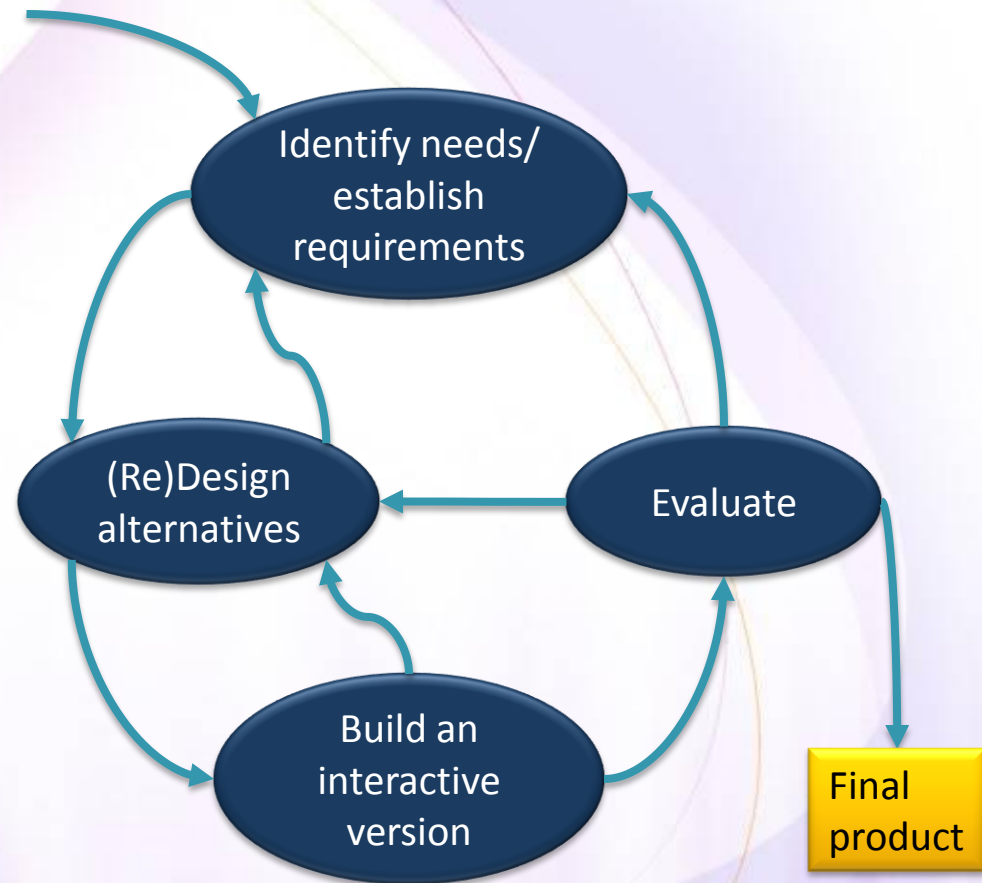


Design Engineering

Interaction design

- 4 basic activities
- 3 key characteristics:
 - Focus on users early in the design and evaluation of the artifact
 - Identify, document and agree specific usability and user experience goals
 - Iteration is inevitable. Designers never get it right first time



Design key points

- Interface design should be user-centered
- An interface should
 - Be logical and consistent
 - Help users recover from errors
- Interaction styles include
 - direct manipulation
 - menu systems form fill-in
 - command languages
 - natural language
- Graphical displays should be used to present trends and approximate values
- Digital displays when precision is required

Design key points

- Colour should be used sparingly and consistently
- Systems should provide on-line help. This should include
 - “help, I’m in trouble”
 - “help, I want information”
- Error messages should be positive rather than negative
- A range of different types of user documents should be provided
- Ideally, a user interface should be evaluated against a usability specification

