Human Computer Interface is the study of the interaction between people, computers and tasks.

Human-Computer Interaction involves the development and application of principals, guidelines and methods to support the design and evaluation of interactive systems.

Human Computer Interface is the study of the interaction between people, computers and tasks.

Human-Computer Interaction involves the development and application of principals, guidelines and methods to support the design and evaluation of interactive systems.

* + Computer Science
  + Psychology
  + Sociology
  + Anthropology
  + Ergonomics
  + Physiology
  + Design
  + Art
  + Engineering
  + Linguistics
  + Philosophy
  + Artificial Intelligence

Neilsen’s 10 Heuristics

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards.
5. Help users recognise, diagnose and recover from errors
6. Error prevention
7. Recognition rather than recall
8. Flexibility and efficiency of use
9. Aesthetic and minimalist design
10. Help and documentation

Summary of lecture 1

* Introduction to the subject of HCI
* Disciplines associated with HCI
* Humans (Users) and Interfacing
* Computers and Interfacing
* Interaction
* Reasons for our interest in HCI
* Usability Principles (Nielsen's Heuristics)

Impact on Design

**Nondiscretionary**

Productivity is the most important factor

Interface is still important, but the user will be willing to invest time and money for training

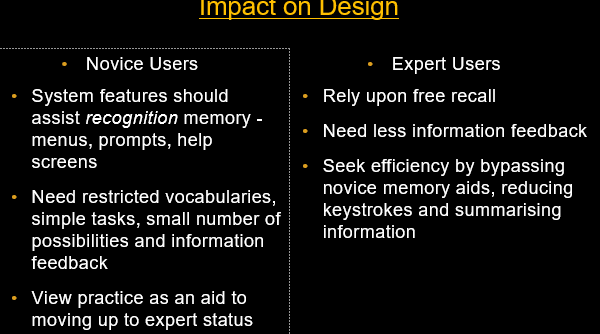
Niche application area, one type of user

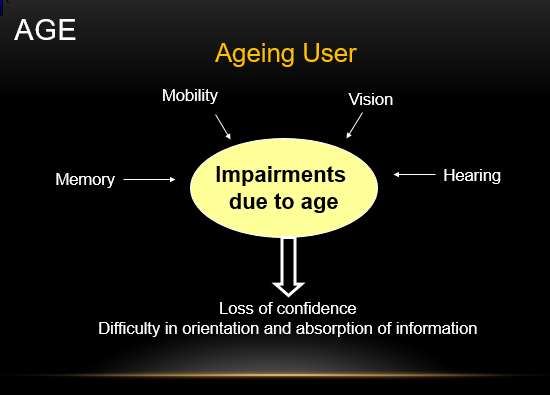
Interface is extremely important, first impression of the system

Must cater for many different user types

Must sell your product to user

If system is not perceived as achieving results with minimum effort, may be refused





Young User

* + Active learners drawing on physical and social experiences
  + Learn by doing
  + Require something that they can see, touch and hear
  + Prior to the age of 8, children rely on visual and auditory perception for knowledge
  + Learning is through first hand experience

Physically Disabled

* + Impairments can include lack of mobility, low vision, blind and hearing…
  + Alternative channels of interactions:
    - Speech recognition
    - Tune recognition
    - Lip-reading
    - Body-electric sensors
  + The more sophisticated and complex a system is, the more specialised its target group gets.

Cognitively Disabled

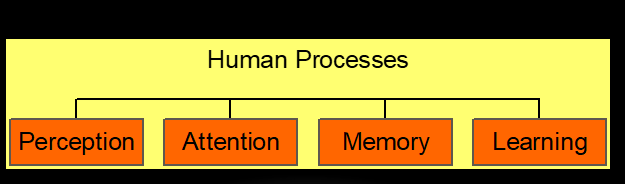
* + Examples of cognitive impairments would be learning disabilities, dyslexia and poor memory.
  + Changes can be made to accommodate these users by improving layouts, controlling vocabulary, and limiting short-term memory demand.

Cultural and international diversity

* + Characters, numerals, special characters, and diacriticals (those that distinguish meanings)
  + Left-to-right versus right-to-left, versus vertical input and reading
  + Date and time formats
  + Numeric and currency formats
  + Weights and measures
  + Telephone numbers and addresses
  + Names and titles (Mr., Ms., Mme.)

