SFWRENG 4HC3 6HC3 /COMPSCI 4HC3
Human Computer Interfaces
16 – Testing

SFWRENG 4HC3 6HC3 | COMPSCI 4HC3

Danny Papagiannis

Acknowledgement: Parts of these lectures are based on material prepared by Ron Baecker, Ravin Balakrishnan, John Chattoe, Ilona Posner, Scott Klemmer, Jeremy Bradbury and Christopher Collins

Today

- By the end of today's lecture, you will be able to:
  - List the three goals of evaluation
  - Describe the principles of learnability, flexibility, and robustness
  - Choose specific evaluation techniques that can be used to:
    - assess the functionality and usability of a system
    - identify possible problems.
  - Distinguish two main classes of evaluation techniques
    - Expert evaluations
    - User-based evaluations (introduced, continue next time)

Practical HCI

EVALUATING PROTOTYPES

Goals of Evaluation

- Evaluation in the context of Human-Computer Interaction has 3 main goals:
  - Test the system functionality (and accessibility of the functionality)
  - Test the user’s experience interacting with the system
  - Identify specific system problems

Goal #1: Test the System Functionality

- Testing the system functionality requires us to answer the following questions:
  - Does the functionality cover all of the user requirements? This measures the extent of the functionality
  - Is needed functionality accessible to a user during a given task? Functionality is not useful if it can’t be accessed at the appropriate time!
  - Does the functionality meet the user’s expectations? This helps us understand if the functionality allows the user to complete tasks as expected

Goal #2: Test the User’s Experience

- Testing the user experience is about usability not functionality
- In this context we are interested in the following questions:
  - Is the user satisfied with the system?
  - Is there an emotional reaction to the system?
  - Does the system overload the user?
- Usability can include ensuring that the design meets different principles:
  - Learnability, flexibility, robustness

"Learnability is the ease with which new users can begin effective interaction and achieve maximal performance."

- Dix, Finlay, Abowd, Beale

LEARNABILITY

Learnability

- Predictability
- Synthesizability
- Familiarity
- Generalizability
- Consistency

Learnability -> Predictability

- In general users like to be able to predict the response from the system for a given action
- The response predicted by a user is based on past experience with the system
- Operation Visibility
  - Related to predictability
  - Does the user know what operations can be performed next?