Covered topics

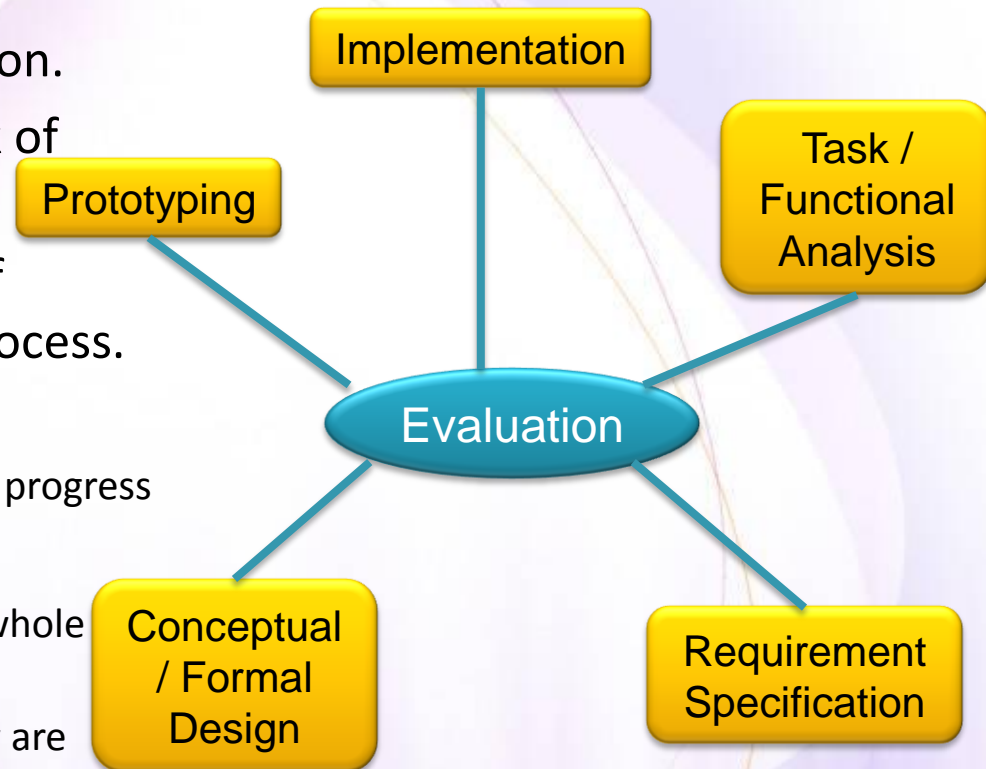
- Product Life Cycle
- User Centered Design
- Three Principles
- Product Development
- Interactive Design

Lifecycle models

- Show how activities are related to each other
- Are:
 - management tools
 - simplified versions of reality
- Many lifecycle models exist:
 - from software engineering perspective:
 - waterfall,
 - spiral,
 - Joint Application Design (JAD) / Rapid Application Design (RAD)
 - ...
 - from HCI perspective:
 - star
 - usability engineering
 -