The journal, *Ergonomics Abstracts* offers this classification of human cognitive processes:

* + - Long-term and semantic memory
    - Short-term and working memory
    - Problem solving and reasoning
    - Decision making and risk assessment
    - Language communication and comprehension
    - Search, imagery, and sensory memory
    - Learning, skill development, knowledge acquisition and concept attainment

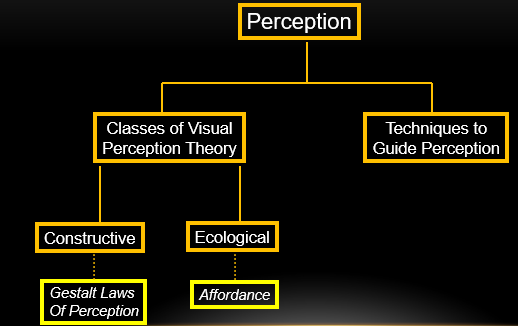


Perception is fundamental to interacting with computers.

To use a computer, a user needs to perceive information that is presented by the interface.

Perception can involve all the different senses (visual, hearing, touch, smell...)

To date, HCI is mostly concerned with visual perception, because the computer screen is usually the interface.



Gestalt laws of perceptual organisation

(Gestalt = shape, and usually a shape made from more that one part)

‘Laws’ of perception that are regarded as being innate

**Proximity**

**Similarity**

**Closure**

**Continuity**

**Symmetry**

Proximity

The elements appear as groups rather than a random cluster of elements

Similarity

There is a tendency for elements of the same shape or colour to be seen as belonging together

Closure

Missing parts of the figure are filled in to complete it, so that it appears as a whole circle

Continuity

The stimulus appears to be made of two lines of dots traversing each other, rather than a random set of dots

Symmetry

Regions bounded by symmetrical borders tend to be perceived as coherent figures

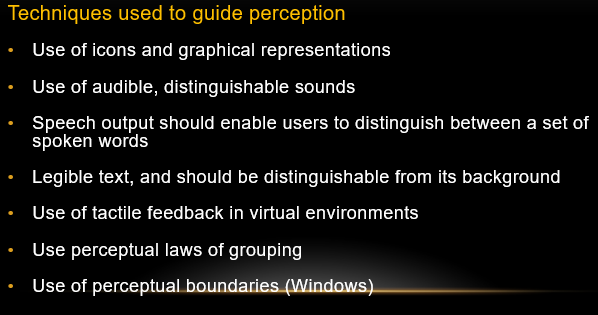
**Ecological theory** argues that visual perception is a direct process. That is, perception is made by the observer, specific to him or her.

Information is simply detected, not constructed

Ecological theory is not really concerned with how we make sense of a scene, more that the scene has a method of presentation.

(How we interpret, how we ‘make sense’, is better related to constructivism.)

A central concept of the ecological approach is the notion of affordance. (This is the ‘readiness to hand’ that make websites, application interfaces and gadgets physically easier to use.)



Cognitive frameworks help us to understand how users become familiar with a system.

Two common frameworks are:

Mental Models

Metaphors

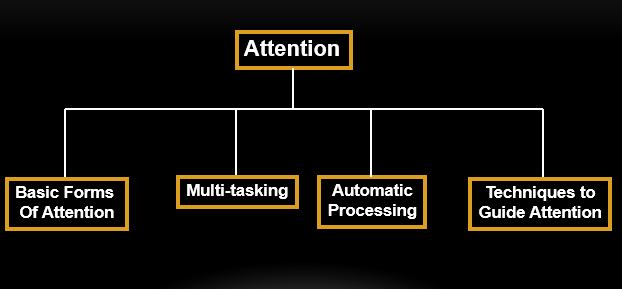
Mental modelscan be describedas “internal constructions of some aspect of the external world enabling predictions to be made”

A metaphor is a figure of speech in which an expression is used to refer to something that it does not literally denote, in order to suggest a similarity.

Summary 2

* + Users
    - Nature of the User
    - Human Considerations
  + Cognitive Processes
    - Perception
    - Attention
    - Memory
    - Learning
  + Cognitive Frameworks
    - Mental Models

Metaphors

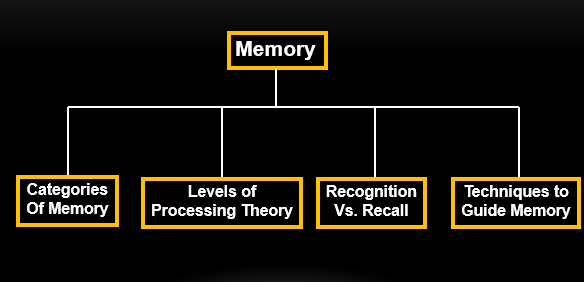


Basic Forms of Attention

* Our ability to attend to one event out of a mass of stimuli is known as focused attention.
* We are also capable of divided attention, e.g. driving while holding a conversation.
* Attention may also be voluntary (we make a conscious effort to change our attention) or involuntary(a stimuli suddenly grabs our attention).

Implications for HCI Design

* Consider a user who has learned a set of keyboard combinations for a particular word processor to the extent that they have become automatic processes.
* If key combos have been changed -> major effort for the user to unlearn their automatic processes.
* That leads to much frustration on the part of the user.
* The change could be dangerous, potentially, in critical applications such as process control plants.



