Lecture outline

This lecture will cover:
• Design principles – key issues for practice
  – Other development methods from last week
  – Using user centred design approach
  – Learning about users and their tasks
  – Using prototyping to improve design
  – Mapping tasks to system functionality
  – How to ensure effective user participation
• Other design considerations

LUCID (Shneiderman 119)

• 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, i.e., 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 what 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. Describe the existing system
4. Define the key objectives and tasks for the development process (includes steps 5 and 6)
5. Identify the deficiencies in the current system
6. Identify the needs of the employee in terms of job satisfaction and the new system
7. Identify 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 we 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.
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 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.

User centred design approach

- Preece et al (2002) define a user centred approach to development as “real users and their goals, not just technology, should be the driving force behind the development of a product.” (285)
- Another definition: “the active involvement of users for a clear understanding of user and task requirements, iterative design and evaluation, and a multidisciplinary approach.” (Mao et al 2005).
- User-centred design approach results in systems that are easier to learn, perform faster, reduce user errors and encourage users to explore more of the software.
- Better helps help organisations align systems functionality with organisational/ business needs. (Shneiderman, 2005, 118).

Principles of UCD

1. Focus early on users and their tasks and focus throughout the process.
   - Users and their goals drive development.
   - Study user behaviour and context of use and design the system to support this.
   - Methodologies should seek to find out as much about users and their tasks as possible.
   - Capture and design for user characteristics.
   - Design decisions should be taken within the context of users and their work environment.
Principles of user centred design

2. Consult users throughout development and take their input seriously.
   – High levels of evaluation need to take place.
   – Strong focus on user participation throughout the development process.
3. Process should be very iterative

“Current users are not dedicated to the technology; their background is more tied to their work needs, while their use of computers for entertainment is discretionary. Design should be based on careful observation of current users, refined by thoughtful analysis of task frequencies and sequences, and validated through early usability and thorough acceptance tests.” (S&P 110)

User centred design in practice – 1. Focus on users

Observe users – learn about their work habits.
Ethnographic Observation
• Involves observing users in the work place
• Used to identify behaviour in the organisational context.
• Easy to misinterpret observations, disrupt work, overlook some aspects/information.
• Data collection can be qualitative (impressions, notes) or quantitative (ratings on scales eg no of errors made). (S&P 122-125)
Principles

- Users’ behaviour and tasks more than just finding out what users do. Need to understand behaviour, preferences etc.
- Understand cognitive aspects of users, attention, memory, perception. Physical aspects.
- Designers remain aware of users.

User Task Analysis

- User and task Analysis
  - Identify group of potential users
  - Create user/task matrix and user/characteristic matrix to serve as a model for your user community.
  - Test your assumptions
  - Understand work environment – tools, tasks etc
  - Conduct work flow analysis
  - Develop task hierarchy

“To get something done, you have to start with some notion of what is wanted – the goal that is to be achieved” (Norman)

Task matrix - Example

<table>
<thead>
<tr>
<th>User Types</th>
<th>Information search (hotel)</th>
<th>Inquiry room availability</th>
<th>Book a room</th>
<th>Check bookings</th>
<th>Management reports</th>
</tr>
</thead>
<tbody>
<tr>
<td>Young person short visit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Older experience business traveler</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tourist</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Older person less familiar with technology</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reservations clerk</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manager</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Developing a task matrix

| • Identify different user types according to: |
| – Experience both with the task and technology |
| – Role in the organisation |
| – Type of task |
| • Identify type of tasks – assumes system used by more than one type of user. |
| • Indicate on the matrix if that task likely to be performed by that user type. |
| • Provides picture of what tasks performed by which users |

---

### 2. Involve, consult, evaluate with users

Involve the users meaningfully.

