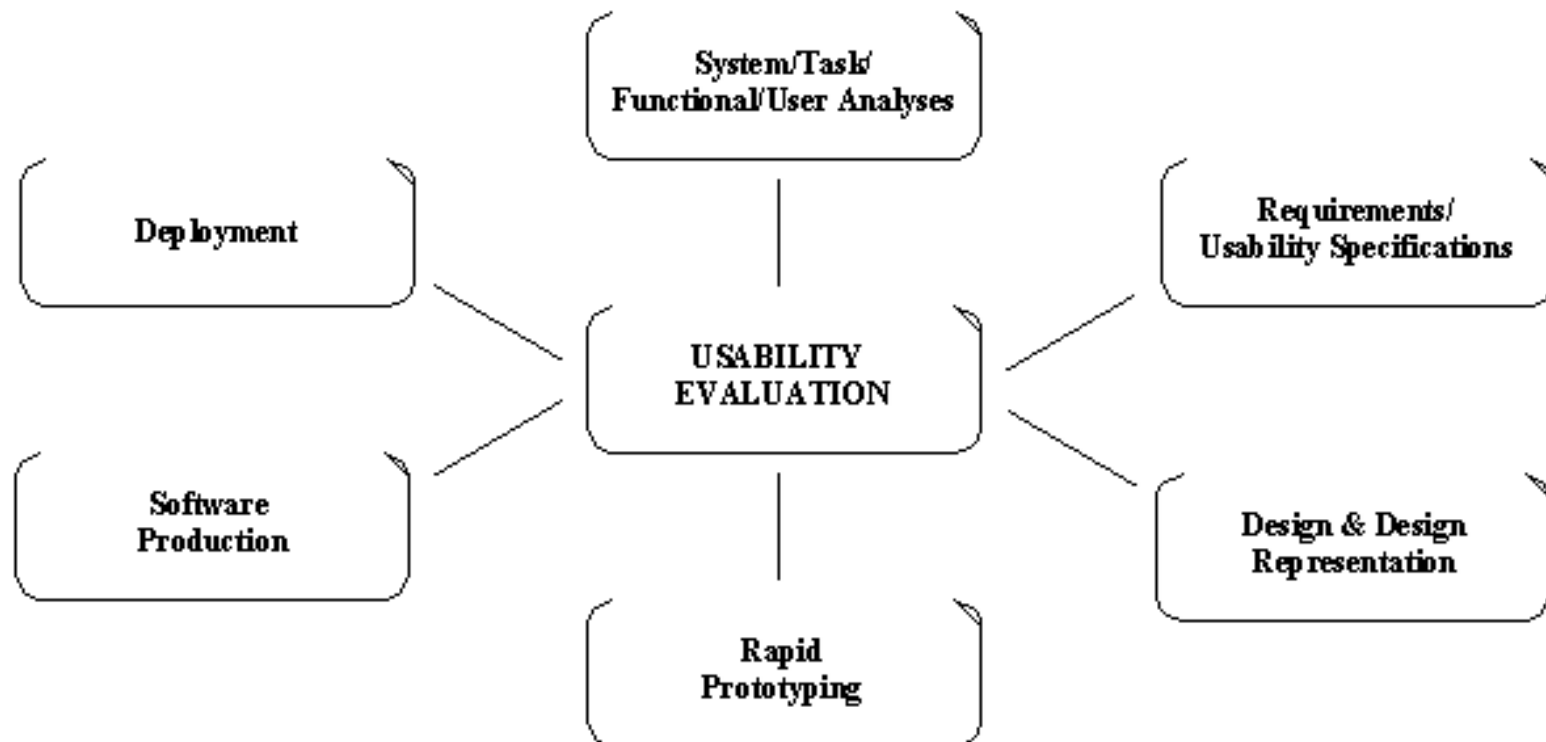


# Usability engineering lifecycle (Mayhew)

- The UEL is a cyclic model that incorporates three phases
- **Requirements analysis**—In this step, you establish your user characteristics, what tasks the product requires for operation so you can determine what the users need to do, set your goals for the usability study, and determine the usability study design guidelines.
- **Design, testing, and development**—In this step, you create a structured, top-down approach to designing the product, be it a user interface, Web site, documentation, or a combination of the three. This is the step that requires the most feedback from your project team.
- **Installation**—In this step, you gather feedback from users during and after the development process and share this feedback with the project team to determine if you need to make any product changes.



