THE PROCESS OF INTERACTION DESIGN

Overview, Chapters 9-11
It is a process:
- a goal-directed problem solving activity informed by intended use, target domain, materials, cost, and feasibility
- a creative activity
- a decision-making activity to balance trade-offs

It is a representation:
- a plan for development
- a set of alternatives and successive elaborations
WHAT IS A USER-CENTERED APPROACH?

User-centered approach is based on:

- Early focus on users and tasks: directly studying cognitive, behavioural, anthropomorphic & attitudinal characteristics
- Empirical measurement: users’ reactions and performance to scenarios, manuals, simulations & prototypes are observed, recorded and analysed
- Iterative design: when problems are found in user testing, fix them and carry out more tests

Participatory design approach is based on user participation in the design process
SOME PRACTICAL ISSUES

• Who are the users?

• What are ‘needs’?

• Where do alternatives come from?

• How do you choose among alternatives?
FOUR BASIC ACTIVITIES

There are four basic activities in Interaction Design:

– 1. Identifying needs and establishing requirements
– 2. Developing alternative designs
– 3. Building interactive versions of the designs
– 4. Evaluating designs

OpenIDEO http://openideo.com
WHO ARE THE USERS/STAKEHOLDERS?

Not as obvious as you think:
– those who interact directly with the product
– those who manage direct users
– those who receive output from the product
– those who make the purchasing decision
– those who use competitor’s products

Three categories of user (Eason, 1987):
– primary: frequent hands-on
– secondary: occasional or via someone else
– tertiary: affected by its introduction, or will influence its purchase
STAKEHOLDERS
WHAT ARE THE USERS’ CAPABILITIES?

Humans vary in many dimensions:

— size of hands may affect the size and positioning of input buttons
— motor abilities may affect the suitability of certain input and output devices
— height if designing a physical kiosk
— strength - a child’s toy requires little strength to operate, but greater strength to change batteries
— disabilities (e.g. sight, hearing, dexterity)
WHERE DO ALTERNATIVES COME FROM?

• Humans stick to what they know works
• But considering alternatives is important to ‘break out of the box’
• Designers are trained to consider alternatives, software people generally are not
• How do you generate alternatives?
  —‘Flair and creativity’: research and synthesis
  —Seek inspiration: look at similar products or look at very different products
HOW DO YOU CHOOSE AMONG ALTERNATIVES?

- Evaluation with users or with peers, e.g. prototypes
- Technical feasibility: some not possible
- Quality thresholds: Usability goals lead to usability criteria set early on and check regularly
  - safety: how safe?
  - utility: which functions are superfluous?
  - effectiveness: appropriate support? task coverage, information available
  - efficiency: performance measurements

Visibility of system status
Match between system and the real world
User control and freedom
Consistency and standards
Help users recognize, diagnose and recover from errors
Error prevention
Recognition rather than recall
Flexibility and efficiency of use
Aesthetic and minimalist design
Help and documentation
TESTING PROTOTYPES TO CHOOSE AMONG ALTERNATIVES
LIFECYCLE MODELS
Are management tools showing how activities are related
DSDM LIFECYCLE MODEL
THE STAR MODEL (HARTSON AND HIX, 1989)
IDENTIFYING NEEDS AND ESTABLISHING REQUIREMENTS

- Users rarely know what is possible
- Users can’t tell you what they ‘need’ to help them achieve their goals
- Instead, look at existing tasks:
  - their context
  - what information do they require?
  - who collaborates to achieve the task?
  - why is the task achieved the way it is?
- Envisioned tasks:
  - can be rooted in existing behaviour
  - can be described as future scenarios
WHAT, HOW AND WHY REQUIREMENTS?

What
Two aims:
1. Understand as much as possible about users, task, context
2. Produce a stable set of requirements

How
Data gathering activities
Data analysis activities
Expression as ‘requirements’
All of this is iterative
ESTABLISHING REQUIREMENTS

• What do users want? What do users ‘need’?
  Requirements need clarification, refinement, completion, re-scoping
  Input: requirements document (maybe)
  Output: stable requirements

• Why ‘establish’?
  Requirements arise from understanding users’ needs
  Requirements can be justified & related to data
DIFFERENT KINDS OF REQUIREMENTS

• Functional:
  — What the system should do
  — Historically the main focus of requirements activities

• (Non-functional: memory size, response time...)