Memory is involved in all our cognitive tasks and has 3 major components:

Sensory memory –

‘Snapshot' of our environment, stores this information for a short period

Act as buffers for stimuli received through the senses

Holds information for a short time e.g., visual information fades away in less than a second

Short-term memory –

Information of the present

Information retained automatically and retrieved without effort

Amount of information that can be retained is severely limited – Miller’s *chunking* concept (7 +/- 2)

Long-term memory –

Information of the past

Amount of information that can be retained is unlimited

The familiarity of a word or concept refers to the frequency with which it occurs on everyday language:

* + ‘Door’, ‘read’ and ‘stop’ are examples of familiar words while ‘compile’ and ‘scan’ are examples of unfamiliar words

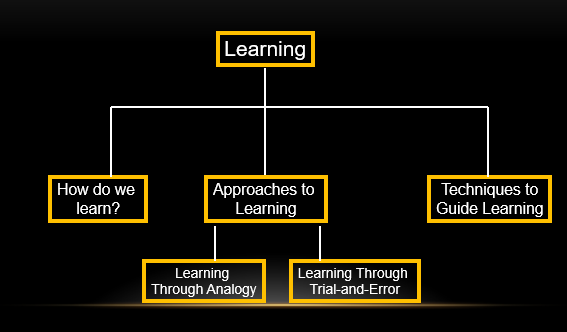
Imagery refers to the ability a word or concept can elicit images in one’s mind:

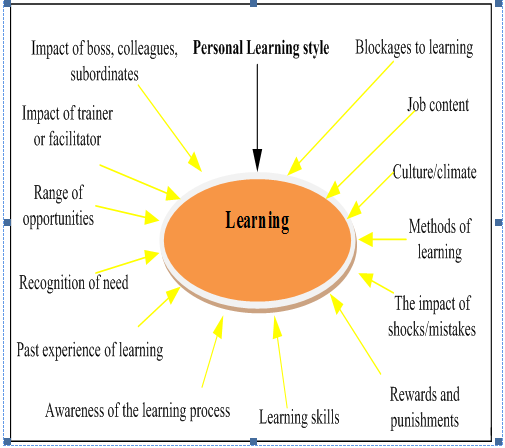
* + High imagery words are ‘eat’ and ‘sleep’ while low imagery words are those such as ‘begin’ and ‘evaluate’

Memory Aids:

To overcome the "bottleneck" of working memory, humans have devised memory aids.

* + Acronyms - Acronyms are formed by taking the first letter of each word in a group of words and creating a new word.
    - * NATO – North Atlantic Treaty Organisation
      * CIE – Coras Iompair Eireann
  + Acrostics/Sentences - Related to acronyms is the use of acrostics or taking the first letter of words and, instead of creating a new word, the letters are used to make a sentence.
  + Rhymes/Songs - Rhythm, melody, rhyme aid memory.





How do people learn?

Nobody really knows, but there are 6 main theories:

1. Behaviourism

* Learning is defined by the outward expression of new behaviours.
* It focuses solely on observable behaviours.
* It describes a biological basis for learning.
* Learning is context-independent.

1. Cognitivism

* Often has inquiry-oriented projects.
* There ought to be opportunities for the testing of hypotheses.
* Curiosity is encouraged.

1. Humanist Theory -
   1. Social Learning Theory

* Learners construct knowledge for themselves.
* Learning is an active process.
* Learning is a social activity.
  1. Social Constructivism

1. Multiple Intelligences
2. Brain-Based Learning

There are two categories of errors:

Mistakes –

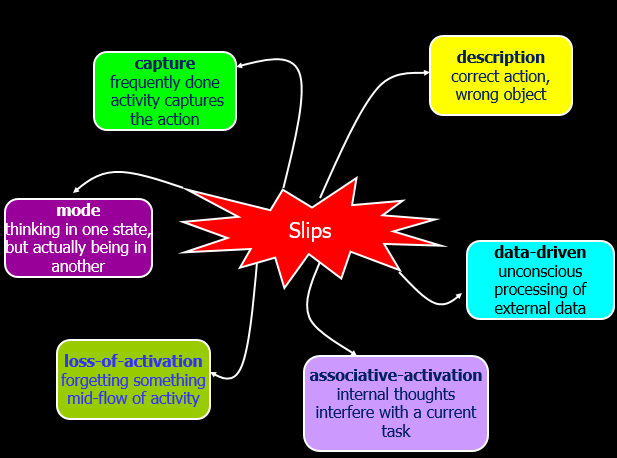
Incorrect mental model, incorrect action is taken based on an incorrect decision

Occur intentionally (i.e. The intentional behaviour is not recognised as a mistake until later)

