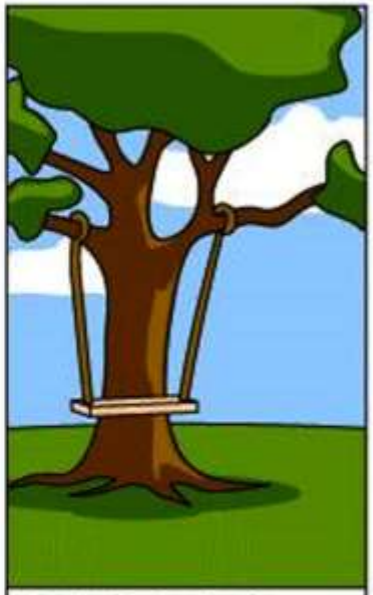
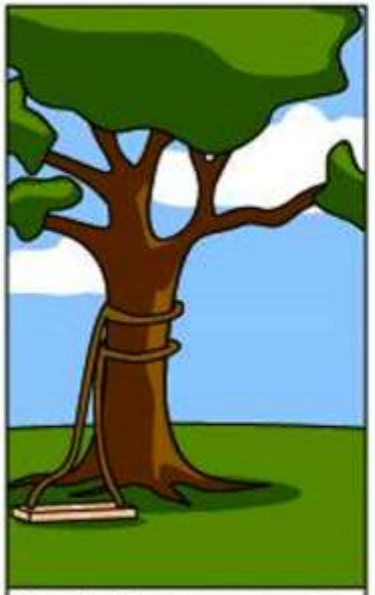
Interactive system development process

How the customer explained it



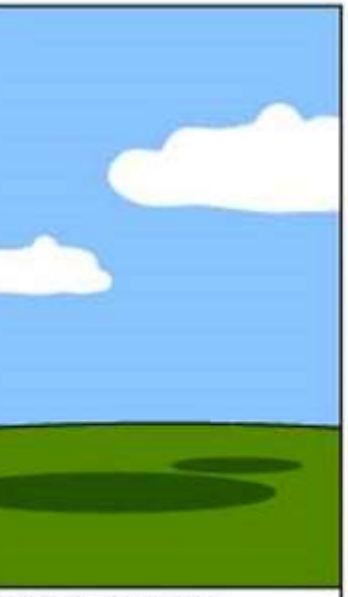
How the Project Leader understood it

How the Analyst designed it

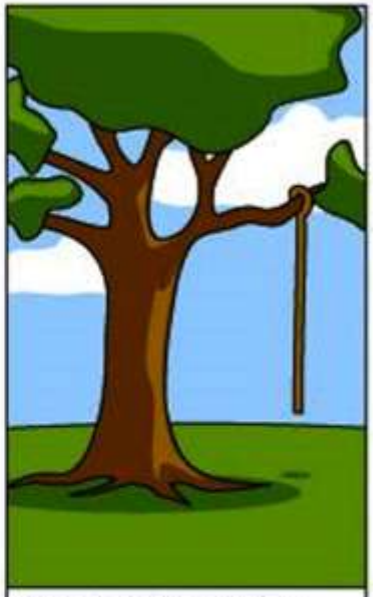


How the Programmer wrote it

How the Business Consultant described it

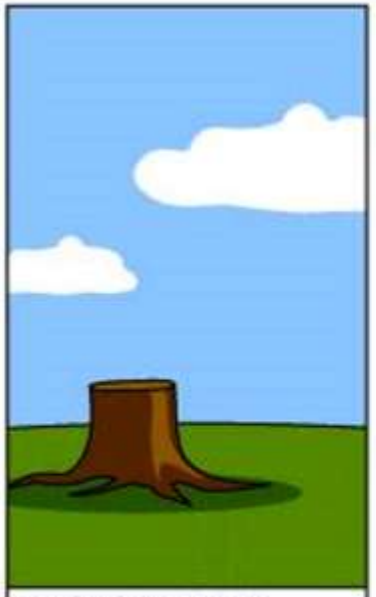


How the project was documented



What operations installed

How the customer was billed



How it was supported

What the customer really needed

Goals of interaction system development process

- Develop usable products
  - Usability means easy to learn, effective to use and provide an enjoyable experience
- Involve users in the interaction design:
  - By means of 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
  - By means of a representation:
    - a plan for development
    - a set of alternatives and successive elaborations