• Data:
  — What kinds of data need to be stored?
  — How will they be stored (e.g. database)?
DIFFERENT KINDS OF REQUIREMENTS

Environment or context of use:

— physical: dusty? noisy? vibration? light? heat? humidity? .... (e.g. OMS insects, ATM)
— social: sharing of files, of displays, in paper, across great distances, work individually, privacy for clients
— organisational: hierarchy, IT department’s attitude and remit, user support, communications structure and infrastructure, availability of training
DIFFERENT KINDS OF REQUIREMENTS

- Users: Who are they?
  - Characteristics: ability, background, attitude to computers
  - System use: novice, expert, casual, frequent
    - Novice: step-by-step (prompted), constrained, clear information
    - Expert: flexibility, access/power
    - Frequent: short cuts
    - Casual/infrequent: clear instructions, e.g. menu paths
PERSONAS

• Capture user characteristics
• Not real people, but synthesised from real user characteristics
• Should not be idealised
• Bring them to life with a name, characteristics, goals, personal background
• Develop multiple personas

http://www.masternewmedia.org/interface_and_navigation_design/usability/how-to-create-effective-personas-20071004.htm
DATA GATHERING FOR REQUIREMENTS

Questionnaires:

— Often used in conjunction with other techniques
— Can give quantitative or qualitative data
— Good for answering specific questions from a large, dispersed group of people

Researching similar products:

— Good for prompting requirements
DATA GATHERING FOR REQUIREMENTS

Direct observation:
— Gain insights into stakeholders’ tasks
— Good for understanding the nature and context of the tasks
— But, it requires time and commitment from a member of the design team, and it can result in a huge amount of data

Indirect observation:
— Not often used in requirements activity
— Good for logging current tasks
DATA GATHERING FOR REQUIREMENTS

Studying documentation:
— Procedures and rules are often written down in manuals
— Good source of data about the steps involved in an activity, and any regulations governing a task
— Not to be used in isolation
— Good for understanding legislation, and getting background information
— No stakeholder time, which is a limiting factor on the other techniques
CONTEXTUAL INQUIRY

• An approach to ethnographic study where user is expert, designer is apprentice
• A form of interview, but
  — at users’ workplace (workstation)
  — 2 to 3 hours long
• Four main principles:
  — Context: see workplace & what happens
  — Partnership: user and developer collaborate
  — Interpretation: observations interpreted by user and developer together
  — Focus: project focus to understand what to look for
DATA INTERPRETATION AND ANALYSIS

• Start soon after data gathering session

• Initial interpretation before deeper analysis

• Different approaches emphasize different elements e.g. class diagrams for object-oriented systems, entity-relationship diagrams for data intensive systems
TASK DESCRIPTIONS

• Scenarios
  — an informal narrative story, simple, ‘natural’, personal, not generalizable

• Use cases
  — assume interaction with a system
  — assume detailed understanding of the interaction

• Essential use cases
  — abstract away from the details
  — does not have the same assumptions as use cases
SCENARIO FOR WORK SITUATIONS

Axes:
- Where to do the work
-- who controls data

1. In "Disciples of the Cloud" the notion of the workplace remains central to work. Among the reasons for this are the need to maintain the culture of the organisation, and to control intellectual property.
2. In "Electronic Cottages" the 'workplace' has shrunk because companies are increasingly virtual. Work takes place either in the home or in more local hub offices. Employees have more control over when and where they work, but see value in being employed.
3. In "Replicants" organisations are highly distributed, physically and structurally. "Replicants" are 'fast followers' who thrive in a world of open innovation by using public knowledge (open source software, or expired patents) as the basis for service or process innovation.
4. In "Mutual Worlds" businesses operate as loose federations of independent contractors. Intellectual property is controlled by workers; service tends to be geographically located.
The Thomson family enjoy outdoor activity holidays and want to try their hand at sailing this year. There are four members of the family: Sky who is 10 years old, Eamonn who is 15 years old, Claire who is 35, and Will who is 40. While out on a shopping trip they call by at the travel agents in their local town to start exploring the possibilities ... The travel organizer is located in a quiet corner of the agents’ office, where there are comfortable seats and play things for young children. They all gather around the organizer and enter their initial set of requirements—a sailing holiday for four novices. The stand-alone console is designed so that all members of the family can interact easily and comfortably with it. The system’s initial suggestion is that they should consider a flotilla holiday, where several novice crews go sailing together and provide mutual support for first-time sailors...”
USE CASE FOR HOLIDAY PLANNER