# Usability engineering lifecycle (Nielsen)



# The Interaction

- interaction models
  - translations between user and system
- ergonomics
  - physical characteristics of interaction
- interaction styles
  - all the ways the user can communicate or otherwise interact with the computer system

# INTERACTION MODELS

---

# Interaction Models

- terms of interaction
- Norman model
- Interaction Framework – Abowd&Beale

# Some Terms of Interaction

domain – the area of work under study  
e.g. graphic design

goal – what you want to achieve  
e.g. create a solid red triangle

task – how you go about doing it  
– ultimately in terms of operations or actions  
e.g. ... select fill tool, click over triangle

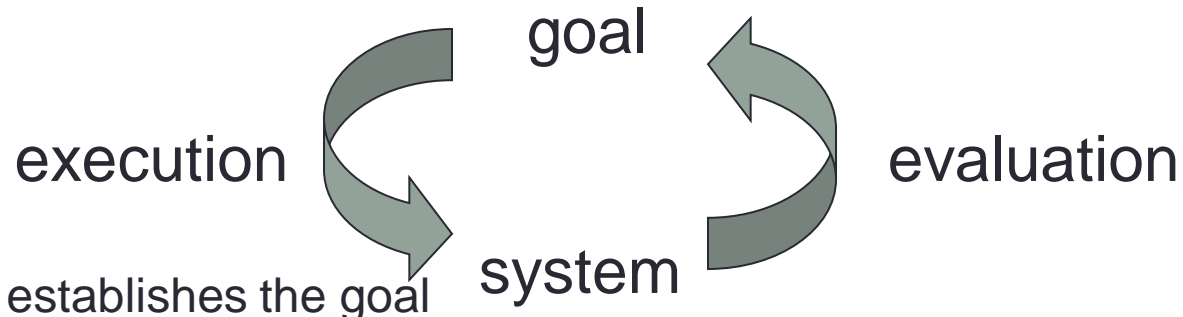
Note ...

- traditional interaction ...
- use of terms differs a lot especially task/goal !!!

# Donald Norman's Model

- Seven stages
  - user establishes the goal
  - formulates intention
  - specifies actions at interface
  - executes action
  - perceives system state
  - interprets system state
  - evaluates system state with respect to goal
- Norman's model concentrates on **user's view of the interface**

# Execution/evaluation loop



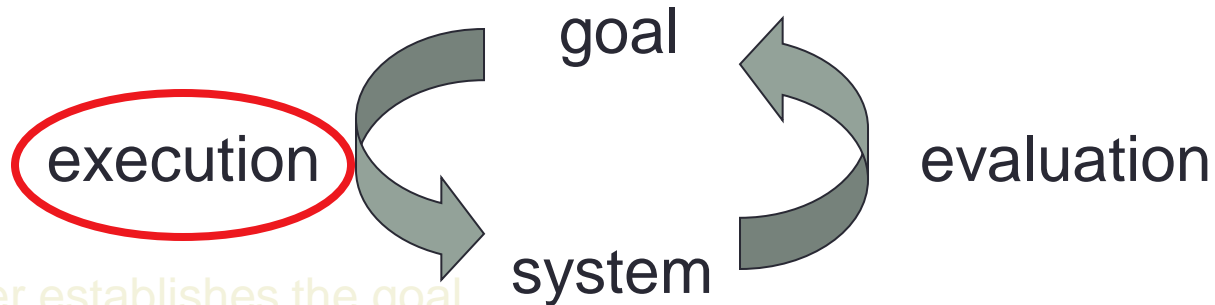
- user establishes the goal
- formulates intention
- specifies actions at interface
- executes action
- perceives system state
- interprets system state
- evaluates system state with respect to goal