Learnability -> Synthesizability

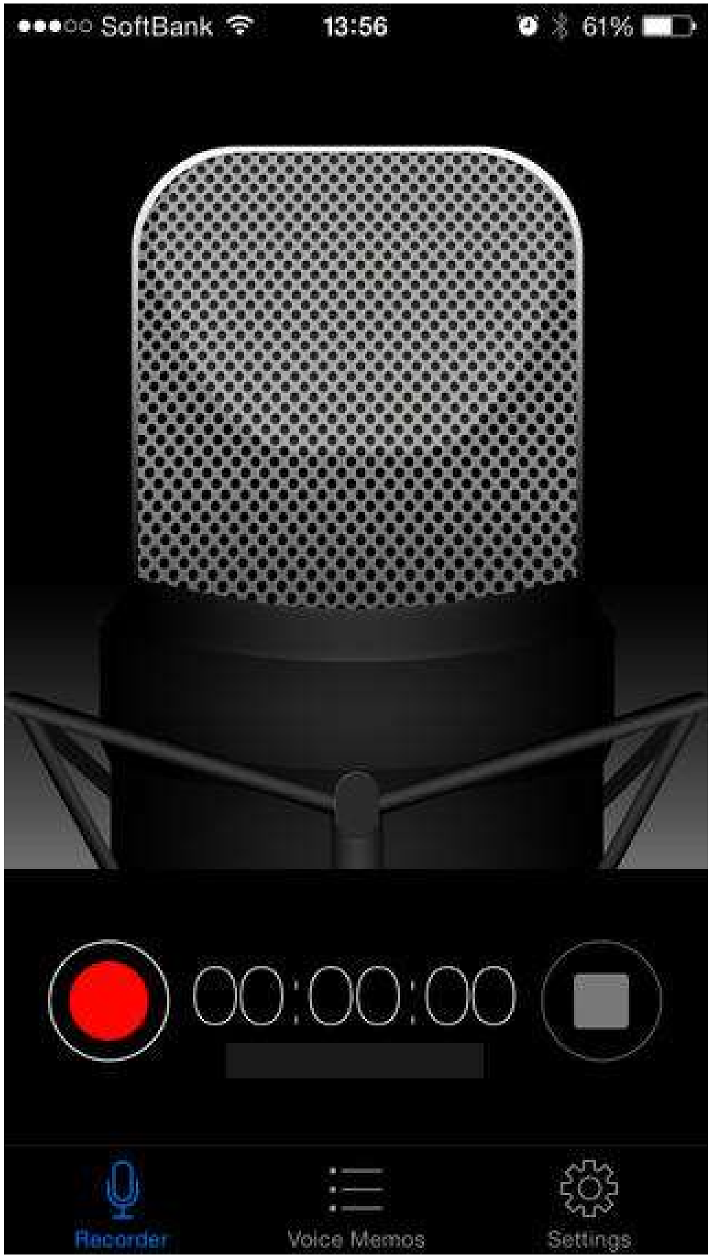
- The ability of the user to assess the effect of past operations on the current state
- Immediate vs. eventual awareness
  - Immediate: You see the results or receive a notification, e.g. drag and drop a file visually
  - Eventual: You can see changes after explicitly asking for them, e.g. pressing “F5” to refresh a folder view

Learnability -> Synthesizability

- Related to awareness of system status:
  - Examples: providing confirmation of an online purchase, or confirmation of removing an item from an online shopping cart.

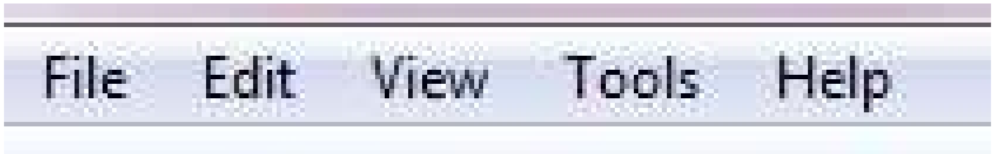
Learnability -> Familiarity

- Make the right first impression
  - Metaphors, “guessability”
- Assesses the benefit from:
  - real-world experience
  - using other interactive systems
  - affordances


●●●○○ SoftBank
13:56
61%
00:00:00
Recorder
Voice Memos
Settings

Learnability -> Generalizability

- Apply knowledge from one situation to a new situation
- For example, cut/copy/paste operation generalizes to many applications
- Consistency



  - Recall the consistency heuristic
  - *Many other principles can be 'reduced' to consistency, e.g. Generalizable = consistent across interfaces, Familiarity = consistent with past experiences*

Consistency Caution

FLEXIBILITY

“Flexibility is the multiplicity of ways in which the user and system exchange information.”