Slips –

Unintentional errors

Skilled behaviour



Error Prevention Guidelines continued

Provide an appropriate type of response -

Prevent user from continuing

Warn user an unusual situation is occurring

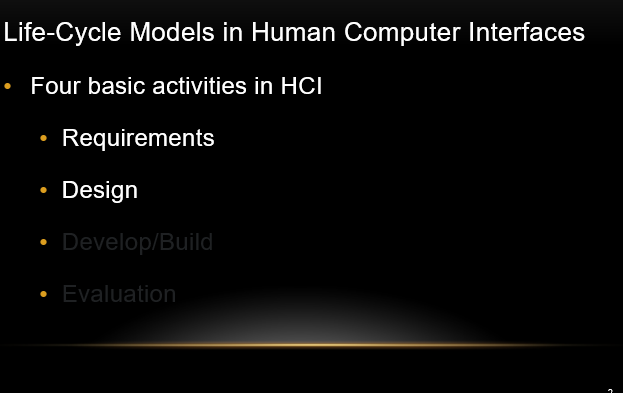
Self-correct - spell-check correction

System opens dialog with user - go into debugger on run-time crash

Summay of lectuture 3

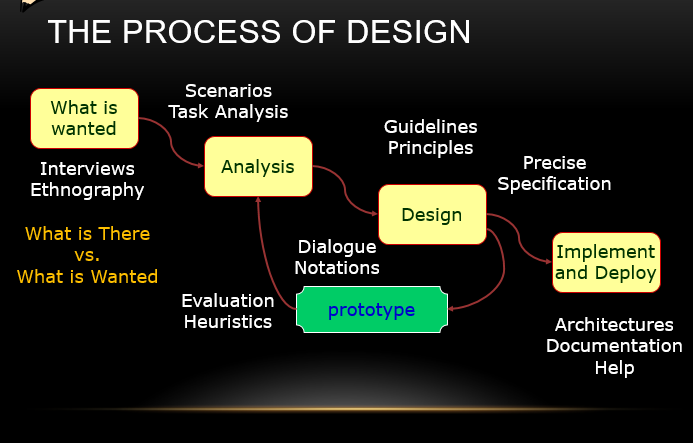
* + Perception
  + Attention
    - Basic forms of attention
    - Multi-tasking and interruptions
    - Automatic processing
    - Techniques to guide attention
  + Memory
    - Categories of memory
    - Levels of Processing Theory
    - Recognition vs. Recall
    - Techniques to guide memory
  + Learning

Learning theories, learning types

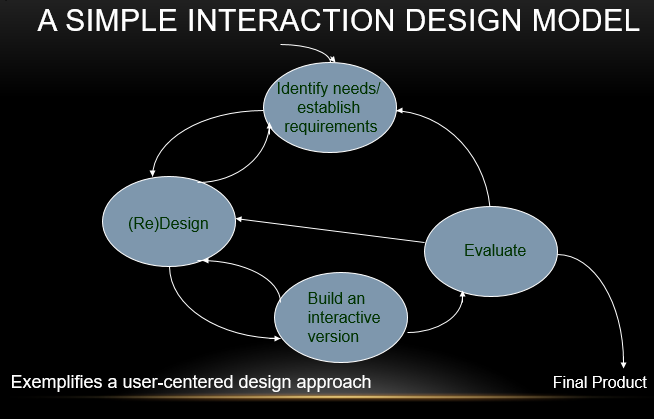


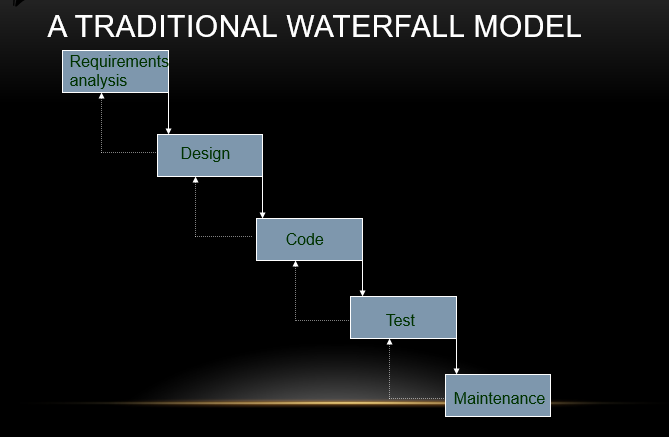
The golden rule of design: understand your materials

