**Human Computer Interaction**

• Is a discipline concerned with the design, evaluation and implementation of interactive systems for human use and with study of major phenomena surrounding them.

**What is HCI?**

* HCI (human-computer interaction) is the study of interaction between people (users) and computers.
* Interaction between users and computers occurs at the user interface
* The golden principle in HCI is that “people should come first”.
* **Human**: could be an individual user or a group of users.
* **Computer**: could be any technology ranging from the general desktop computer to a large scale computer system.
* **Interaction**: any direct or indirect communication between a human and computer.

**HCI is not about**

* Making the interface look pretty
* Only about desktop computers (and that goes for computing as well!)
* Something that would be nice to do but usually there’s no time for it

**HCI is about**

* Understanding the users
* Understanding users tasks
* Understanding the surrounding environment
* GUI requirements gathering and analysis
* Design prototype
* Evaluate the system

**The goal of HCI** “is to develop or improve the safety, utility, effectiveness, efficiency and usability of system that include computers.”

**What is Usability?**

* A usable system is:

– easy to use

– easy to learn

– easy to remember how to use

– effective to use

– Set Theory

– efficient to use

– safe to use

– enjoyable to use

**Disciplines Contributing to Human-Computer Interaction.**



**Is HCI really Important?**

**Example about: Health and safety concerns**

If the video doesn’t record a TV program because we pressed the wrong button, we are likely to feel angry. A real example: a pilot shuts down the wrong engine and the plane crashes (as happened near Leicestershire, in England on the M1 motorway in 1989), this is obviously more serious. 47 died.

**Reasons for Failures**

Projects in general fail for various reasons:

* lack of senior management commitment
* lack of user involvement
* lack of user requirements specifications
* poor project planning and team problems

**The importance of HCI**

1. Can Preventing accidents

2. Health and safety concerns

3. Can reduce the cost of customer training and support

4. Direct correlation between HCI and sales

5. HCI can provide you a job.

**User Centered Development**

1. Data Collection

2. Data Analysis

3. Prototyping

4. Design

5. Evaluation

**Data Collection**

• Data recording

– Using media

• Interviews

– Stakeholder interviews

– Subject Matter Expert interviews

– User and customer interviews

• Questionnaires

– Surveys, product reviews

• Literature review

– Studying existing systems

**Data Analysis**

• Requirement analysis

– Formal specifications of the system

• User analysis

– Identifying and understanding the user

• Task analysis

– Steps user take to accomplish this task

• Functional analysis

– Functions that system perform to help the users carry out their task.

**Prototyping**

**Advantages of Prototyping**:

• Users are actively involved in the development

• It provides a better system to users

• The users get a better understanding of the system being developed.

• Errors can be detected much earlier

• Quicker user feedback is available leading to better solutions

**Design**

• Goals

– Achieving goals

• Users and systems

– Understanding the raw materials: computer and human

• Limitations

– Accepting limitations of humans and of design

**Evaluation**

• Testing the usability, functionality and acceptability of an interactive system

• Expert evaluation

– Evaluation by Subject Matter Experts

• User evaluation

– Evaluation by user or customer

**HCI Principles Mobile Categories**

* Console
* Smartphones
* Tablets
* Cellphones

**Some Concepts**

The actual effectiveness of a system is achieved when there is an appropriate balance between functionality and usability.

**Functionality**set of actions and or services that a system offers to users.

**Usability**Varying and degree by which the system can be used efficiently and properly performing goals for certain users.