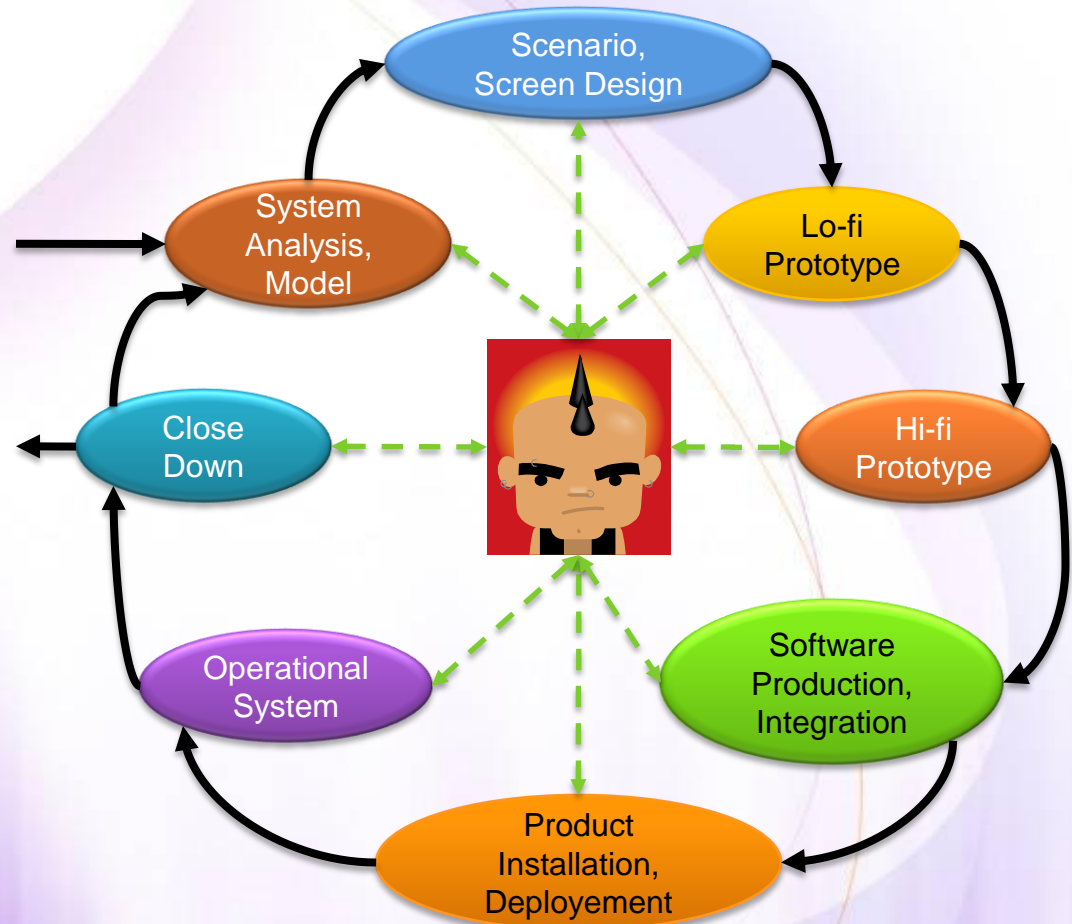
Star Life Cycle Design

- Is centered around evaluation
 - Does not specify any order to the activities surrounding this evaluation.
 - Derived from some empirical work of interface designers
 - Strongly encourages a high level of iteration throughout the design process.
- Disadvantage:
- Make it hard to manage time and track progress
- Advantage:
- Make user evaluation throughout the whole design process
 - Adapt to the needs of the users as they are discovered and as they (invariably) change



Usability engineering lifecycle model

- Holistic view of usability engineering
- Provides links to software engineering approaches, e.g. OOSE
- Stages of identifying requirements, designing, evaluating, prototyping
- Can be scaled down for small projects
- Uses a style guide to capture a set of usability goals



REQUIREMENTS ANALYSIS



Function/Data Modeling
OOSE: Requirements Model

Style Guide

LEVEL 1

Work Reengineering

Concept Model Design

CM Mockups

Iterative CM Evaluation

Eliminated Major Flaws?

NO

Start Application Architecture
OOSE: Analysis Model

Style Guide

YES

LEVEL 2

Screen Design Standards

SDS Prototyping

Iterative SDS Evaluation

Met Usability Goals?

NO

Start App. Design/Development
OOSE: Design Model/Imp. Model

Style Guide

YES

DESIGN/TESTING/DEVELOPMENT

LEVEL 3

Detailed UI Design

Unit/System Testing
OOSE: Test Model

Iterative DUID Evaluation

Met Usability Goals?

NO

YES

Style Guide

All Functionality Addressed?

NO

YES

INSTALLATION

Installation

User Feedback

All Issues Resolved?

NO

YES

DONE!

UE Task

T Development Task

Decision Point

Documentation

Complex Application

Simple Application

Covered topics

- Product Life Cycle
- User Centered Design
- Three Principles
- Product Development
- Interactive Design