# Execution/evaluation loop



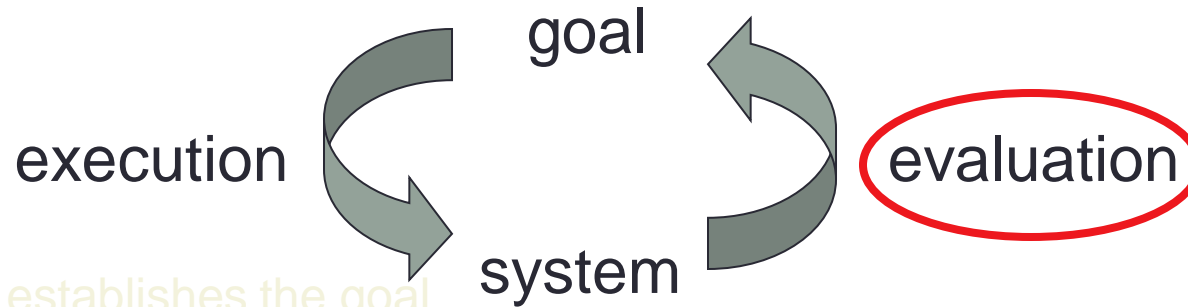


# Execution/evaluation loop



- user establishes the goal
- formulates intention
- specifies actions at interface
- executes action
- perceives system state
- interprets system state
- evaluates system state with respect to goal

# Execution/evaluation Loop



- user establishes the goal
- formulates intention
- specifies actions at interface
- executes action
- perceives system state
- interprets system state
- evaluates system state with respect to goal

# Using Norman's model

Some systems are harder to use than others

## Gulf of Execution

user's formulation of actions

≠

actions allowed by the system

## Gulf of Evaluation

user's expectation of changed system state

≠

actual presentation of this state

# Human Error - Slips and Mistakes

slip



understand system and goal



correct formulation of action



incorrect action

mistake



may not even have right goal!

Fixing things?

slip – better interface design

mistake – better understanding of system

# Abowd and Beale framework

extension of Norman...

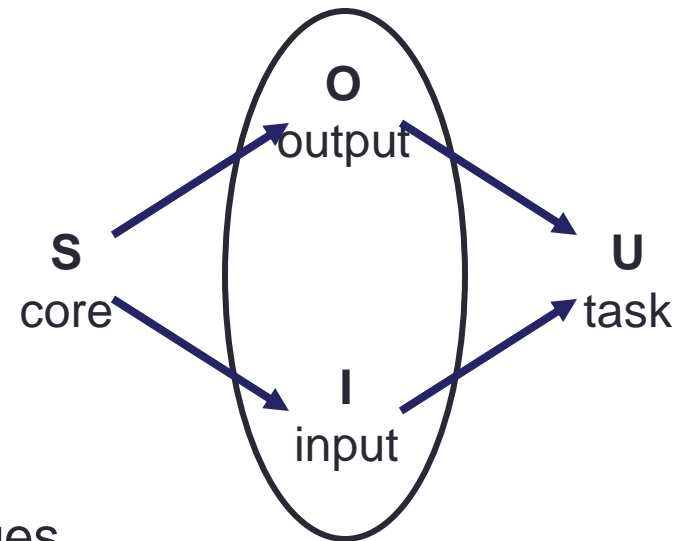
their interaction framework has 4 parts

- user
- input
- system
- output

each has its own unique language

interaction  $\Rightarrow$  translation between languages

problems in interaction = problems in translation



# Using Abowd & Beale's model

user intentions

- translated into actions at the interface
  - translated into alterations of system state
    - reflected in the output display
      - interpreted by the user

general framework for understanding interaction

- not restricted to electronic computer systems
- identifies all major components involved in interaction
- allows comparative assessment of systems
- an abstraction

# Ergonomics

- derives from two Greek words: **ergon**, meaning work, and **nomoi**, meaning natural laws - the science of work and a person's relationship to that
- Ergonomics (human factors) - the scientific discipline concerned with the understanding of interactions among humans and other elements of a system, and the profession that applies theory, principles, data and methods to design in order **to optimize human well-being** and overall system performance.
  - physical aspects
  - cognitive aspects