- Dix, Finlay, Abowd, Beale

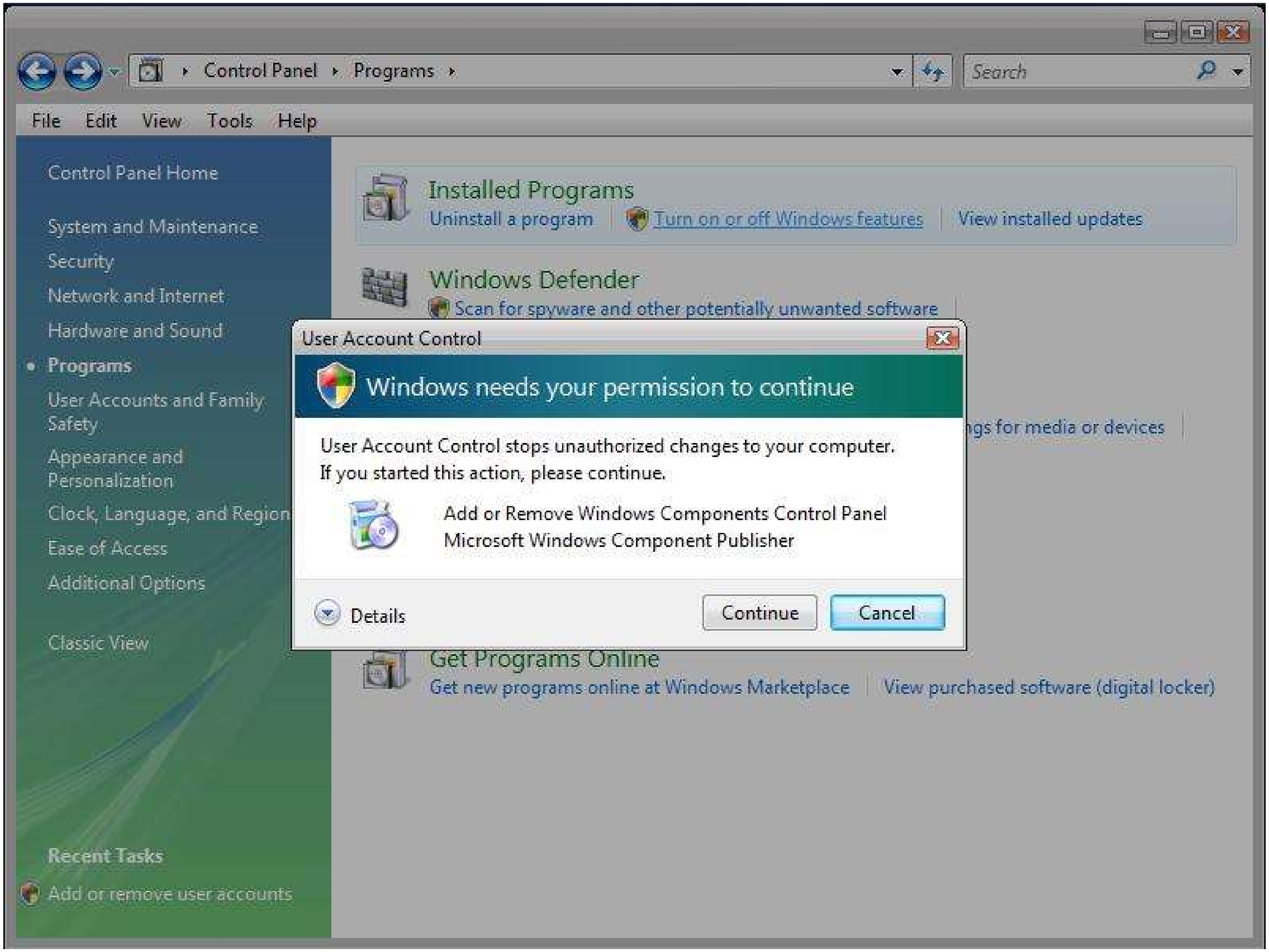
Flexibility

- Dialog initiative
- Multi-threading
- Task Migratability
- Substitutivity
- Customizability

Flexibility -> Initiative

- System pre-emptive vs. user pre-emptive

System Pre-emptive



User Account Control

Windows needs your permission to continue

User Account Control stops unauthorized changes to your computer.
If you started this action, please continue.