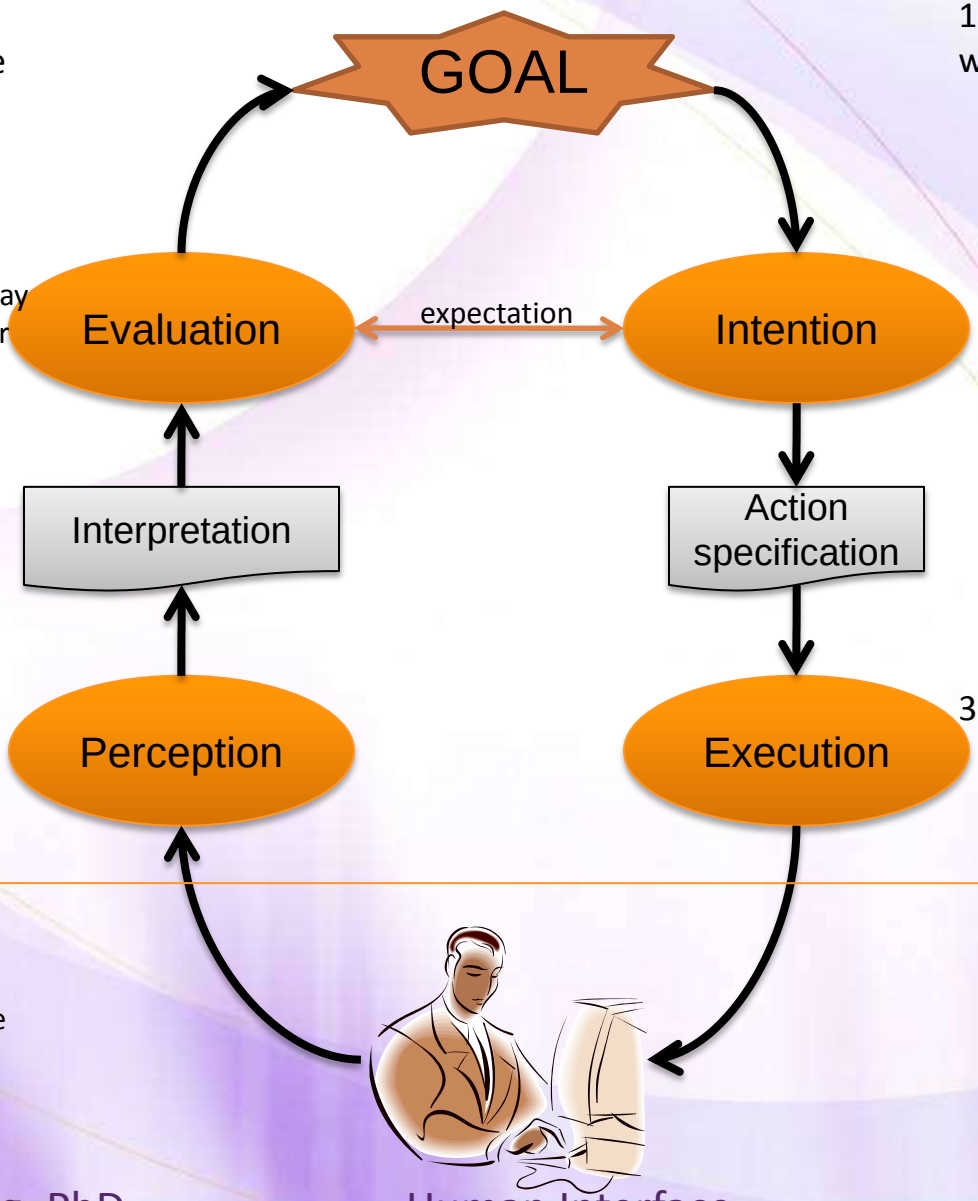
4 stages of an interaction: user activities when performing a task

1. Forming an intention: "What we want to happen"
   - •internal mental characterization of a goal
   - •may comprise goals and sub-goals (but rarely are they well planned)
   - •similar to task semantics
   - •e.g. "writing a business letter"

2. Selecting an action
   - •review possible actions and select most appropriate
   - •similar to mapping between task and compute semantics
   - •e.g. "use the emacs editor to create a file business.letter"

3. Executing the action:
   - carry out the action using the computer
   - similar to mapping between semantics and computer syntax
   - e.g. type "emacs –nw business.letter"

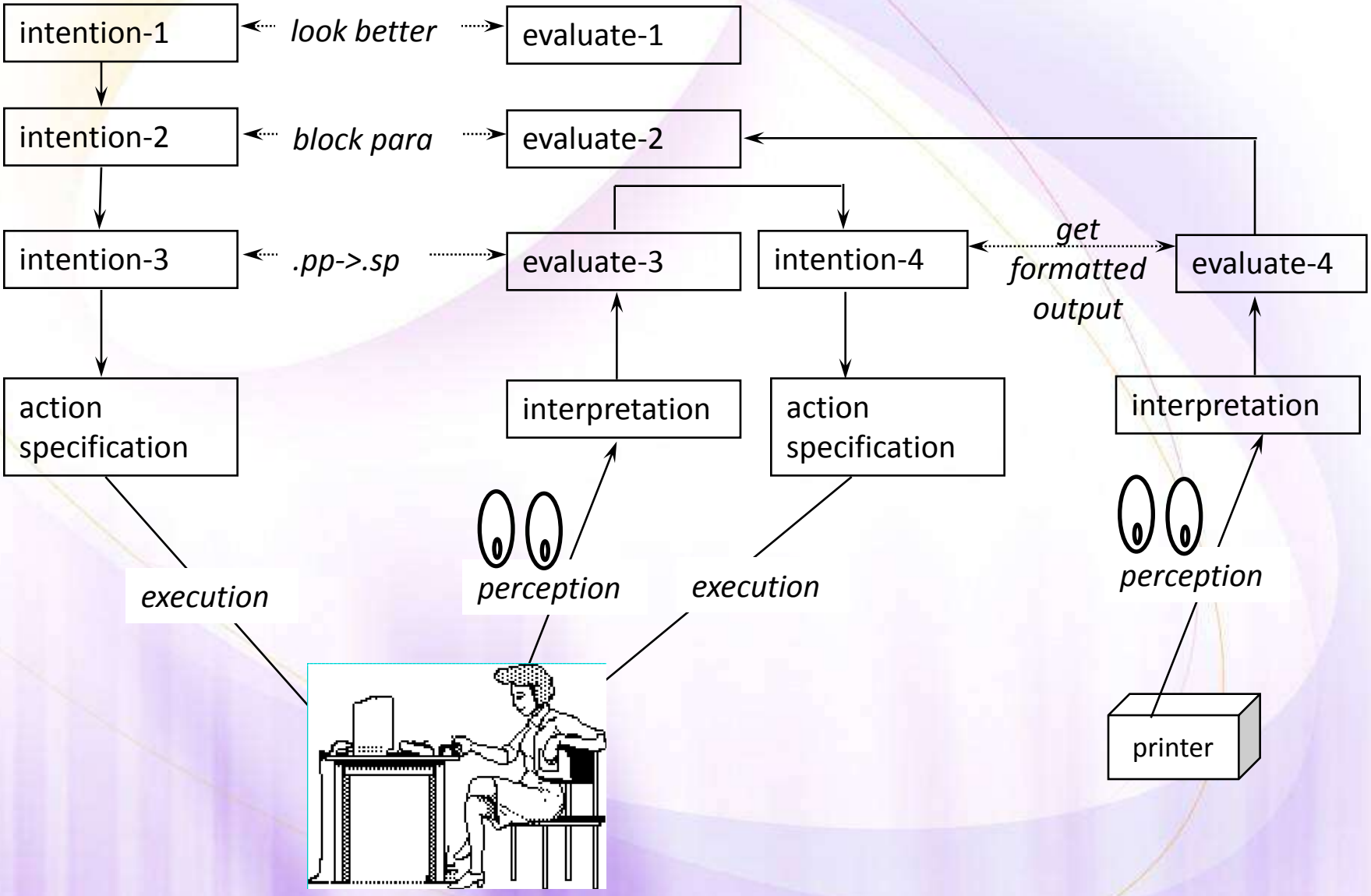
4. Evaluate the outcome
   - •check the results of executing the action
   - •compare it with the expectations
   - •e.g. see if emacs editor is on the display and verify that buffer name is "business.letter"