**Top 10 Principles**

1. Don’t Miniaturize

2. Context

3. Integrity Aesthetics

4. Consistency

5. Multitouch

6. Feedback

7. Metaphors

8. Rapid Selections

9. User Control

10. Minimize the Pain

* + **Don’t Miniaturize** – The size of the target of the System
  + **Context**– People context activities, who you are, who makes it possible, what you do, where you do, why you do, your needs and wishes.
  + **Integrity Aesthetics** – Main Function apparent, short terms.
  + **Consistency**– Internal consistency of elements or flows within a single system. External inconsistency elements or flows between two systems.
  + **Multitouch** – Direct Manipulation.
  + **Feedback**– Explain the problem.
  + **Metaphors**– Real world references.
  + **Rapid Selections** – Make user actions easier. Make Shortcuts to main tasks, avoid extends forms, store recent activities.
  + **User Control** – User must control actions.
  + **Minimize the Pain** – Calm the u

Concerns.

1. How can an interactive system be developed to ensure its usability?

2. How can the usability of an interactive system be demonstrated or measured?

**What are Paradigms?**

• Predominant theoretical frameworks or scientific world views

– e.g., Aristotelian, Newtonian, Einsteinian (relativistic) paradigms in physics

• Understanding HCI history is largely about understanding a series of paradigm shifts

– Not all listed here are necessarily “paradigm” shifts, but are at least candidates

– History will judge which are true shifts

**Paradigms of interaction**

 New computing technologies arrive, creating a new perception of the human computer relationship. We can trace some of these shifts in the history of interactive technologies.

**The Initial Paradigm.**

* Example Paradigm Shifts
* Batch processing
* Timesharing
* Networking
* Graphical display
* Microprocessor
* WWW
* Ubiquitous Computing

**Time-sharing**

• 1940s and 1950s – explosive technological growth

• 1960s – need to channel the power

• J.C.R. Licklider at ARPA

• Single computer supporting multiple users

**Video Display Units**

• More suitable medium than paper

• 1962 – Sutherland's Sketchpad

• Computers for visualizing and manipulating data

• One person's contribution could drastically change the history of computing.

**Programming toolkits**

• Engelbart at Stanford Research Institute

• 1963 – augmenting man's intellect

• 1968 NLS/Augment system demonstration

• The right programming toolkit provides building blocks to producing complex interactive systems

**Personal computing**

• 1970s – Papert's LOGO language for simple graphics programming by children

• A system is more powerful as it becomes easier to user

• Future of computing in small, powerful machines dedicated to the individual

• Kay at Xerox PARC – the Dynabook as the ultimate personal computer

**Window systems and the WIMP interface.**

• humans can pursue more than one task at a time

• windows used for dialogue partitioning, to “change the topic”

• 1981 – Xerox Star first commercial windowing system

• Windows, icons, menus and pointers now familiar interaction mechanisms.

**Metaphor**

• Relating computing to other real-world activity is effective teaching technique

– LOGO's turtle dragging its tail

– File management on an office desktop.

– Word processing as typing.

– Financial analysis on spreadsheets.

– Virtual reality – user inside the metaphor.

• Problems

– Some tasks do not fit into a given metaphor.

– Cultural bias

**Introduction to Models and Theories in HCI**

* In Carroll’s book, 'HCI Models, Theories and Frameworks', he states that HCI “is concerned with understanding how people make use of devices and systems that incorporate or embed computation, and how such devices and systems can be more useful and more usable.” Carroll goes on to talk about what HCI professionals do “analyze and design user interfaces and new user-interface technologies”, “created software tools and development environment to facilitate the construction of graphical user interfaces”, “pioneered the user of voice and video in user interfaces, hypertext links, interactive tutorials and context-sensitive help systems.”
* Carroll also goes on to talk about some of the work in mobile computing, information visualization for digital libraries and navigation techniques for virtual environments. Since this book was published in 2003, I think that this section is even more important today than ever. According to Smart Insights the number of mobile uses has increased dramatically and surpassed desktop users. This is important as while the concepts in the book are a solid starting point, they must be adapted to anything in use with mobile since it has a different style of working.
* Another concept Carroll talks about is scientific fragmentation and the current challenges. Carroll states that in the 1980s it was reasonable to expect HCI professionals to have a fairly comprehensive understanding of the concepts and methods in use, but in today’s world it is far more challenging for individuals to attain that breadth of knowledge since there are so many theories, methods, application domains and systems. I think this is spot on because as HCI has grown over the past decade there have only been more and more models and theories come up, many of them having to do with mobile. This makes it harder to know all of them and we start to run into individualization where researchers and HCI professionals start to focus on just a couple concepts instead of all of them

