MONASH University
Information Technology

Unit IMS5302

Lecture 3
Design Approaches and Methods

Lecture outline

This lecture will cover:

- Organisation of knowledge
- Gulf of execution and evaluation
- Design principles
- Methodologies for developing effective systems from HCI perspective

Recap Week 1

Covered range of theories and how they impact or influence design.

- Affordance and visibility
- Mental and cognitive models
- Knowledge in the world and knowledge in the head
- Cognitive aids and natural mappings

- Norman's three design principles:
  - *Visceral design: appearance*
  - *Behavioural design: the pleasure and effectiveness of use*
  - *Reflective design: self-image, personal satisfaction, memories*

Gulfs of execution and evaluation (Norman)

Execution
- How well the system allows the user to do the intended action directly without extra effort

Evaluation
- Amount of effort user must exert to interpret physical state of system and determine how well expectations and intentions have been met.

Gulf of execution

- Does the system provide actions that correspond to the intentions of the person?
- Difference between intentions and allowable actions is the gulf of execution
- Measure is how well the system allows a person to do what they wanted to do without extra effort.

Gulf of evaluation

- Does the system provide a physical representation that can be directly perceived, directly interpretable in terms of person's intentions and expectations?
- Reflects the amount of effort person must exert to interpret the state of the system and work out how well what they expected to happen has happened.
- Gulf is small when system provides information about its state in a way that is easy to interpret and matches the way the user thinks

Putting the theories into practice

- How then can the theories be translated into development methods?
  - Affordance and visibility
  - Mental and cognitive models
  - Norman's three design principles
  - Knowledge in the world and knowledge in the head
  - Gulfs of evaluation and execution

Objects and actions

- Objects and actions can be described at high and low levels.
- User may understand high- level concept and refined lower- level concept.
- Interface actions can also be decomposed to lower- level actions.
- Object action model helps understand multiple complex processes that occur for users when trying to complete the task.

Steps/factors in HCI

- Preece suggests considering these factors for example:
  - Users (motivation, experience , cognitive capability)
  - Customers/clients
  - User interface
  - Work activity
  - Organisation
  - Comfort
  - Productivity

Shneiderman

- Suggests consideration must be given to:
  - Proper functionality (task analysis)– what tasks to be carried out, frequency of tasks
  - Reliability, availability, security and data integrity
  - Standardisation, integration, consistency and portability
  - Schedules and budgets

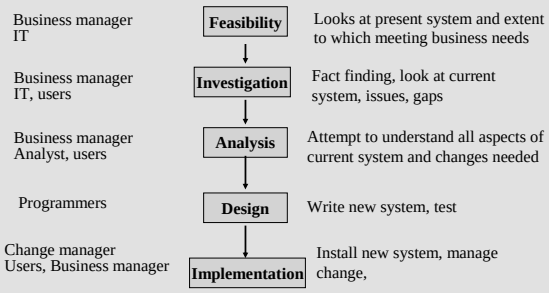
Norman (1990) Seven stages of action

1. Forming the goal
2. Forming the intention
3. Specifying the action
4. Executing the action
5. Perceiving the system state
6. Interpreting system state
7. Evaluating the outcome. (Norman)

How easily can one?

1. Determine the function of the device?
2. Tell what actions are possible?
3. Determine mapping from intention to physical movement?
4. Perform the action?
4. Tell if system is in desired state?
5. Determine mapping from system state to interpretation?
6. Tell what state the system is in?

Standard Systems Development Life Cycle



SDLC

- Oldest method, still most common. Weaknesses include:
  - Often does not meet management needs
  - Unambitious systems design
  - User dissatisfaction
  - Problems with documentation
  - Lack of control
  - Incomplete systems
  - Harder to maintain

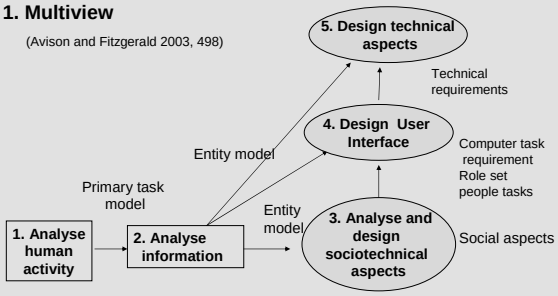
Weaknesses cont..