Mental activity

Physical activity

A typical task: making a business letter look better

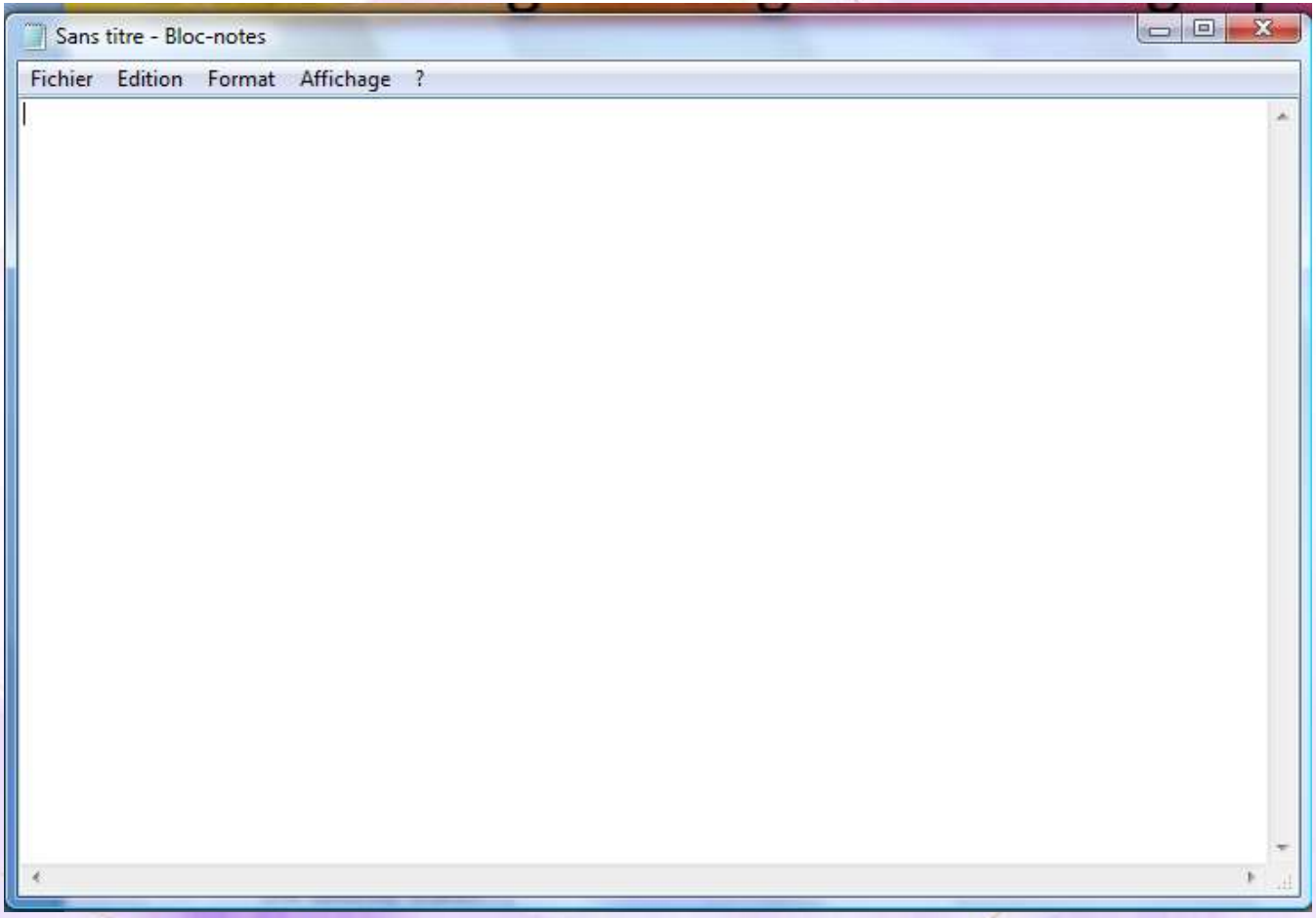


Using four stages to ask design questions

Exercise: making a business letter look better using note pad

How easily can a user:

- determine the function of the system?
- tell what actions are possible?
- determine mapping from intention to selection?
- perform the action?
- tell what state the system is in?
- determining mapping from system state to interpretation?
- tell if system is in the desired state?