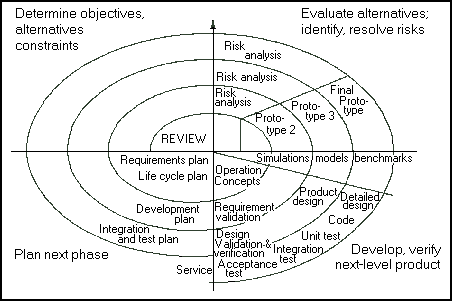
* + - * + Understand computers, and their:
  + limitations, capacities, tools, platforms
* Understand people – their:
  + psychological, social aspects
  + human errors
* and their interaction…
* Accident reports...
  + Aircrash, Industrial Accident, Hospital Mistakes…
  + Enquiry… blames… ‘human error’
* But… if a concrete lintel breaks because too much weight in a wall, do we blame ‘lintel error’?  
  …No – this is a design error. We know how concrete behaves under stress.
* Human ‘error’ is normal.
  + We know how users behave under stress…
  + …so design for it!
* Treat the user at least as well as physical materials.

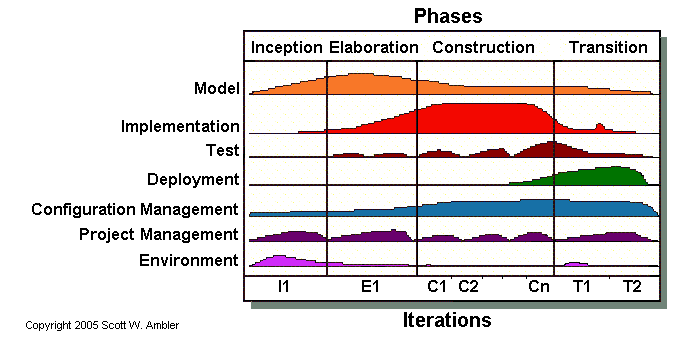


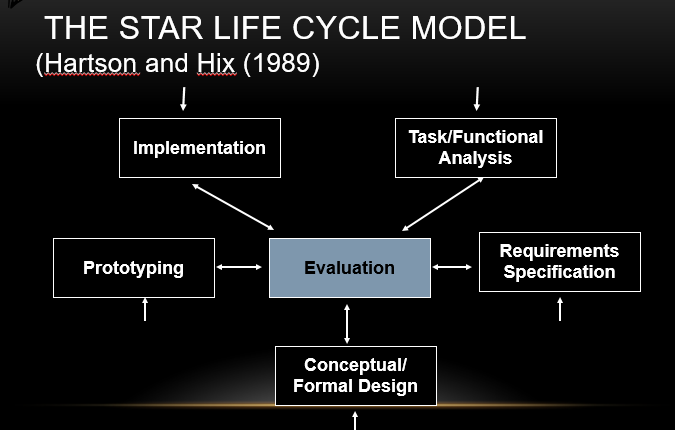
1. Requirements
   * What is there and what is wanted
2. Analysis
   * Placing in order, getting an understanding
3. Design
   * What to do, and how to decide on options
4. Iteration and Prototyping
   * Getting the design right… Finding what is really needed!
5. Implementation and Deployment
   * Making the system, and getting it out there

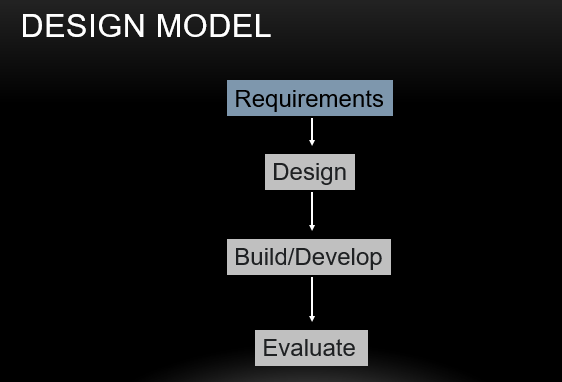












A requirement is something the product must do, or a quality that the product must have.

Different kinds of requirements:

* + - Functional:
      * What the system should do
      * Historically the main focus of requirements activities
    - Non-functional:
      * memory size,
      * response time…

Users’ Involvement in Design

* Know your user
* Personae (plural of ‘persona’)
* Cultural investigation

Expectation management

* + Realistic expectations
  + No surprises, no disappointments
  + Timely training
  + Communication - but no hype
* Ownership
  + Make the users active stakeholders
  + They might be more likely to forgive or accept problems
  + They can make a big difference to acceptance and success of the product
* Early focus on users and tasks: directly studying cognitive, behavioural, anthropomorphic and attitudinal characteristics.
* Empirical measurement:users’ reactions and performance to scenarios, manuals, simulations and prototypes are observed, recorded and analysed.
* Iterative design: when problems are found in user testing, fix them and carry out more tests.
* Preparation
  + Understand the organisation’s policies and work culture.
  + Familiarise yourself with the system and its history.
  + Set initial goals and prepare questions.
  + Gain access and permission to observe/interview users.
* Field Study
  + Establish rapport with managers and users.
  + Observe/interview users in their workplace and collect subjective/objective quantitative/qualitative data.
  + Follow any leads that emerge from the visits
* Analysis
  + Compile the collected data in numerical, textual, and multimedia databases.
  + Quantify data and compile statistics.
  + Reduce and interpret the data.
  + Refine the goals and the process used.
* Reporting
  + Consider multiple audiences and goals.
  + Prepare a report and present the findings.
* Participatory (or Participative) Design
* This is an approach to design that attempts to actively involve the end users in the design process.