# Ergonomics

- Study of the physical characteristics of interaction
- Also known as human factors – but this can also be used to mean much of HCI!
- Ergonomics **good at defining standards and guidelines for constraining the way we design** certain aspects of systems



# Ergonomics - examples

- arrangement of controls and displays
  - e.g. controls grouped according to function or frequency of use, or sequentially
- surrounding environment
  - e.g. seating arrangements adaptable to cope with all sizes of user
- health issues
  - e.g. physical position, environmental conditions (temperature, humidity), lighting, noise,
- use of colour
  - e.g. use of red for warning, green for okay, awareness of colour-blindness etc.

# INTERACTION STYLES

---

# Common Interaction Styles

- command line interface
- menus
- natural language
- question/answer and query dialogue
- form-fills and spreadsheets
- WIMP
- point and click
- three-dimensional interfaces
- Post WIMP: gestural, eye tracking, speech

# Command Line Interface

- Way of expressing instructions to the computer directly
  - function keys, single characters, short abbreviations, whole words, or a combination
- suitable for repetitive tasks
- better for expert users than novices
- offers direct access to system functionality
- command names/abbreviations should be meaningful!

Typical example: the Unix system

# Menus

- Set of options displayed on the screen
- Options visible
  - less recall - easier to use
  - rely on recognition so names should be meaningful
- Selection by:
  - numbers, letters, arrow keys, mouse
  - combination (e.g. mouse plus accelerators)
- Often options hierarchically grouped
  - sensible grouping is needed
- Restricted form of full WIMP system

# Natural Language

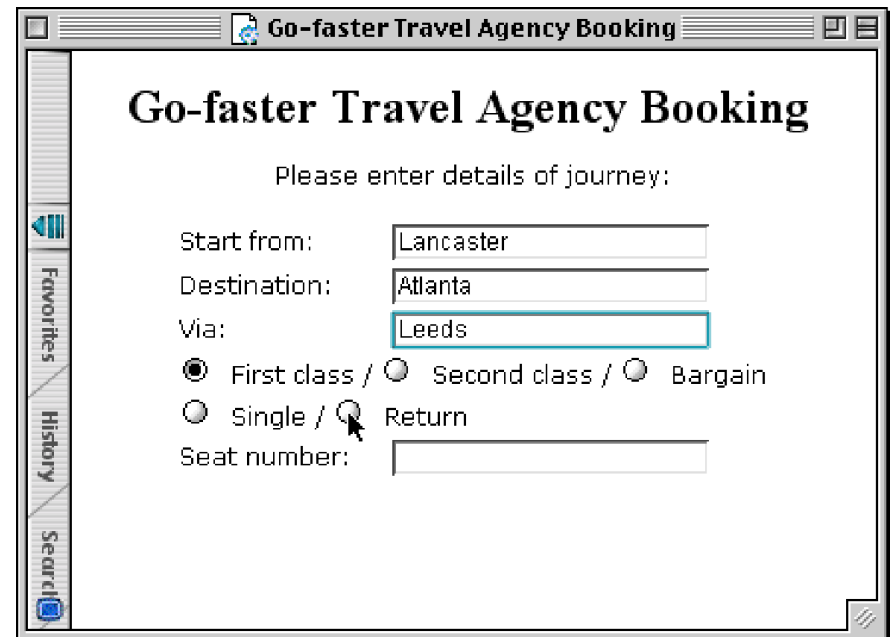
- Familiar to user
- speech recognition or typed natural language
- Problems
  - vague
  - ambiguous
  - hard to do well!
- Solutions
  - try to understand a subset
  - pick on key words

# Query Interfaces

- Question/answer interfaces
  - user led through interaction via series of questions
  - suitable for novice users but restricted functionality
  - often used in information systems
- Query languages (e.g. SQL)
  - used to retrieve information from database
  - requires understanding of database structure and language syntax, hence requires some expertise