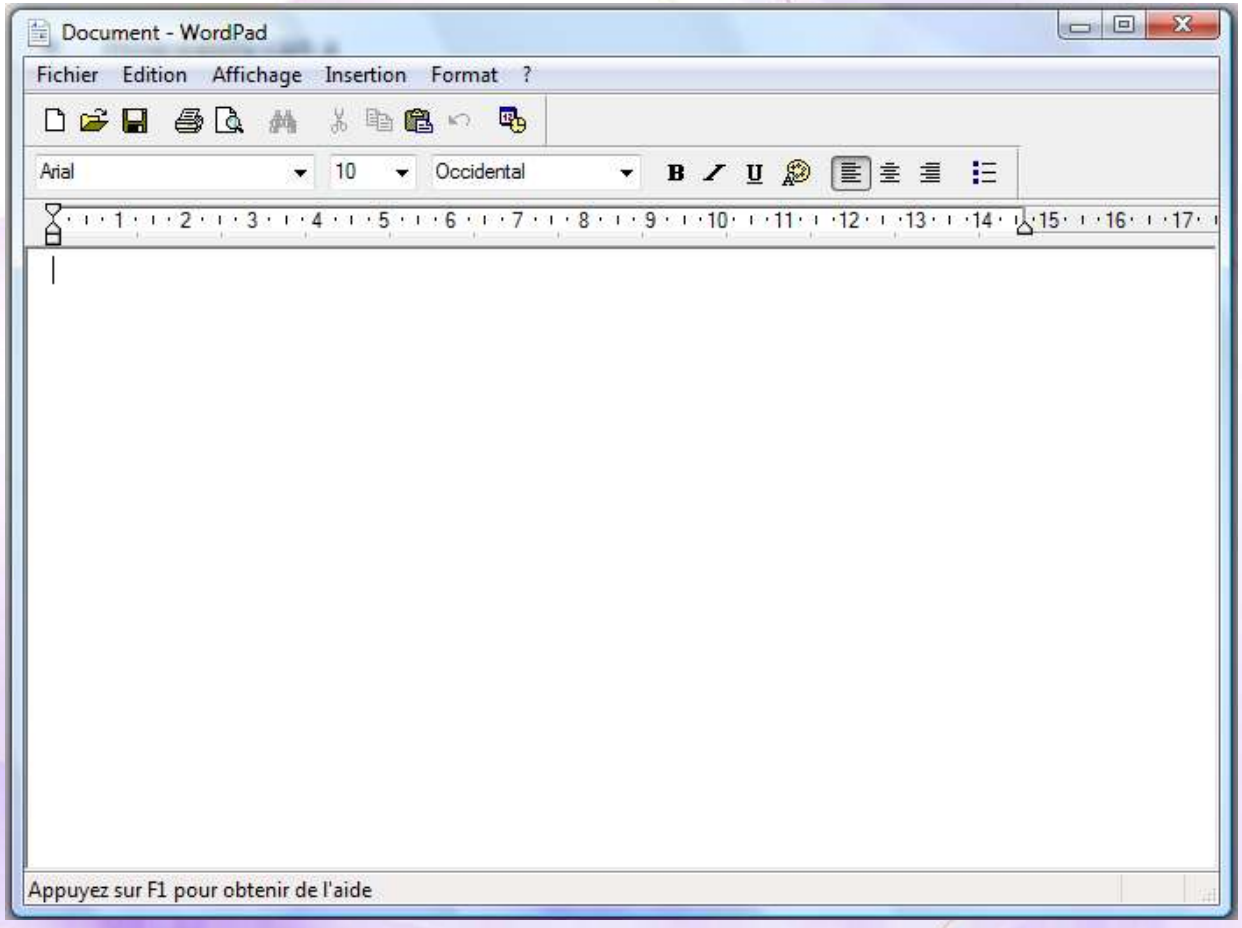


Using four stages to ask design questions

Exercise: making a business letter look better using word pad

How easily can a user:

- determine the function of the system?
- tell what actions are possible?
- determine mapping from intention to selection?
- perform the action?
- tell what state the system is in?
- determining mapping from system state to interpretation?
- tell if system is in the desired state?