- Use of a development method is important to keep projects on time and on budget.
- Methodologies often effective in facilitating the development process.
- Most common methodologies however do not provide clear guidance on and processes for studying users, understanding their needs and creating appropriate, effective interfaces.
- User centred design only part of process, must fit with the whole development approach.
- S&P pg 118 cites example of IBM's approach

User centred development methodologies

1. Multiview

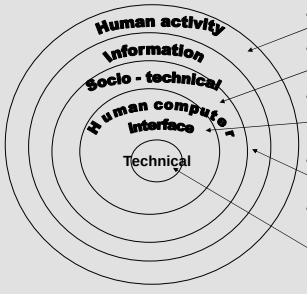
(Avison and Fitzgerald 2003, 498)



Multiview principles

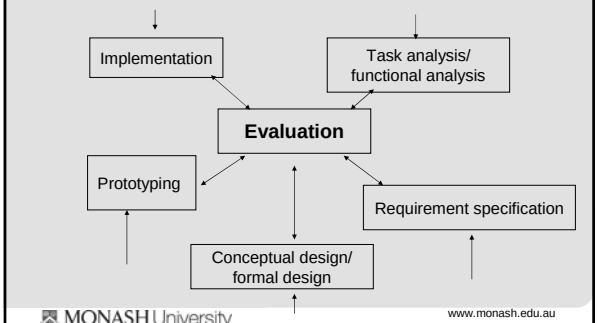
- System described in terms of purpose of the system, stakeholders involved, perspective of the system owners.
- Strong socio-technical approach
- Functional model (stage 3) describes people's tasks, basis of task allocation.
- Only once these are established are the technical requirements specified for the designers.
- Methodology uses entity relationship modelling and data flow modelling to develop a conceptual model before physical design takes place

Framework - multiview

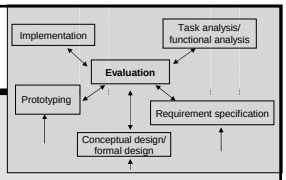


- How will the computer system further the organisation's aims?
- How can the system fit into the working lives of the people in the organisation that will use it?
- How can the individuals concerned best relate to the system in terms of operating and using it?
- What information system processing function is it designed to perform?
- What are the technical specifications of system? (must come close enough to doing the things that have been written down in the answers to the other four questions) (499)

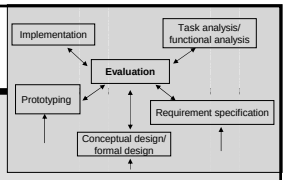
Star model (Preece 381)



The star life cycle



- Was derived from actual design practices among HCI designers.
- Takes the idea of prototyping and evaluation further than other approaches.
- SDLC works from a system view to a user's view, notion of top down, formal.
- Star suggests moving from user's view out to the system.



- Evaluation is central. All aspects of the development process are subject to constant evaluation by users and experts.
- Stresses rapid prototyping an incremental development rather than top-down or analytic approach.
- Development can start at any stage, this is quite common.
- Requirements, design and product gradually evolve. (Preece et al 380-381)

- Star model stresses the difference between:
  - conceptual design (what is required, what system should do, what data rest required, what users need to know) design
  - physical design (how can things be achieved?).
- Distinction between the two important for good systems because defers decisions as to who or what ultimately performs which functions, or provide which data until late in design process.

LUCID (Shneiderman 119 122)

- Logical User-Centred Interactive Design Methodology
- Well-respected and well used methodology.
- Involves six stages:
  1. Envision: align with organisational strategy
  2. Discovery; identify high-level user requirements
  3. Design foundation: develop conceptual design, usability test and refine
  4. Design detail: high-level design converted to specifications
  5. Build
  6. Release

  (more detailed description can be found on page 120)

- Key focus is on screen prototypes that illustrate major navigational paths through the system.
- Allows users to evaluate and refine early in the development process.
- Uses rapid prototyping and iterative usability testing.
- Methodology makes a commitment to use the centre design and highlights the role of a usability engineer.