However, extensive user involvement may:

* + be more costly
  + lengthen the implementation period
  + build antagonism with people not involved or whose suggestions rejected
  + force designers to compromise their design to satisfy incompetent participants
  + build opposition to implementation
  + exacerbate personality conflicts between design-team members and users
  + show that organisational politics and preferences of certain individuals are more important than technical issues
* **PICTIVE**
* Plastic Interface for Collaborative Technology Initiatives through Video Exploration .
* This method is intended to empower users to act as full participants in design.

**CARD**

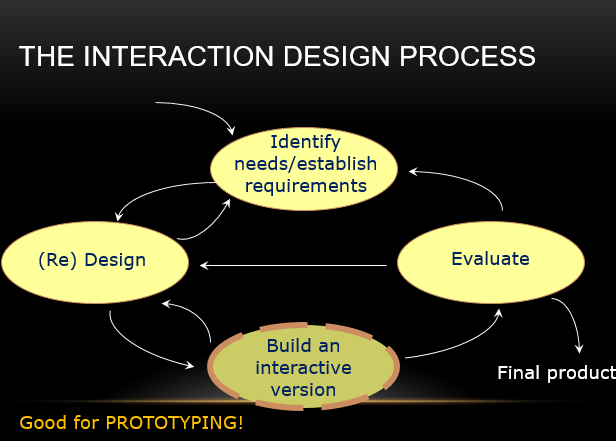
Collaborative Analysis of Requirements and Design

* Similar to PICTIVE, but at a higher level of abstraction: explores work flow not detailed screen design.
* Uses playing cards with pictures of computers and screen dumps.
* Similar structure to the session as for PICTIVE.
* PICTIVE and CARD can be used together to give complementary views of a design.

Summary 3

Lifecycle Models

* + Software engineering lifecycle models
  + HCI lifecycle models
    - Usability Engineering Lifecycle Model
    - Star Lifecycle Model
* HCI Design Models
  + Requirements
  + Design
    - User-Centred Design
  + Develop/Build
  + Evaluation



In HCI design it can be (among other things):

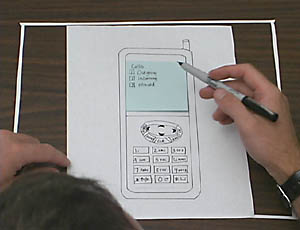
* + a series of screen sketches
  + a storyboard, i.e. a cartoon-like series of scenes
  + a PowerPoint slide show
  + a video simulating the use of a system
  + a lump of wood (e.g. PalmPilot – Google ‘PalmPilot wooden model’)
  + a cardboard mock-up
  + a piece of software with limited functionality, written in the target software language or in another language
  + Technical issues
  + Work flow, task design
  + Screen layouts and information display
  + Difficult, controversial or critical areas of systems design
* For software, you might prototype at various times in the lifecycle.
  + There may be different goals, you may be using different techniques.
* During:
  + Conceptual Design
  + Interaction Design
  + Screen Design
* Early in development
* Explore high-level issues
  + Different conceptual models
  + Interaction styles
  + User needs and characteristics
  + Usability goals
* High-level representations
  + Far from final code or GUIs
* Later in development
* Focus on user work-flows
  + Tasks and scenarios you’ve identified
* Might focus at the screen (or page) level. Possibly like this:
  + identify screens, pages, activities
  + Organise these into groups
  + Define flows or transitions between them
* Involve users in the evaluation
* Representations
  + Still probably not much like final code or GUIs
* Before development
* Define and refine screens (pages)
  + Blue-prints for final physical design
* User evaluation
  + Both achieving tasks and navigation, and  
    other usability criteria (as we’ve studied)
* Representations
  + Low-fidelity or high-fidelity prototypes

‘Fidelity’ refers to the level of detail, and how much the prototype matches the finished article:

* + - Low Fidelity
    - Medium Fidelity
    - High Fidelity

‘Low fidelity’ uses a medium which is unlike the final medium, e.g. paper, cardboard

* It is quick, cheap and easily changed
* Examples:  
   Sketches of screens, task sequences, etc  
   ‘Post-it’ notes  
   Storyboards  
   ‘Wizard-of-Oz’