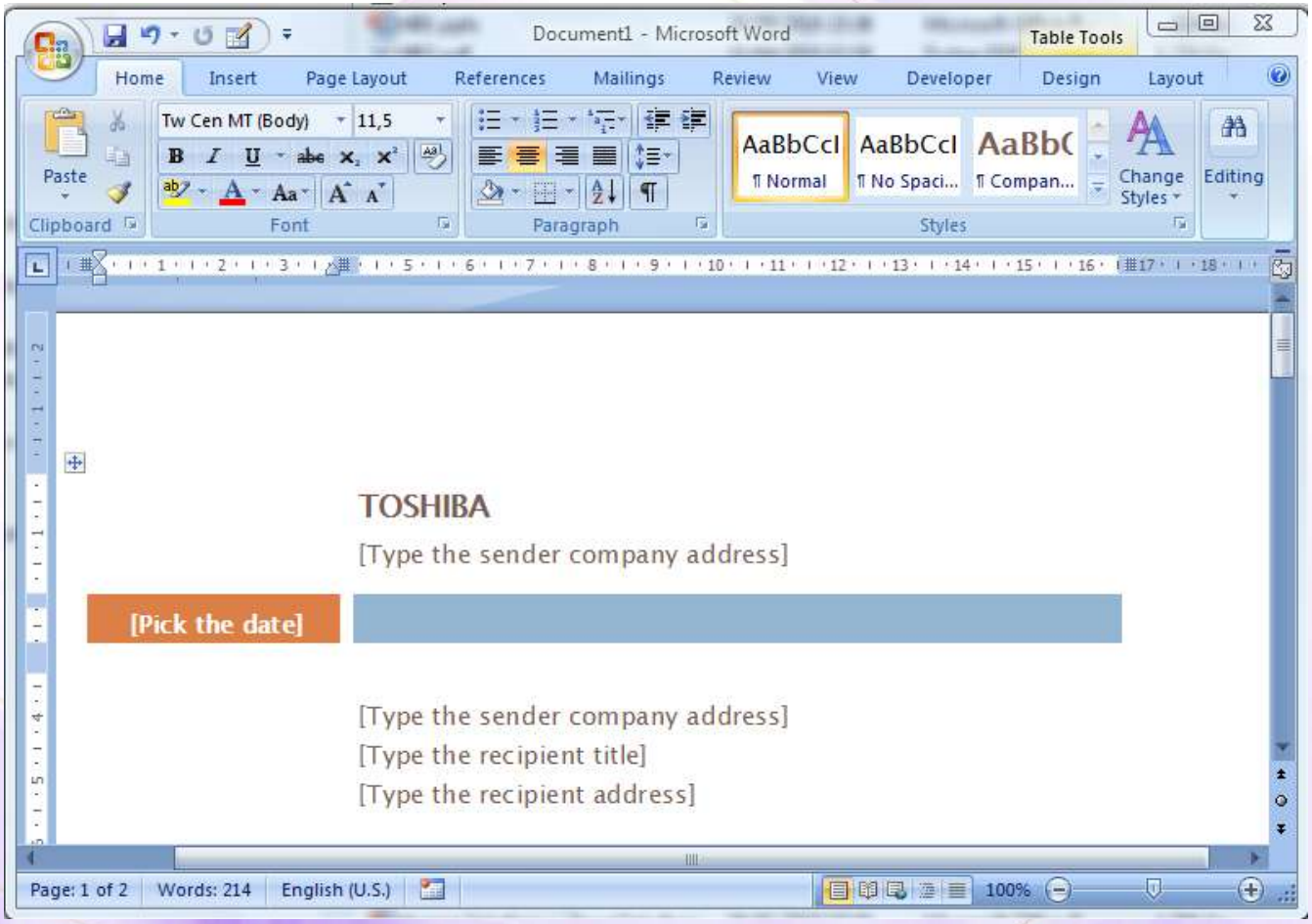


Using four stages to ask design questions

Exercise: making a business letter look better using office word

How easily can a user:

- determine the function of the system?
- tell what actions are possible?
- determine mapping from intention to selection?
- perform the action?
- tell what state the system is in?
- determining mapping from system state to interpretation?
- tell if system is in the desired state?



Using four stages to ask design questions

- How easily can a user:
  - determine the function of the system?
  - tell what actions are possible?
  - determine mapping from intention to selection?
  - perform the action?
  - tell what state the system is in?
  - determining mapping from system state to interpretation?
  - tell if system is in the desired state?

- Questions similar to principles of good design:
  - visibility
  - can see state of application and alternatives for actions
  - good conceptual model
    - consistency in presentations of operations and results
    - coherent system image
  - good mappings
    - relations between actions and results, controls and their effects, system state and what is visible
  - feedback
    - full and continuous feedback about results of actions
- Principle of transparency
  - "the user is able to apply intellect directly to the task; the tool itself seems to disappear"

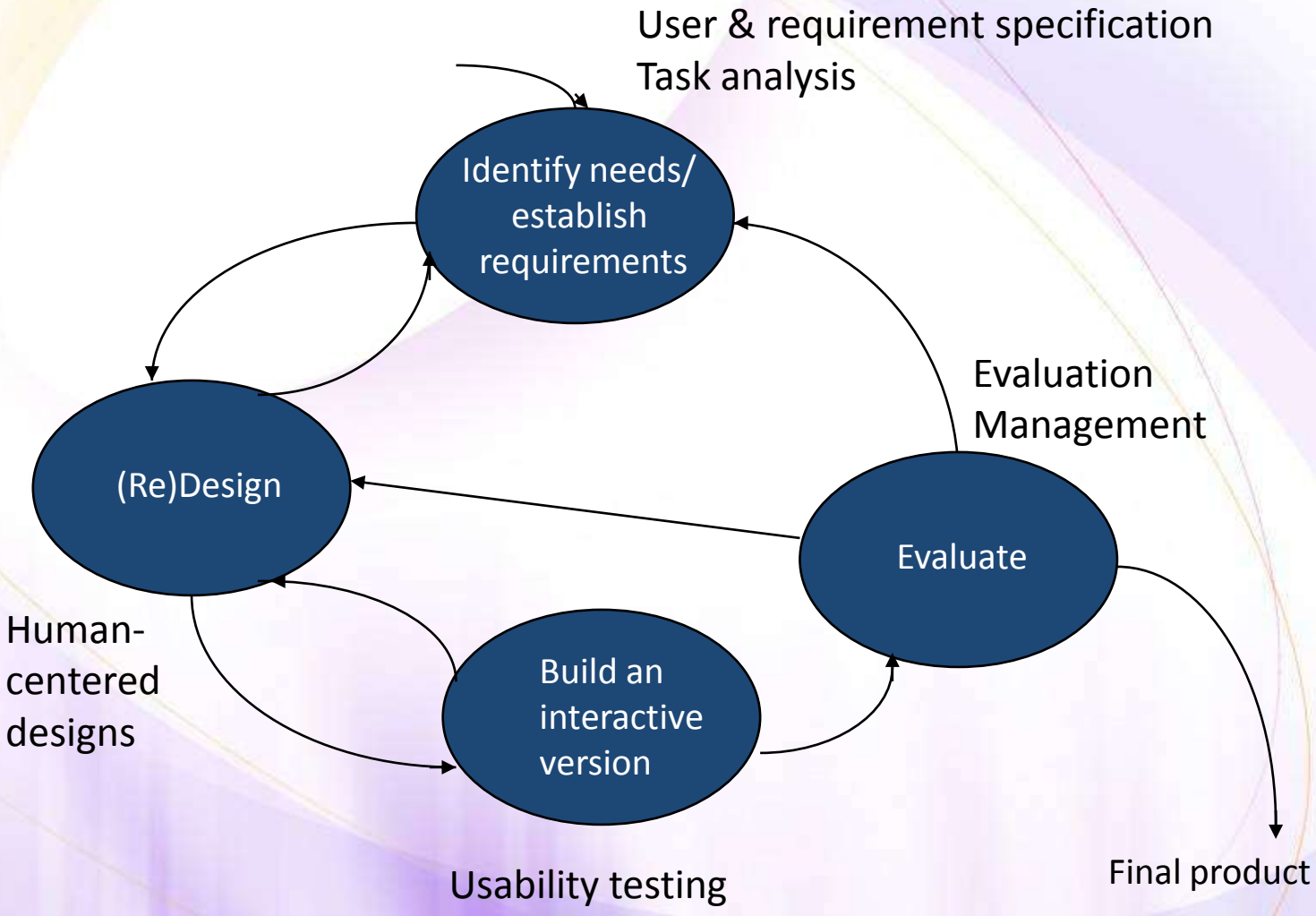
→ This is the design from HCI point of view.