**Interaction Design Basics**

• Design

What it is, interventions, goals, constraints

• The design process

What happens when

• Users

Who they are, what they are like …

• Scenarios

Rich stories of design

• Navigation

Finding your way around a system

• Iteration and Prototypes

Never get it right first time!

**Interactions and Interventions**

Design interactions not just interfaces not just the immediate interaction

e.g. stapler in office – technology changes interaction style

• Manual: write, print, staple, write, print, staple,

• Electric: write, print, write, print, staple

• Designing interventions not just artefacts Not just the system, but also

• Documentation, manuals, tutorials

• What we say and do as well as what we make.

**What is design?**

Achieving goals within Constraints.

**• Goals purpose**

– Who is it for, why do they want it.

**• Constraints**

– Materials, Platforms

**Golden rule of Design**

Understand your Materials.

**For Human Computer Interaction**

* **Understand computers**

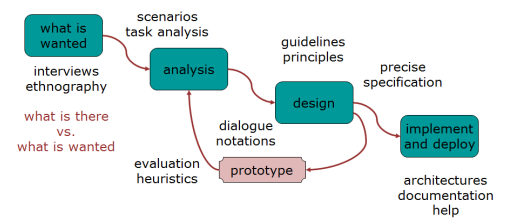
– Limitations, capacities, tools, platforms.

* **Understand people**

– Psychological, social aspects

– Human error.

**The Process of Design**



* **Requirements**

– What is there and what is wanted.

* **Analysis**

– Ordering and understanding.

* **Design**

– What to do and how to decide.

* **Iteration and Prototyping**

– Getting it right and finding what is really needed.

* **Implementation and Deployment**

 – Making it and getting it out there

**HCI in the software process**

* Software engineering and the design process for interactive systems
* Usability engineering
* Iterative design and prototyping
* Design rationale

**The software lifecycle**

* Software engineering is the discipline for understanding the software design process, or life cycle.
* Designing for usability occurs at all stages of the life cycle, not as a single isolated activity.

**The waterfall model**

**Activities in the life cycle**

* **Requirements specification**

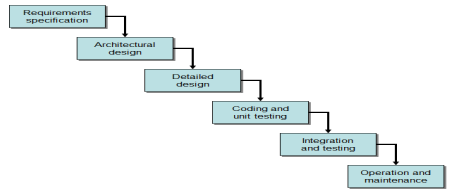
Designer and customer try capture what the system is expected to provide can be expressed in natural language or more precise languages, such as a task analysis would provide.

* **Architectural design**

High-level description of how the system will provide the services required factor system into major components of the system and how they are interrelated needs to satisfy both functional and non-functional requirements.

* **Detailed design**

Refinement of architectural components and interrelations to identify modules to be implemented separately the refinement is governed by the non-functional requirements.



**Verification and Validation**

* Verification

Designing the product right.

* Validation

Designing the right product.

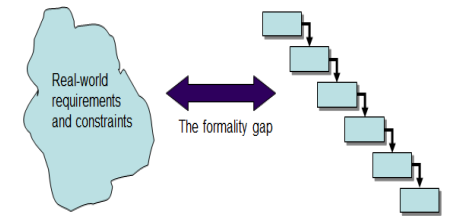
* The formality gap

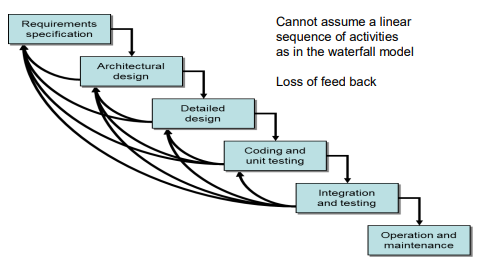
Validation will always rely to some extent on subjective means of proof

* Management and contractual issues

Design in commercial and legal contexts

**The life cycle for interactive systems**





**Usability engineering**

The ultimate test of usability based on measurement of user experience Usability engineering demands that specific usability measures be made explicit as requirements.