Storyboards are often used with scenarios, bringing more detail of the design

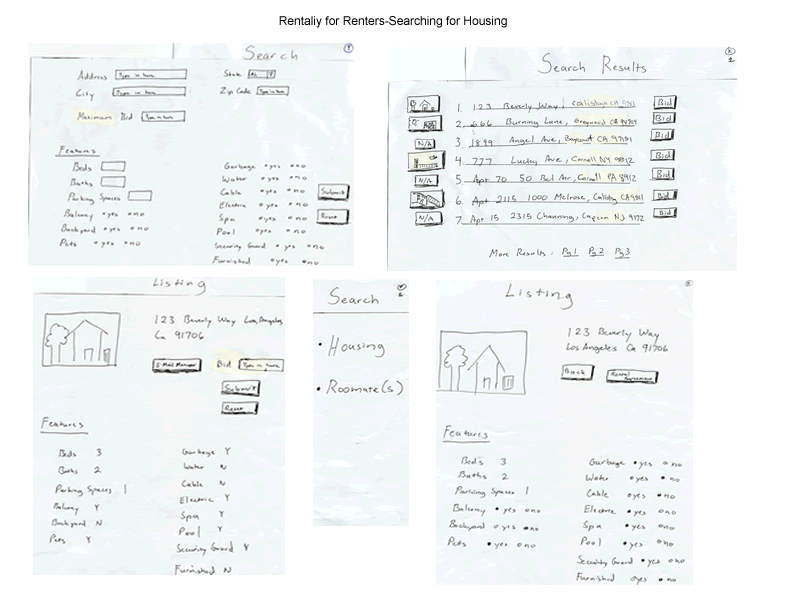
* It is a series of sketches showing how a user might progress through a task using the device
* Used early in the design process

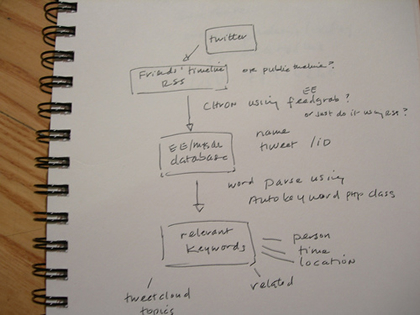
Storyboards are:

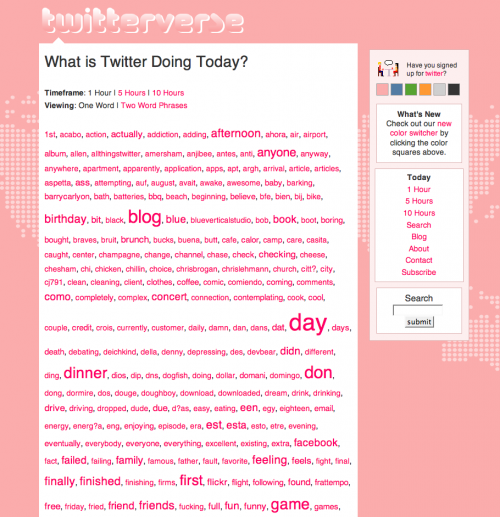
* + a low fidelity visual representation where
  + steps or actions represented by panels, like a comic book

Goals are to

* + flesh out the scenarios in an interaction design
  + effectively communicate with users or stakeholders







Prototyping with a computer

* + Simulating some, but not all, features of the interface.
  + This can be engaging for end users.

Purpose

* + Provides sophisticated, but limited, scenarios for the user to try.
  + You can test more subtle design issues.

Dangers

* + Users’ reactions are often “in the small”.
  + Users often reluctant to challenge the designer.
  + Users often reluctant to touch the design.
  + Management may think it is real!
* ‘High fidelity’ uses materials that you would expect to be in the final product.
* The prototype looks more like the final system than a low-fidelity version.
* Danger - that users think they have a full system...

Benefits of high-fidelity prototypes

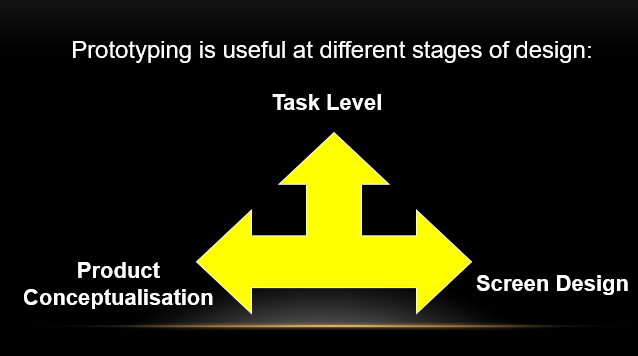
* + The most realistic prototype
  + Closer to final product