→ What do you remember about phases in software design?

What do you remember about phases in SW Design?

- User requirements
- Specification
  - verification (after all steps?)
- Design
  - Validation
- Prototyping to get feedback from customer
  - Or, to see how something works
- Implementation
- // Testing (includes alpha beta versions)
- Integration…
- Release (with celebration)
  - Packaged and delivered / installers / help, documentation
- Maintenance

A simple interaction design model



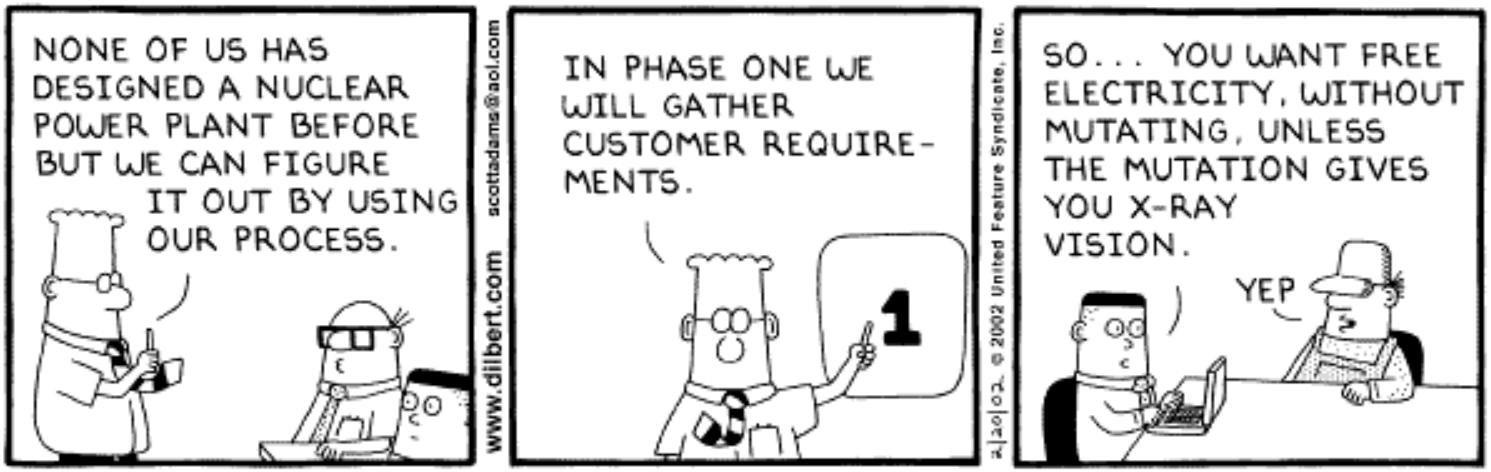
Covered topics

- User & requirement specification
- Task analysis
- Human-centered design
- Usability testing
- Evaluation
- Management

What, how and why?

- What?
  - Understand as much as possible about users, task, context
  - Produce a stable set of requirements
- How?
  - Data gathering activities
  - Data analysis activities
  - Expression of 'requirements'
  - All of this is iterative
- Why?
  - Requirements definition: the stage where failure occurs most commonly
  - Getting requirements right is crucial

But of course...



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
  - …
  - Also usability!
- Data:
  - What kinds of data need to be stored?
  - How will they be stored (e.g. database)?
- Environment / context of use:
  - physical: dusty? noisy? vibration? light? heat? humidity? …. (e.g. jet fighter cockpit, ATM)
  - social: sharing of files, of displays, in paper, across great distances, work individually, privacy for clients
  - organizational: hierarchy, IT department's attitude and remit, user support, communications structure and infrastructure, availability of training

Different kinds of requirements

- User requirements: Who are the users?
  - 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

User Persona

- Prepare a user persona for each user role
- A description of a archetypical, hypothetical, imaginary user
  - Description should focus on the behaviors and goals related to the specific domain of a product
    - What is this person's end goals for using the product?
- Example:
  - Patricia is an English professor who has written several books of poetry.
  - She has been using computers for word-processing since 1980, although the only two programs she's used are Nota Bene and Microsoft Word.
  - She doesn't want to spend time learning details of how a computer works, and tends to store all her files in whatever directory they'd go in if you didn't know about directories. (From Spolsky.)

How Personas Can Be Useful

- Descriptions should help identify
  - workflow and behavior patterns
  - goals
  - environment
  - attitudes
- From multiple user points of view!
- Can help in
  - conceptual models
  - defining usability requirements
  - prioritizing (or cutting) features
  - evaluating alternatives
- Will make engineers more user-focused

Different kinds of requirements

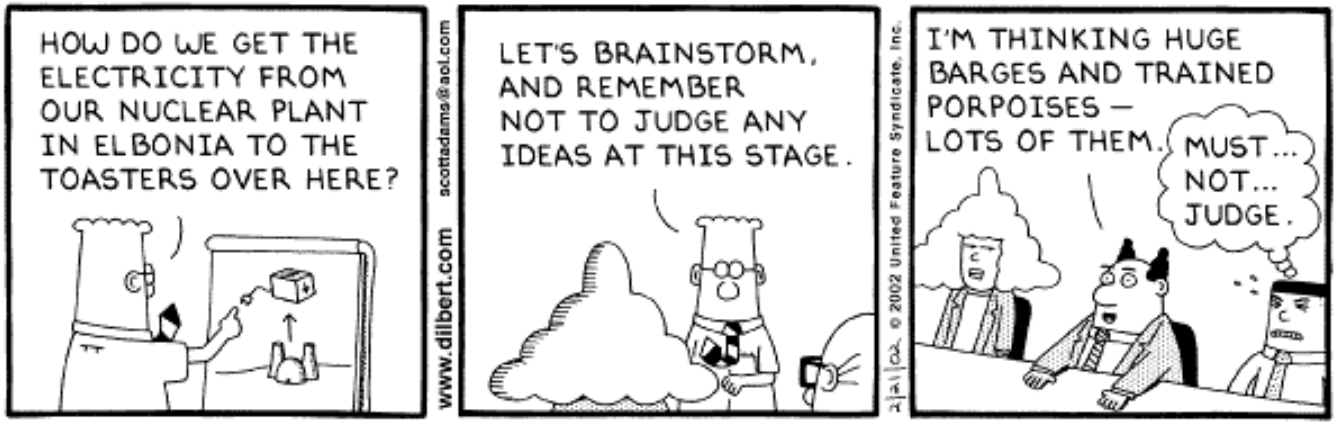
- Usability requirements:
  - learnability,
  - throughput,
  - flexibility,
  - attitude
- Note that user requirements and usability requirements refer to different things

How to get information ?