# Form-fills

- Primarily for data entry or data retrieval
- Screen like paper form.
- Data put in relevant place
- Requires
  - good design
  - obvious correction facilities



The screenshot shows a web browser window with the title "Go-faster Travel Agency Booking". The page content includes the heading "Go-faster Travel Agency Booking" and the instruction "Please enter details of journey:". Below this, there are several form fields and radio buttons. The "Start from:" field contains "Lancaster", the "Destination:" field contains "Atlanta", and the "Via:" field contains "Leeds". There are three radio buttons for "First class / Second class / Bargain", with "First class" selected. There are two radio buttons for "Single / Return", with "Return" selected. The "Seat number:" field is empty. On the left side of the browser window, there is a vertical sidebar with buttons for "Favorites", "History", and "Search".

Go-faster Travel Agency Booking

Please enter details of journey:

Start from: Lancaster

Destination: Atlanta

Via: Leeds

☒ First class / ☐ Second class / ☐ Bargain

☐ Single / ☒ Return

Seat number:



# Spreadsheets

- first spreadsheet VISICALC, followed by Lotus 1-2-3  
MS Excel most common today
- sophisticated variation of form-filling.
  - grid of cells contain a value or a formula
  - formula can involve values of other cells  
e.g. sum of all cells in this column
  - user can enter and alter data spreadsheet maintains consistency

# WIMP Interface

Windows

Icons

Menus

Pointers

... or windows, icons, mice, and pull-down menus!

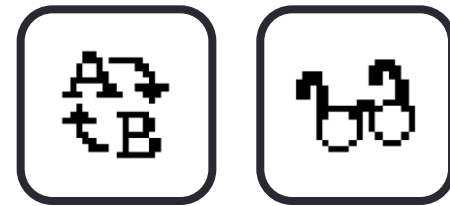
- default style for majority of interactive computer systems, especially PCs and desktop machines

# Point and click interfaces

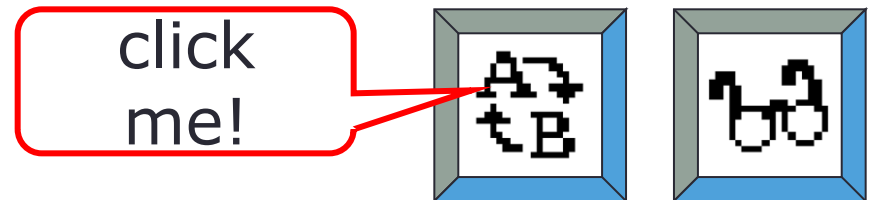
- used in ..
  - multimedia
  - web browsers
  - hypertext
- just click something!
  - icons, text links or location on map
- minimal typing

# Three Dimensional Interfaces

- virtual reality
- ‘ordinary’ window systems
  - highlighting
  - visual affordance
  - indiscriminate use just confusing!
- 3D workspaces
  - use for extra virtual space
  - light and occlusion give depth
  - distance effects



flat buttons ...



... or sculptured

# Elements of the Wimp Interface

- windows, icons, menus, pointers
- +++
- buttons, toolbars,  
palettes, dialog boxes

also see supplementary material  
on choosing wimp elements

# Windows

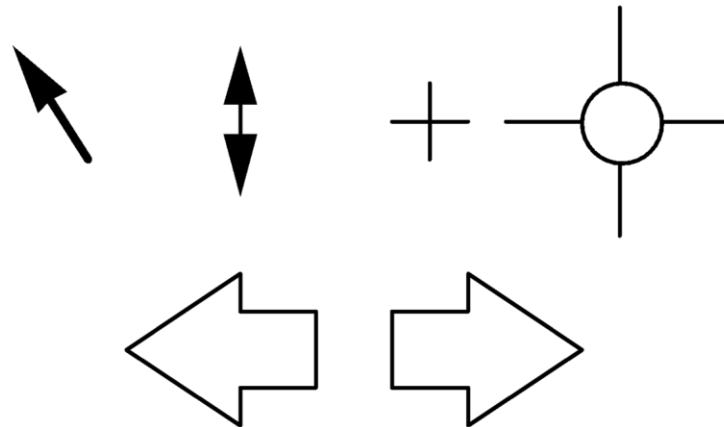
- Areas of the screen that behave as if they were independent
  - can contain text or graphics
  - can be moved or resized
  - can overlap and obscure each other, or can be laid out next to one another (tiled)
- scrollbars
  - allow the user to move the contents of the window up and down or from side to side
- title bars
  - describe the name of the window