Usability Specification.

* Usability attribute/principle
* Measuring concept
* Measuring method
* Now level/ worst case/ planned level/ best case

Problems

* Usability specification requires level of detail that may not be
* Possible early in design satisfying a usability specification
* Does not necessarily satisfy usability

**Parts of a usability specification for a VCR**

**Attribute:**Backward recoverability

**Measuring concept:**Undo an erroneous programming sequence

**Measuring method:** Number of explicit user actions to undo current program

**Now level:** No current product allows such an undo

**Worst case:**As many actions as it takes to program-in mistake

**Planned level:** A maximum of two explicit user actions

**Best case:**One explicit cancel action

**Some metrics from ISO 9241**

Adopts traditional usability categories:

* effectiveness

– Can you achieve what you want to?

* efficiency

– Can you do it without wasting effort?

* satisfaction

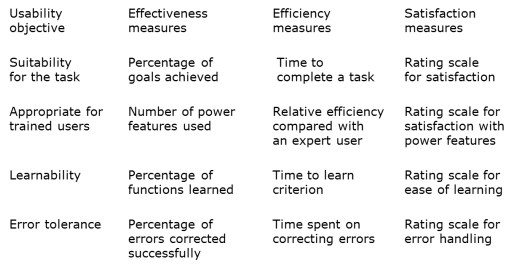
– Do you enjoy the process?

**Iterative design and prototyping**

* Iterative design overcomes inherent problems of incomplete requirements
* Prototypes
* Simulate or animate some features of intended system

Different types of prototypes

* + Throw-away
  + Incremental
  + Evolutionary
* Management issues
* time
* planning
* non-functional features contracts



**Techniques for prototyping**

• Storyboards need not be computer-based can be animated.

• Limited functionality simulations some part of system functionality provided by designers tools like HyperCard are common for these Wizard of Oz technique.

• Warning about iterative design inertia early bad decisions stay bad Diagnosing real usability problems in prototypes and not just the symptoms.

**Design rationale**

Design rationale is information that explains why a computer system is the way it is.

**Benefits of design rationale**

– Communication throughout life cycle

– Reuse of design knowledge across products

– Enforces design discipline

– Presents arguments for design trade-offs

– Organizes potentially large design space

– Capturing contextual information

**Types of DR:**

• Process-oriented

– preserves order of deliberation and decision-making

• Structure-oriented

– emphasizes post hoc structuring of considered design alternatives

• Two examples:

– Issue-based information system (IBIS)

– Design space analysis

**Issue-based information system (IBIS)**

• basis for much of design rationale research

• process-oriented

• main elements:

**Issues**

– Hierarchical structure with one ‘root’ issue.

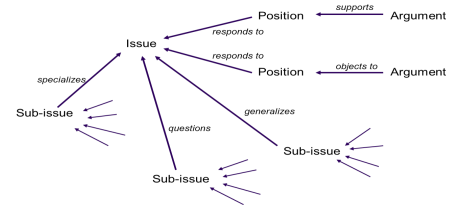
**Positions**

– Potential resolutions of an issue

**Arguments**

– modify the relationship between positions and issues

**Structure of Gibis**



**Psychological design rationale**

• To support task-artefact cycle in which user tasks are affected by the systems they use

• Aims to make explicit consequences of design for users

• Designers identify tasks system will support

• Scenarios are suggested to test task

• Users are observed on system

• Psychological claims of system made explicit

• Negative aspects of design can be used to improve next iteration of design