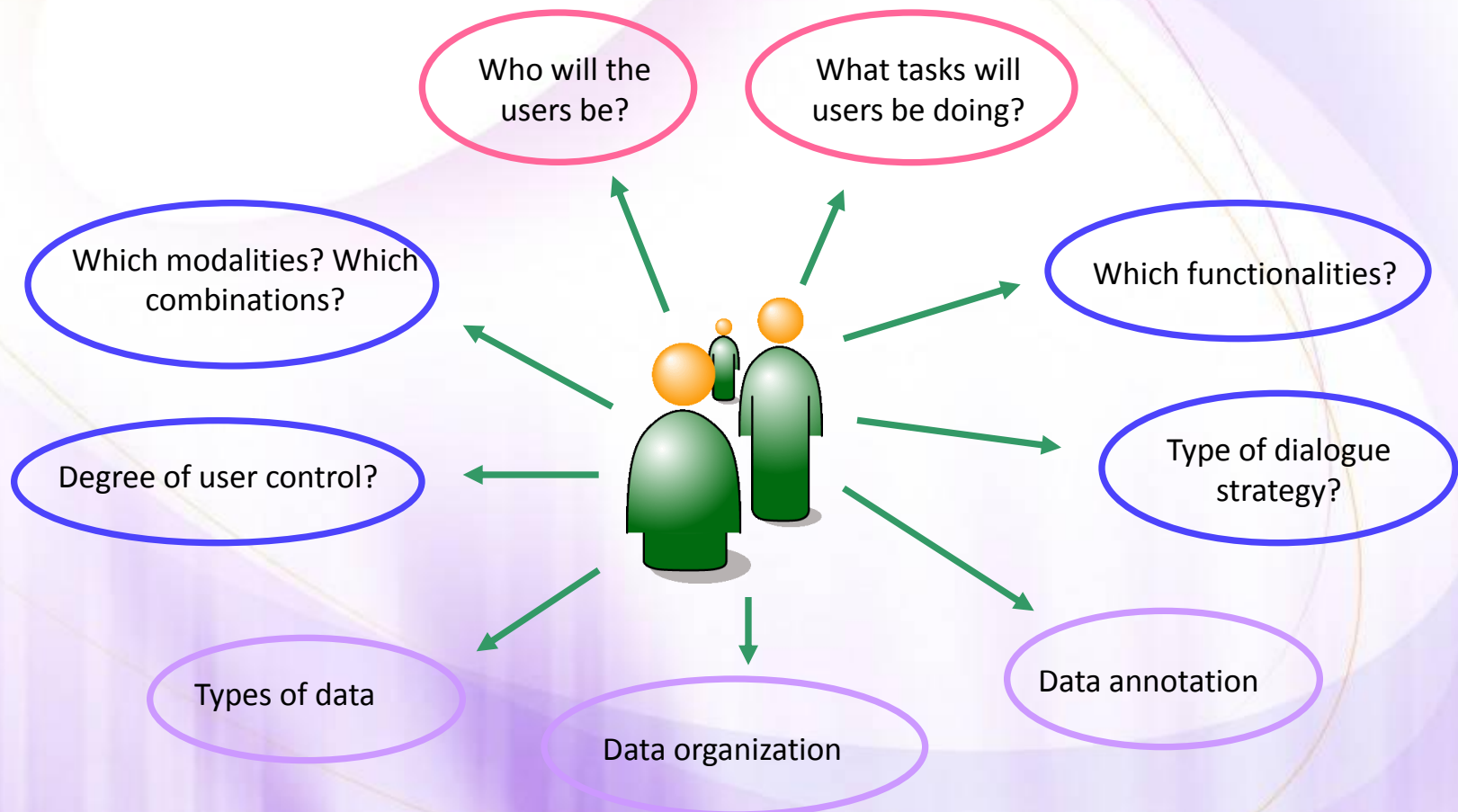
User centered design

- Idea : Bend and structure the functioning of a user interface around **how people can, want or need to work**
 - The needs, wants and limitations of the end user of an interface or document are given extensive attention at each stage of the design process
- UCD is characterized as a multi-stage problem solving process that requires designers to:
 - Analyze and foresee how users are likely to use an interface
 - Test the validity of their assumptions with regards to user behavior in real world tests with actual users.
 - what a first-time user of their design experiences
 - what each user's learning curve may look like

User centered design

“The bottom line is that the tool should be designed around the needs of the user to allow for effective and efficient use of the application.”

(Scanlon and Percival, IBM)



Some UCD approaches

- Cooperative design: involving designers and users on an equal footing. This is the Scandinavian tradition towards design of IT artefacts and it has been evolving since 1970.
- Participatory design : attempts to actively involve the end users in the design process to help ensure that the product designed meets their needs and is usable
- Contextual design (customer centered design) : some ideas from PD
- Rapid prototyping: automatic construction of physical objects with 3D printers

Covered topics

- Product Life Cycle
- User Centered Design
- Three Principles
- Product Development
- Interactive Design

Intuitive interface: three principles

- Context: the design of function, location and appearance of product and interface features.
- Objective: allow designers to draw on users past experience in order to develop products which facilitate intuitive interaction and ready acceptance of new technologies.