- Involving users in the design process can be costly and can take longer. Generally positive.
- Variety of methods can be used:
  - PICTIVE (Plastic Interface for Collaborative Technology Initiatives through Video Exploration)
  - CARD (Collaborative Analysis of Requirements and Design)
  - Scenario development

---

### PICTIVE

- Preece et al (2002) describe this technique as using "low fidelity office items, such as sticky notes and pens, and a collection of design objects to investigate specific screen and window layouts for a system." (Preece et al 2002 p. 307)
- Usually a PICTIVE session is video recorded,
- Participants are asked to interact with the system Using aids brought to the session
**CARD**

- Similar to PICTIVE but uses playing cards with pictures of computers and screen dumps to explore workflow options.
- During a workshop session the cards are moved around by the participants in order to show workflow between computer screens and tasks.
- Can be used in conjunction with PICTIVE because it provides a different perspective on the workflow issue.

**Issues**

- For participation to be effective careful selection of users needs to be made.
- Users will be required to commit quite a lot of time to the process, senior management have to be supportive.
- Must ensure all users have an opportunity to contribute meaningfully.
- Participation is more than just getting users sign off on stages of the development process.

**3. Be iterative**

- Prototyping helps with iterations, provides users with view of the system
- Software prototype is a system that:
  - “actually works
  - will not have a generalised lifetime
  - may serve many different purposes
  - must be built quickly and cheaply
  - is an integral part of interactive user-centred design in which evaluation and subsequent modification of the design are fundamental concepts.” (Preece, et al 1996 p538)
Using prototyping to improve design

- Rapid prototyping for interface development for new products is a well-established design technique.
- Prototyping can be an effective medium of communication between end user and designer, and provides a concrete forum for exchange which has benefits over more abstract discussion.
- Requirements emerge over time, based on experience in using products.

Value of prototypes

- Help designers make decisions by obtaining information from users on:
  - Necessary functionality
  - Operation sequences
  - User support needs
  - Required representations
  - Interface look and feel
- Gives both designers and users a hands-on feel for the product.
- Prototypes can help support design by helping refine user needs.

Static, paper-based prototypes

- Sometimes called a screenplay or storyboard.
- Potential users shown paper-based screen images of the product.
- Users asked to show how they would interact with the system.
- Useful for getting users involved in design early as they can sit beside designers and redesign.
- Suited to very early stage of design.
- Can help identify usability problems with, for example, menus.
Interactive software based prototypes

- Different design concepts can be explored
- Possible to simulate look and feel of software user interface in a short time using prototyping tools or other environments such as HTML
- Allow designers to make changes before too late
- Can be used to evaluate several iterations of design
- Prototypes can also be useful for marketing

Building a prototype

- Created to explore part of the interface, not full interface. Depends on primary concerns, these could be:
  - Flow of screens for major tasks
  - Overall metaphor and how it will be carried out
  - Screen layout and basic task screens
  - Alternative design ideas
  - Menus and menu hierarchies
- Should cover common tasks but restricted interface and functionality.

Benefits of computer-based prototypes

- Users work with prototype directly. Value is user can see product’s look and feel early.
- Enables designers to try out design concepts quickly.
- Can be inexpensive depending on method used.
- Can be used by marketing and sales as product demonstrators.
- Can give designers an opportunity to test ideas and alternatives before building product.
Drawbacks of computer-based prototypes

- If used to create entire interface it becomes very demanding in terms of time.
- Limitations also exist in software prototyping tools, difficulty in simulating user response times in real product.
- May raise unrealistic customer expectations as to how soon product is available.

Steps/factors in development

- Preece suggests considering these factors in design:
  - 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
Issue:

- Development teams frequently focus more on technology than on users when designing systems.
- Result: systems that do not satisfy users’ needs.
- Teams: dominated by technical people, people with human factors skills still rarely included.

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.
- Multiview offers a more prescriptive, it uses scented 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

- What contribution did this paper make to the topic (systems development)?
- What contribution did the paper make to understanding conceptual versus physical models?
- What was the contribution to the wider business world and practice particularly systems design/development?

References

- Shneiderman and Plaisant. Ch 3
- Dumas J, Redish J (1994) A practical guide to usability testing, Ablex publishing