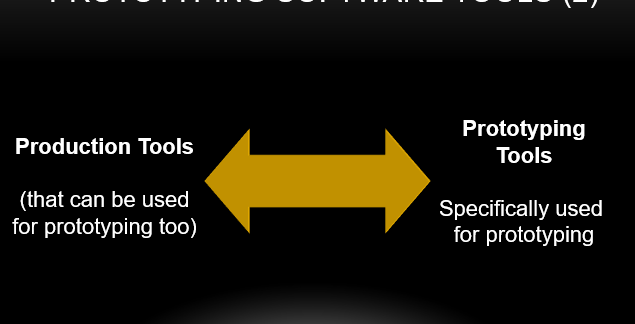
- So good for developers and users

* + You can collect metrics at this point
* Limitations
  + More expensive, less rapid than low or medium fidelity prototypes
  + Reluctance to change when prepared

Two common types of compromise:

* Horizontal prototype – shows a user interface, but has limited functionality behind the buttons/controls. No database links included.
* Vertical prototype – contains all of the high level and low level functionality of particular areas in the system.
* Requirements Animation – functional requirements  
  are demonstrated in a s/w prototype
* Throw-away prototyping– similar to requirements animation... and prototype is not developed into the final product
* Evolutionary prototyping – prototype is not discarded and is used as the basis for the next iteration of design. I.e. system “evolves” from prototype
* Incremental prototyping – final product is built as separate components, one at a time – allows large systems to be installed in phases to avoid delays between specification and delay
* Requirements Animation – functional requirements  
  are demonstrated in a s/w prototype
* Throw-away prototyping– similar to requirements animation... and prototype is not developed into the final product
* Evolutionary prototyping – prototype is not discarded and is used as the basis for the next iteration of design. I.e. system “evolves” from prototype
* Incremental prototyping – final product is built as separate components, one at a time – allows large systems to be installed in phases to avoid delays between specification and delay





Production tools:

* produce re-usable software
* The constraints of producing quality software are not necessarily compatible with prototyping
* The code management overhead can slow production
* may cause higher costs of producing the prototype
* The methods (tools) require development skills

Prototyping specific tools:

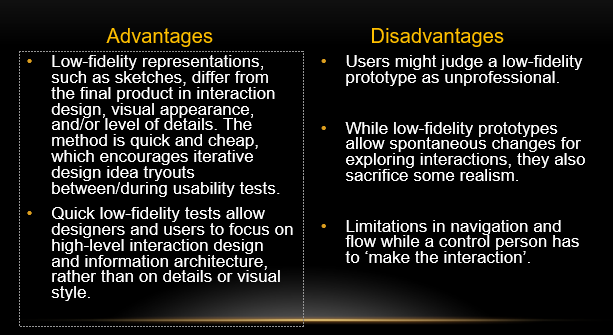
* are faster
* can be used by non-technical staff
* must be ‘throw-away’
* may not look exactly like future system
* may limit what can be configured

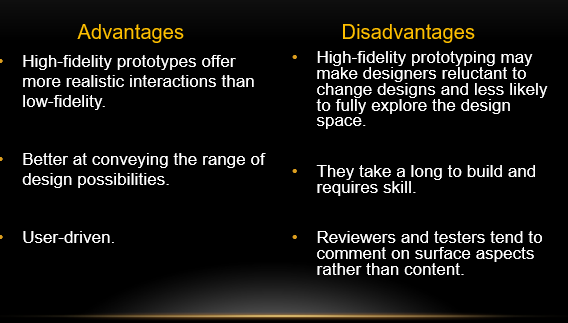
Examples:

* Pressure to enhance prototype to become delivered system
  + From the client
  + From management
    - When these people see code, see an almost-working “system”
* Why not use the prototype?
* Prototypes are built for quick updates, so...
  + No real design, therefore hard to maintain
  + Code can be unstructured, no error checking
  + The prototype may be in the ‘wrong environment’

Examples:

* Pressure to enhance prototype to become delivered system
  + From the client
  + From management
    - When these people see code, see an almost-working “system”
* Why not use the prototype?
* Prototypes are built for quick updates, so...
  + No real design, therefore hard to maintain
  + Code can be unstructured, no error checking
  + The prototype may be in the ‘wrong environment’
* Construction: taking the prototypes (or learning from them) and creating a final product.





* HCI Design Models
  + Requirements
  + Design (a quick mention, based on last week)
    - User-centred design
  + Develop/Build
    - Prototyping
      * Why, how…
      * Low fidelity prototyping (medium fidelity, very briefly)
      * High fidelity prototyping
      * A quick, introductory mention of Evaluation
* HCI Design Models
  + Requirements
  + Design (a quick mention, based on last week)
    - User-centred design
  + Develop/Build
    - Prototyping
      * Why, how…
      * Low fidelity prototyping (medium fidelity, very briefly)
      * High fidelity prototyping
      * A quick, introductory mention of Evaluation