Intuitive interface: three principles

- Use familiar symbols and/or words for well-known functions
 - Put them in a familiar or expected position and make the function comparable with similar functions users have seen before.
 - Involve utilizing existing features, labels or icons that users have seen before in similar products that perform the same function.
- This is the simplest level of applying intuitive use.
- Example ???

Intuitive interface: three principles

- Make it obvious what less well-known functions will do by using familiar things as metaphors to demonstrate their function.
- Use of metaphor to make something completely new familiar by relating it to something already existing.

Example ???

Intuitive interface: three principles

- Increase consistency so that function, location and appearance of features are consistent between different parts of the design and throughout each part.
- Allows users to apply the same knowledge and metaphors across all parts of the interface.

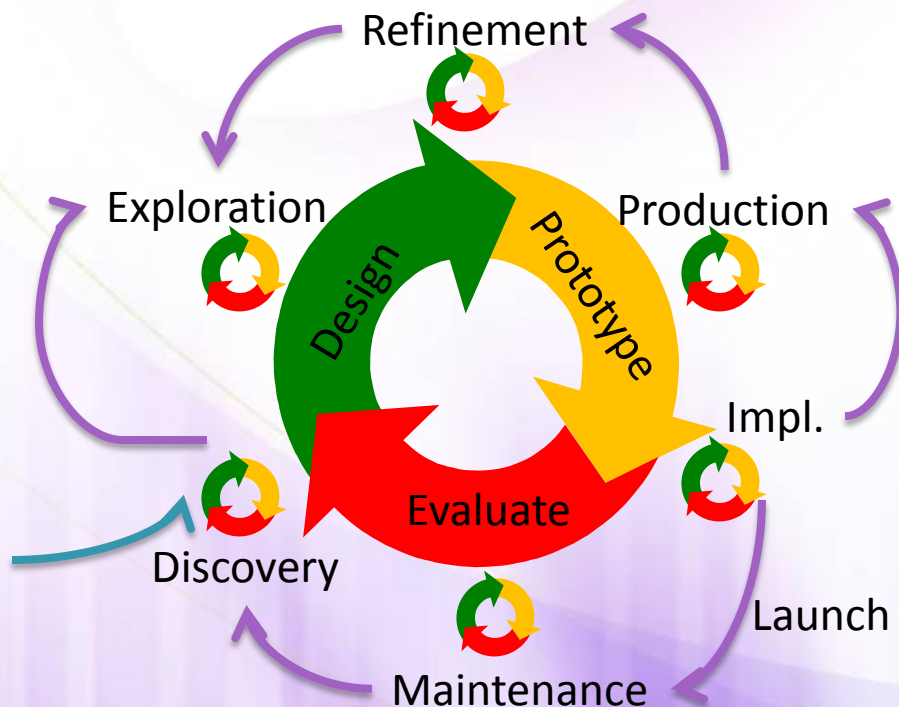
Example ???

Covered topics

- Product Life Cycle
- User Centered Design
- Three Principles
- Product Development
- Interactive Design

Website Development Process

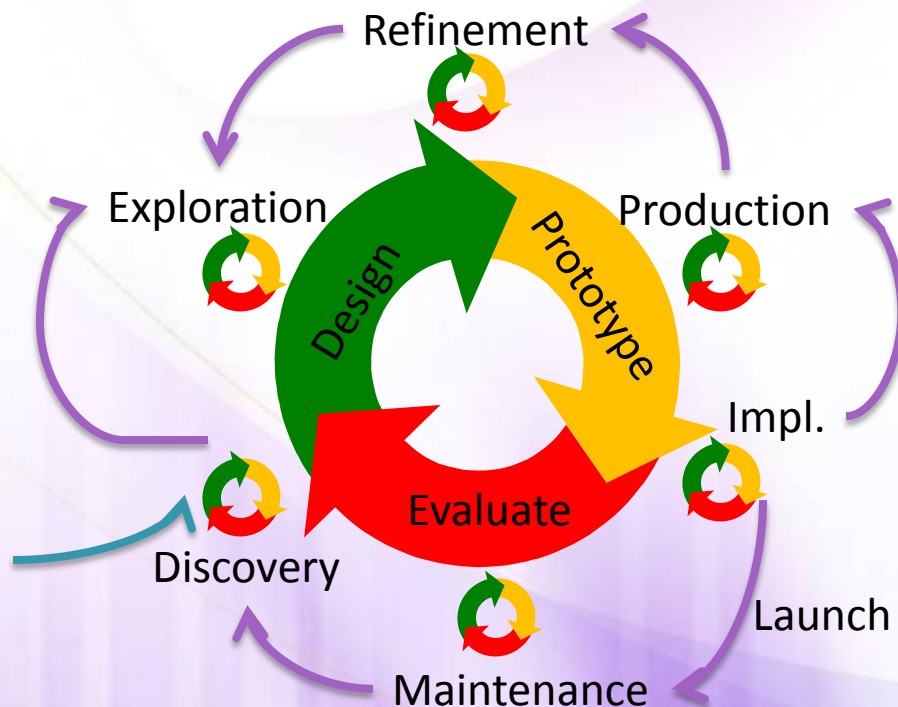
- Both as a whole and at each individual phase, uses iterative design