- Questionnaires:
  - A series of questions designed to elicit specific information
  - Questions may require different kinds of answers:
    - simple YES/NO;
    - choice of pre-supplied answers;
    - comment
  - 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
- Interviews:
  - Forum for talking to people
  - Structured, unstructured or semi-structured
  - Props, e.g. sample scenarios of use, prototypes, can be used in interviews
  - Good for exploring issues
  - But are time consuming and may be infeasible to visit everyone
- Workshops or focus groups:
  - Group interviews
  - Good at gaining a consensus view and/or highlighting areas of conflict

How to get information ?

- Naturalistic observation:
  - Spend time with stakeholders in their day-to-day tasks, observing work as it happens
  - 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
  - Ethnography is one form
- 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



Choosing between techniques

- Data gathering techniques differ in two ways:
  - Amount of time, level of detail and risk associated with the findings
  - Knowledge the analyst requires
- The choice of technique is also affected by the kind of task to be studied:
  - Sequential steps or overlapping series of subtasks?
  - High or low, complex or simple information?
  - Task for a layman or a skilled practitioner?

Some basic guidelines

- Focus on identifying the stakeholders' needs
- Involve all the stakeholder groups
- Involve more than one representative from each stakeholder group
- Use a combination of data gathering techniques
- Support the process with props such as prototypes and task descriptions
- Run a pilot session
- You will need to compromise on the data you collect and the analysis to be done, but before you can make sensible compromises, you need to know what you'd really like
- Consider carefully how to record the data

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

Covered topics

- User & requirement specification
- Task analysis
- Human-centered design
- Usability testing
- Evaluation
- Management

Introduction

- Task analysis is the study of the way people perform tasks with existing systems
  - Ex: to wash your car
    - Park the car on the right spot;
    - Buy/search cleansing material;
    - Wash and polish it until is shines;
    - Put away your stuff;
    - Go watch television.
- Three approaches
  - Task decomposition
  - Knowledge-based methods
  - Entity-relation based techniques

Task decomposition

- Observe the user (from the outside)
  - What, not why
  - Towards conceptual model
- Hierarchical task analysis
  - Wash the car and polish it

    1. Wash the outside:

       1.1 Make it wet

       1.2 Soapen it

       1.3 Brush it clean

       1.4 Use the water hoase

       1.5 After drying it, rub it in with wax

       1.6 rub it out (until it shines)

    2. Wash the inside:
- Planning in HTA
  - Fixed sequential tasks
  - Optional tasks
  - Waiting for events
  - Cycles
  - Time-sharing
  - Discretionary
  - Mixtures
- Example:
  - Plan 0: do 1 and 2 in random order
  - Plan 1: do 1.1 - 1.6 in sequential order

Knowledge-based techniques

- Describe all objects and actions
- Build an ontology (hierarchical knowledge description) from these elements
- Acronyms
  - TAKD (task analysis for knowledge description)
  - TDH (task descriptive hierarchy)
  - KRG (knowledge representation grammar)
- Classification
  - Function
  - Shape
  - Location

- Example:
  - Pictures
    - Cartoons
    - Photographs
      - Landscapes
        - Mountains
        - Seasides
    - Portraits
      - Family
      - Celebrity
    - Graphics
    - Artwork

Why do we need task analysis?

- Task analysis: write down the objects, actions, features exhaustively.
  - Manuals
  - Requirements capture/systems design
  - Detailed interface design (TDH trees, UI, dialog)

→ Ascertain the user's needs

- Determine what tasks and subtasks must be carried out
- Include tasks which are only performed occasionally. Common tasks are easy to identify.
- Functionality must match need or else users will reject or underutilize the product

→ Ensure reliability

- Actions must function as specified
- Database data displayed must reflect the actual database
- Appease the user's sense of mistrust
- The system should be available as often as possible
- The system must not introduce errors
- Ensure the user's privacy and data security by protecting against unwarranted access, destruction of data, and malicious tampering

Why do we need task analysis?

- Task analysis: write down the objects, actions, features exhaustively.
  - Manuals
  - Requirements capture/systems design
  - Detailed interface design (TDH trees, UI, dialog)

→ Complete projects on time and within budget

- Late or over budget products can create serious pressure within a company and potentially mean dissatisfied customers and loss of business to competitors

→ Promote standardization, integration, consistency, and portability

- Standardization: use pre-existing industry standards where they exist to aid learning and avoid errors (e.g. the W3C and ISO standards)
- Integration: the product should be able to run across different software tools and packages (e.g. Unix)
- Consistency:
  - compatibility across different product versions
  - compatibility with related paper and other non-computer based systems
  - use common action sequences, terms, units, colors, etc. within the program
- Portability: allow for the user to convert data across multiple software and hardware environments

Covered topics

- User & requirement specification
- Task analysis
- Human-centered design
- Usability testing
- Evaluation
- Management

What is User-Centered Design?