# Icons

- small picture or image
- represents some object in the interface
  - often a window or action
- windows can be closed down (iconised)
  - small representation of many accessible windows
- icons can be many and various
  - highly stylized
  - realistic representations.

# Pointers

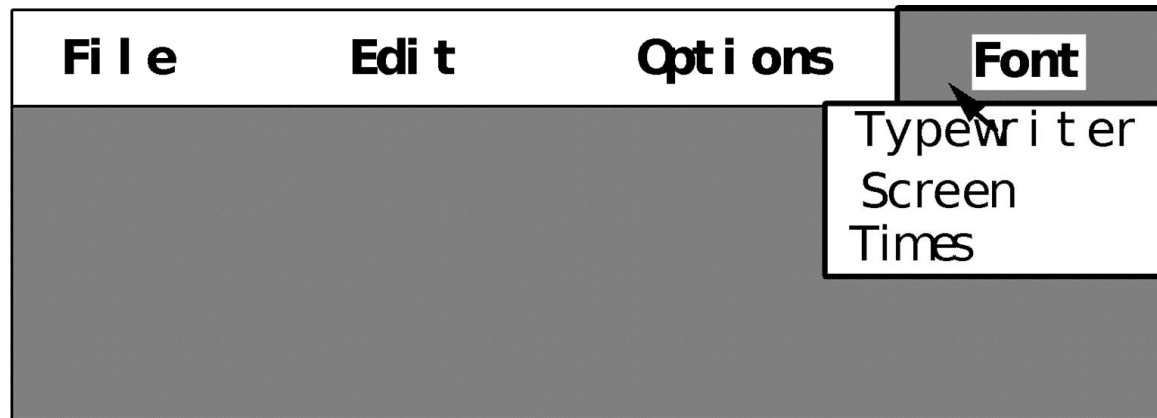
- important component
  - WIMP style relies on pointing and selecting things
- uses mouse, trackpad, joystick, trackball, cursor keys or keyboard shortcuts
- wide variety of graphical images





# Menus

- Choice of operations or services offered on the screen
- Required option selected with pointer

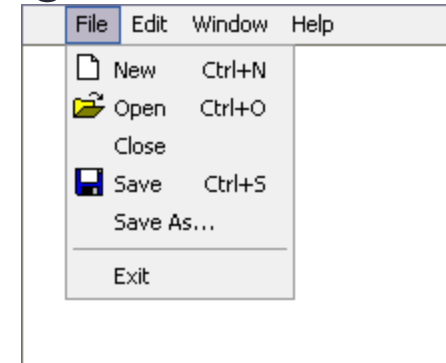


problem – take a lot of screen space

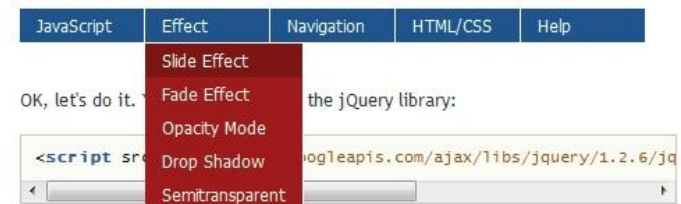
solution – pop-up: menu appears when needed

# Kinds of Menu

- Menu Bar at top of screen (normally), menu drags down
  - pull-down menu - mouse hold and drag down menu
  - drop-down menu - mouse click reveals menu
  - fall-down menus - mouse just moves over bar!
- Contextual menu appears where you are
  - pop-up menus - actions for selected object
  - pie menus - arranged in a circle
    - easier to select item (larger target area)
    - quicker (same distance to any option)  
... but not widely used!