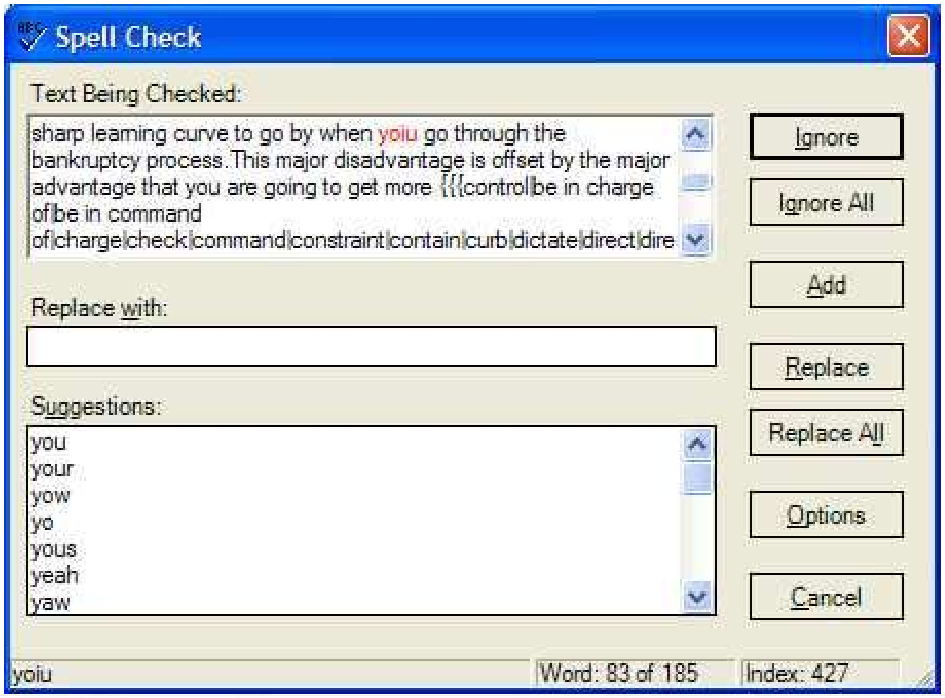
Add or Remove Windows Components Control Panel
Microsoft Windows Component Publisher

Flexibility -> Multi-threading

- Multi-threading
  - Related to multi-tasking
  - Different kinds of multi-threading: concurrent, interleaved
    - audible bell for new mail while browsing a website
    - multiple touch inputs on a tabletop display

Flexibility -> Task Migratability

- the transfer of control for execution of tasks between system and user
  - Example: Spell check, Auto-pilot



Spell Check

Text Being Checked:

sharp learning curve to go by when yoiu go through the bankruptcy process.This major disadvantage is offset by the major advantage that you are going to get more {{{control|be in charge of|be in command of|charge|check|command|constraint|contain|curb|dictate|direct|dire

Replace with:

Suggestions:

you
your
yow
yo
yous
yeah
yaw

yoiu | Word: 83 of 185 | Index: 427

Flexibility -> Substitutivity

- Related to input and output
- Can we substitute equivalent inputs? outputs?
- Input example:
  - in Microsoft Office we can input paragraph spacing in pt, cm, etc.
- Output example:
  - temperature as a thermometer graphic or a line chart (different representation)

Flexibility -> Customizability

- Can the interface by changed to suit the user’s preferences?
- Example: hide/show toolbars in Firefox

ROBUSTNESS

Robustness

- Observability
- Recoverability
- Responsiveness
- Task Conformance

“Robustness is the level of support provided to the user in determining successful achievement and assessment of goals.”

- Dix, Finlay, Abowd, Beale

Robustness -> Observability

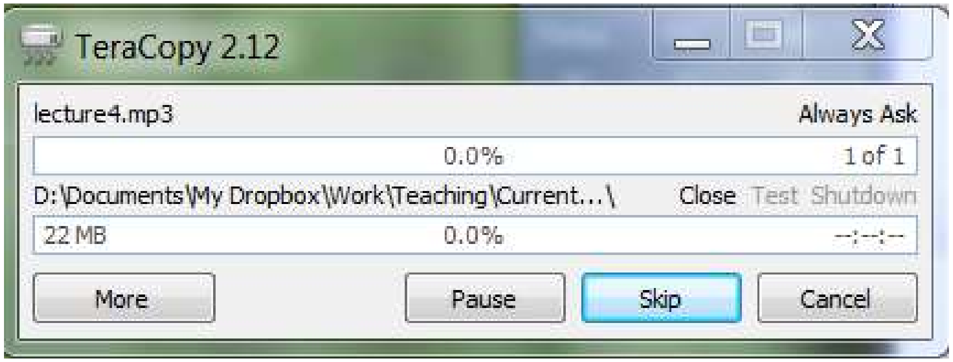
- Ability of the user to evaluate the internal state of the system from its perceivable representation
- Which heuristic does this relate to?