- An approach to UI development and system development.
- Focuses on understanding:
  - Users, and
  - Their goals and tasks, and
  - The environment (physical, organizational, social)
- Pay attention to these throughout development

ISO on User-centered Design

- ISO 13407 describes human-centered design processes for interactive systems
- Principles of human-centered design:
  - Active involvement of users
  - Appropriate allocation of function between user and system
  - Iteration of design solutions
  - Multidisciplinary design teams
- Essential activities in human-centered design:
  - Understand and specify the context of use
  - Specify the user and organizational requirements
  - Produce design solutions (prototypes)
  - Evaluate designs with users against requirements

Are You Experienced in HC Design ?

- Think about a significantly complex software project you've been involved in
  - Work, research, course, etc.
- Did it seem like an example of human-centered design?
  - How did it, or how did it not?

Covered topics

- User & requirement specification
- Task analysis
- Human-centered design
- Usability testing
- Evaluation
- Management

Usability

- A definition from ISO standard 9241
  - The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.
- Effectiveness:
  - accuracy and completeness in achieving goals
- Efficiency:
  - resources expended…
- Satisfaction:
  - comfort, acceptability (happiness, pleasure)

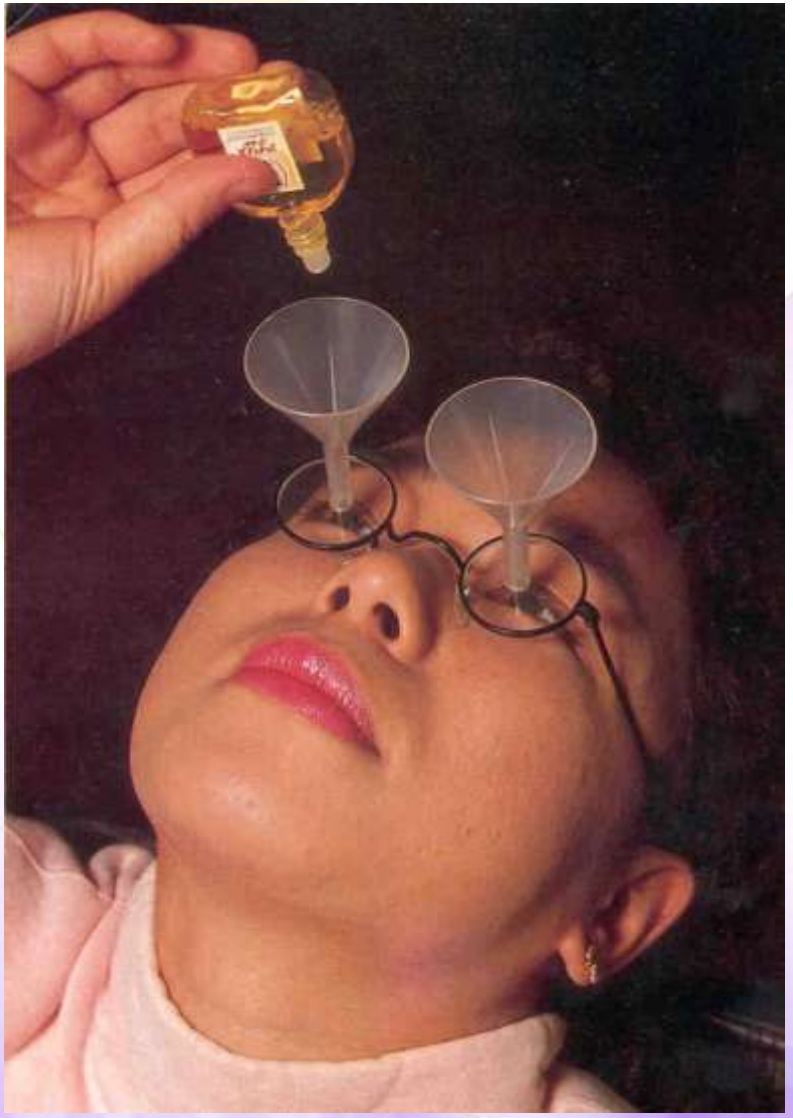
Discussion

- Effectiveness, efficiency and satisfaction:
  - Are all of these equally important?
  - All the time?

Vu Thi Huong Giang, PhD

Human Interface

2009-2010



101 Unuseless Inventions by Kenji Kawakami


ESKOM VEHICLES

Covered topics

- User & requirement specification
- Task analysis
- Human-centered design
- Usability testing
- Evaluation
- Management

Usability measures

- Define the target user community and class of tasks associated with the interface
  - Communities evolve and change
- 5 human factors central to community evaluation:
  - Time to learn: How long does it take for typical members of the community to learn relevant task?
  - Speed of performance: How long does it take to perform relevant benchmarks?
  - Rate of errors by users : How many and what kinds of errors are made during benchmark tasks?
  - Retention over time: Frequency of use and ease of learning help make for better user retention
  - Subjective satisfaction: Allow for user feedback via interviews, free-form comments and satisfaction scales

Usability measures

- Trade-offs in design options frequently occur.
  - Changes to the interface in a new version may create consistency problems with the previous version
  - Changes may improve the interface in other ways or introduce new needed functionality.
- Design alternatives can be evaluated by designers and users via mockups or high-fidelity prototypes.
  - The basic tradeoff is getting feedback early and perhaps less expensively in the development process versus having a more authentic interface evaluated.

Covered topics

- User & requirement specification
- Task analysis
- Human-centered design
- Usability testing
- Evaluation
- Management

Management concerns

- Understanding users and their tasks
  - Task-centered system design
    - how to develop task examples
    - how to evaluate designs through a task-centered walk-through
- Designing with the user
  - User centered design and prototyping
    - methods for designing with the user
    - low and medium fidelity prototyping
  - Evaluating interfaces with users
    - the role of evaluation in interface design
    - how to observe people using systems to detect interface problems

Summary

- Four basic activities in the design process
  - Identify needs and establish requirements
  - Design potential solutions ((re)-design)
  - Choose between alternatives (evaluate)
  - Build the artefact

- Human-centered design rests on three principles
  - Early focus on users and tasks
  - Empirical measurement using quantifiable & measurable usability criteria
  - Iterative design

Lifecycle models show how these are related