1. The system displays options for investigating visa and vaccination requirements.
2. The user chooses the option to find out about visa requirements.
3. The system prompts user for the name of the destination country.
4. The user enters the country’s name.
5. The system checks that the country is valid.
6. The system prompts the user for her nationality.
7. The user enters her nationality.
8. The system checks the visa requirements of the entered country for a passport holder of her nationality.
9. The system displays the visa requirements.
10. The system displays the option to print out the visa requirements.
11. The user chooses to print the requirements.
**TASK ANALYSIS**

- Task descriptions are often used to envision new systems or devices
- Task analysis is used mainly to investigate an existing situation
- It is important not to focus on superficial activities
  - What are people trying to achieve?
  - Why are they trying to achieve it?
  - How are they going about it?
- Many techniques, the most popular is Hierarchical Task Analysis (HTA)
PROTOTYPING AND CONSTRUCTION

• What is a prototype?
• Why prototype?
• Different kinds of prototyping
  low fidelity
  high fidelity
• Compromises in prototyping
  vertical
  horizontal
• Construction
WHAT IS A PROTOTYPE?

In other design fields a prototype is a small-scale model:

• a miniature car
• a miniature building or town
WHAT IS A PROTOTYPE?

In interaction 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)
- a cardboard mock-up
- a piece of software with limited functionality written in the target language or in another language
WHY PROTOTYPE?

• Evaluation and feedback are central to interaction design
• Stakeholders can see, hold, interact with a prototype more easily than a document or a drawing
• Team members can communicate effectively
• You can test out ideas for yourself
• It encourages reflection: very important aspect of design
• Prototypes answer questions, and support designers in choosing between alternatives
WHAT TO PROTOTYPE?

• Technical issues

• Work flow, task design

• Screen layouts and information display

• Difficult, controversial, critical areas
LOW-FIDELITY PROTOTYPING

• Uses a medium which is unlike the final medium, e.g. paper, cardboard

• Is quick, cheap and easily changed

• Examples:
  sketches of screens, task sequences, etc
  ‘Post-it’ notes
  storyboards
  ‘Wizard-of-Oz’
STORYBOARDS

• Often used with scenarios, bringing more detail, and a chance to role play

• It is a series of sketches showing how a user might progress through a task using the device

• Used early in design
SKETCHING

- Sketching is important to low-fidelity prototyping
- Don’t be inhibited about drawing ability. Practice simple symbols

1. Drive car to gas pump
2. Take nozzle from pump...
3. ...and put it into the car’s gas tank
4. Squeeze trigger on the nozzle until tank is full
5. Replace nozzle when tank is full
6. Pay cashier
CARD-BASED PROTOTYPES

- Index cards (3 X 5 inches)
- Each card represents one screen or part of screen
- Often used in website development
‘WIZARD-OF-OZ’ PROTOTYPING

• The user thinks they are interacting with a computer, but a developer is responding to output rather than the system.
• Usually done early in design to understand users’ expectations
• What is ‘wrong’ with this approach?
HIGH-FIDELITY PROTOTYPING

- Uses materials that you would expect to be in the final product.
- Prototype looks more like the final system than a low-fidelity version.
- For a high-fidelity software prototype common environments include Macromedia Director, Visual Basic, and Smalltalk.
- Danger that users think they have a full system……..see compromises
COMPROMISES IN PROTOTYPING

• All prototypes involve compromises
• For software-based prototyping maybe there is a slow response? sketchy icons? limited functionality?
• Two common types of compromise
  • ‘horizontal’: provide a wide range of functions, but with little detail
  • ‘vertical’: provide a lot of detail for only a few functions
• Compromises in prototypes mustn’t be ignored. Product needs engineering
CONSTRUCTION

• Taking the prototypes (or learning from them) and creating a whole

• Quality must be attended to: usability (of course), reliability, robustness, maintainability, integrity, portability, efficiency, etc

• Product must be engineered

  Evolutionary prototyping

  ‘Throw-away’ prototyping