- Discovery – understanding the target customers and their needs, and conceptualizing the business and customer goals for the web site
- Exploration – generating several rough initial web site designs, of which one or more will be chosen for further development
- Refinement – polishing the navigation, layout and flow of the selected design

Website Development Process

- Both as a whole and at each individual phase, uses iterative design



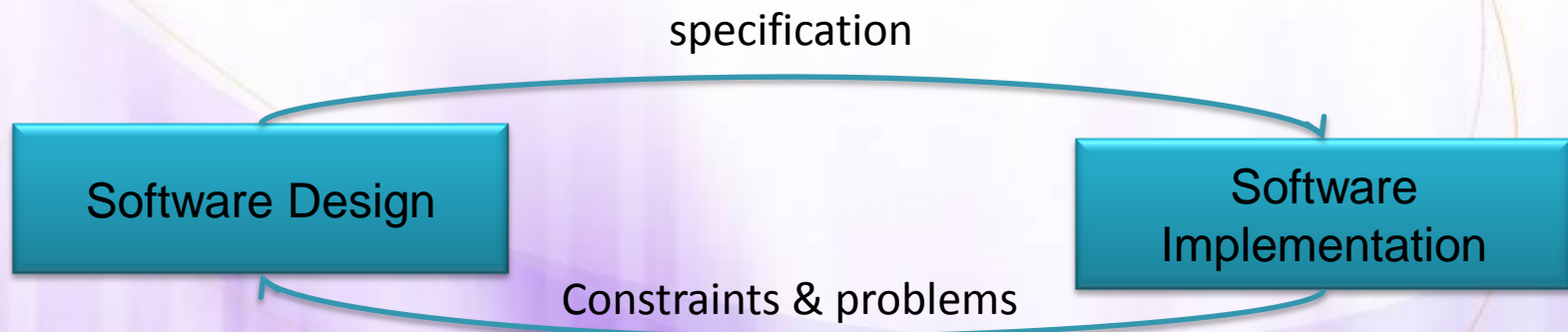
- Production – developing a fully interactive prototype and a design specification
- Implementation – developing the code, content and images for the web site
- Launch – deploying the web site for actual use
- Maintenance – supporting the existing site, gathering and analyzing metrics of success and preparing for the next redesign

Covered topics

- Product Life Cycle
- User Centered Design
- Three Principles
- Product Development
- Interactive Design

Process of user interaction development

- Connections of user interaction development to software engineering
- All these figures depict communication paths, not temporal ordering of activities
- Distinction between software design and implementation



Some practical issues

- Who are the users?

- 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:

- primary: frequent hands-on
- secondary: occasional or via someone else
- tertiary: affected by its introduction, or will influence its purchase.

→ Wider term: stakeholders

Some practical issues

- What are their capabilities?
 - Humans vary in many dimensions!
- Example:
 - 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

Some practical issues

- What are 'needs'?
 - 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

Some practical issues

- 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 & synthesis
 - Seek inspiration: look at similar products or look at very different products

Some practical issues

- 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 and checked regularly)
 - safety: how safe?
 - utility: which functions are superfluous?
 - effectiveness: appropriate support? task coverage, information available
 - efficiency: performance measurements