Example:



As you can see, the link points to Google address. There is much probability that the jQuery already cached in the browser.

Now you can create main menu structure using HTML unordered list. Using of an unordered list for navigation is a very good practice. First of all it is

# Menus extras

- Cascading menus
    - hierarchical menu structure
    - menu selection opens new menu
    - and so in ad infinitum
  - Keyboard accelerators
    - key combinations - same effect as menu item
    - two kinds
      - active when menu open – usually first letter
      - active when menu closed – usually Ctrl + letter
- usually different !!!

# Menus design issues

- which kind to use
- what to include in menus at all
- words to use (action or description)
- how to group items
- choice of keyboard accelerators

# Buttons

- individual and isolated regions within a display that can be selected to invoke an action
- Special kinds
  - radio buttons
    - set of mutually exclusive choices
  - check boxes
    - set of non-exclusive choices

Gender: ☐ Male ☒ Female

Interests: ☒ web development ☐ user interfaces ☒ music

# Toolbars

- long lines of icons ...  
... but what do they do?
- fast access to common actions
- often customizable:
  - choose *which* toolbars to see
  - choose *what* options are on it

# Dialog boxes

- information windows that pop up to inform of an important event or request information.

e.g: when saving a file, a dialogue box is displayed to allow the user to specify the filename and location. Once the file is saved, the box disappears.

# Speech-driven interfaces

- rapidly improving ...  
... but still inaccurate
- how to have robust dialogue?  
... interaction of course!

e.g. airline reservation:  
reliable “yes” and “no”  
+ system reflects back its understanding  
“you want a ticket from New York to Boston?”



# Look and ... feel

- WIMP systems have the same elements:  
windows, icons., menus, pointers, buttons, etc.
- but different window systems  
... *behave* differently  
e.g. MacOS vs Windows menus

appearance + behaviour = look and feel

# Initiative

- who has the initiative?
  - old question—answer — computer
  - WIMP interface — user
- WIMP exceptions ...
  - pre-emptive* parts of the interface
- modal dialog boxes
  - come and won't go away!
  - good for errors, essential steps
  - but use with care

# Error and Repair

can't always avoid errors ...

... but we can put them right

make it easy to *detect* errors

... then the user can *repair* them

hello, this is the Go Faster booking system

what would you like?

(user) *I want to fly from New York to London*

you want a ticket from New York to Boston

(user) *no*

sorry, please confirm one at a time

do you want to fly from New York

(user) *yes*

... ..

# Context

Interaction affected by social and organizational context

- other people
  - desire to impress, competition, fear of failure
- motivation
  - fear, devotion, ambition, self-satisfaction
- inadequate systems
  - cause frustration and lack of motivation

# Physical design

- many constraints:
  - ergonomic – minimum button size
  - physical – high-voltage switches are big
  - legal and safety – high cooker controls
  - context and environment – easy to clean
  - aesthetic – must look good
  - economic – ... and not cost too much!

# Fluidity

- do external physical aspects reflect logical effect?
  - related to affordance

logical state revealed in physical state?

e.g. on/off buttons

inverse actions inverse effects?

e.g. arrow buttons, twist controls

# inverse actions

- yes/no buttons
- 'joystick'
- also left side control



# physical layout

controls:

logical relationship

~ spatial grouping





# compliant interaction



state evident in  
mechanical buttons

rotary knobs reveal internal state and  
can be controlled by both user and  
machine

# Managing value

people use something

**ONLY IF**

it has perceived value

**AND**

value exceeds cost

## **BUT NOTE**

- exceptions (e.g. habit)
- value **NOT** necessarily personal gain or money

# Weighing up value

## value

- helps me get my work done
- fun
- good for others

## cost

- download time
- money £, \$, €
- learning effort

# Value and organisational design

- coercion
  - tell people what to do!
  - value = keep your job
- enculturation
  - explain corporate values
  - establish support (e.g share options)
- emergence
  - design process so that  
individuals value → organisational value

# General lesson ...

if you want someone to do something ...

- make it easy for them!
- understand their values