ETHICS (Avison and Fitzgerald 2003)

- Effective, Technical and Human Implementation of Computer Based Systems.
- Socio technical view of systems, ie to be effective system must fit closely with the social or organisational factors.
- Suggests development is not a technical issue but an organisational one concerned with change.
- Participation key to the method

- Mumford (1983) defines socio technical approach as:

"*one which recognises the interaction of technology in people and produces work systems which are both technically efficient and have social characteristics which lead to high job satisfaction*." (Avison and Fitzgerald pp 449)

- Job satisfaction is where there is a good fit between or the employee wants from his/her work and what the organisation wants from her.

ETHICS – 15 steps

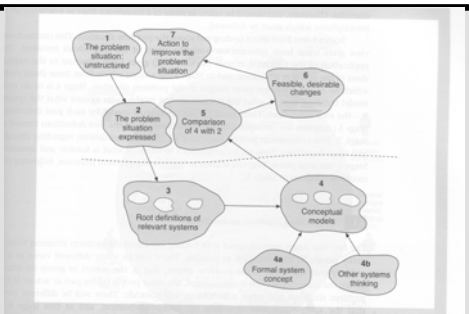
1. Identify why there is a need for change
2. Identify the boundaries of the new system
3. Describes the existing system
4. Define the key objectives and tasks for the development process (includes steps 5 and 6)
7. Identify the deficiencies in the current system
8. Identify the needs of the employee is in terms of job satisfaction and the new system
9. Identified future needs

10. Specify and weight efficiency and job satisfaction needs and objectives.
11. Organisational design of the new system
12. Identification of technical options in line preparation of the details were could design
13. Implementation
14. Evaluation

Soft Systems Methodology

- Developed by Checkland (1981)
- Suggests "systems analysts apply their craft" to problems which are not well or clearly defined.
- Organisations are complex and so are system problems.
- Assumes systems development is a complex problem situation and system solution more likely to be addressed using this methodology than more simplistic structural data oriented approach.
- Acknowledges the importance of people in organisations.

SSM (Avison and Fitzgerald pg 471)

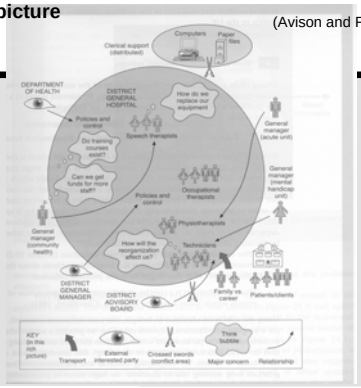


Stages in SSM

1. Identify problem situation: finding out what the problem is from as many stakeholders as possible.
2. Expressing the problem situation. Can involve of drawing rich pictures.
3. Root definitions
4. Building conceptual models: usually involves a diagram of activities showing what the system, described the route definition will do.
5. Compare model with reality. 6. Assess changes
7. Action to improve situation.

Rich picture

(Avison and Fitzgerald pg 473)



Summary

- Need to understand gaps in users' understanding, (gulf of execution and gulf of evaluation)
- HCI design methods seek to involve users into all aspects of systems development. Not the case with SDLC.
- Variety of different methods have evolved over time.
- Multiview is a user centred methodology whereas star lifecycle reflects what designers do and has a high focus on the evaluation.
- Design of human computer systems needs to recognise organisational as well as local needs focus on the whole work situation not on perceived problem.

- Designing software is complex knowledge intensive, design support is needed.
- Design support is needed both at individual task level and overall process level

References

- Dumas J, Redish J (1994) *A practical guide to usability testing*, Ablex publishing
- Norman D. (1990) *The design of Everyday Things*, Doubleday
- Shneiderman B. (2005) *Designing the User Interface,* Addison Wesley
- Preece et al (2002) *Interaction Design*, Wiley
- Avison and Fitzgerald (2003) *Information Systems Development Methodologies,* Techniques and Tools, McGraw Hill