Robustness -> Observability

- Ability of the user to evaluate the internal state of the system from its perceivable representation
- Which heuristic does this relate to?
  - Related to “visibility of system status” heuristic

Robustness -> Observability

- Browsability: change view without altering system state
- Defaults: static defaults and dynamic defaults
  - Static: Yes/No/Skip
  - Dynamic: font colour palette in Word



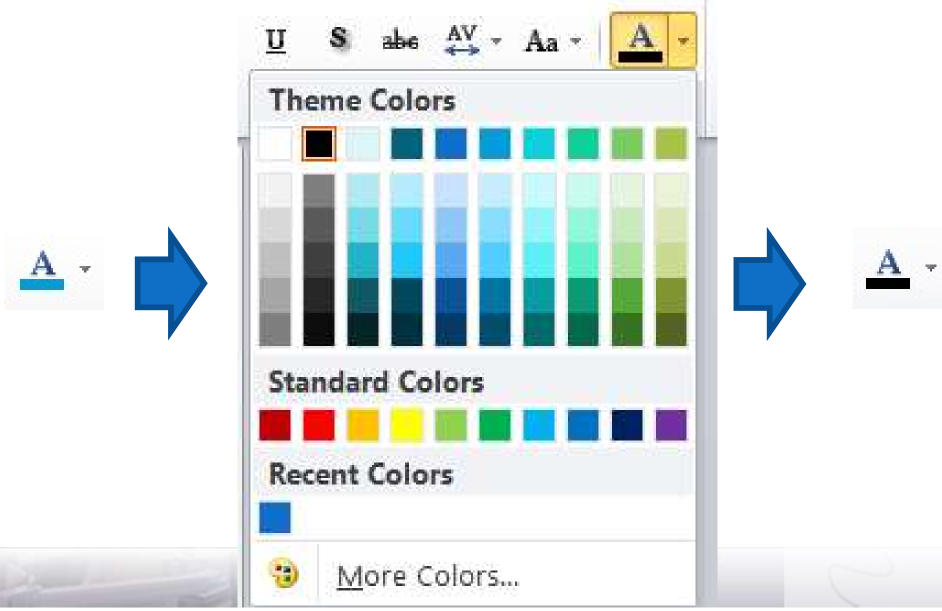
TeraCopy 2.12

lecture4.mp3 Always Ask

0.0% 1 of 1

D:\Documents\My Dropbox\Work\Teaching\Current...\ Close Test Shutdown

22 MB 0.0% --:--:--



Robustness -> Observability

- Browsability: change view without altering system state
- Defaults: static defaults and dynamic defaults
- Reachability: possibility of navigation through the observable system states
- Persistence: duration of effect of communication
  - Email beep vs. flag in notification area
  - Tri-state buttons: hover for preview/click for permanent <demo>

Robustness -> Recoverability

- Can the user recover after making an error?
- Forward (start over) vs. backward (undo) error recovery

Robustness -> Responsiveness

- Communication rate of interaction
  - Stability is important
  - Provide awareness of progress for long processes

Robustness -> Task Conformance

- Does the system allow the user to complete the tasks he/she wants to do?

Goals of Evaluation

- Evaluation in the context of Human-Computer Interaction has 3 main goals:
  - Test the system functionality (and accessibility of the functionality)
  - Test the user's experience interacting with the system
  - Identify specific system problems

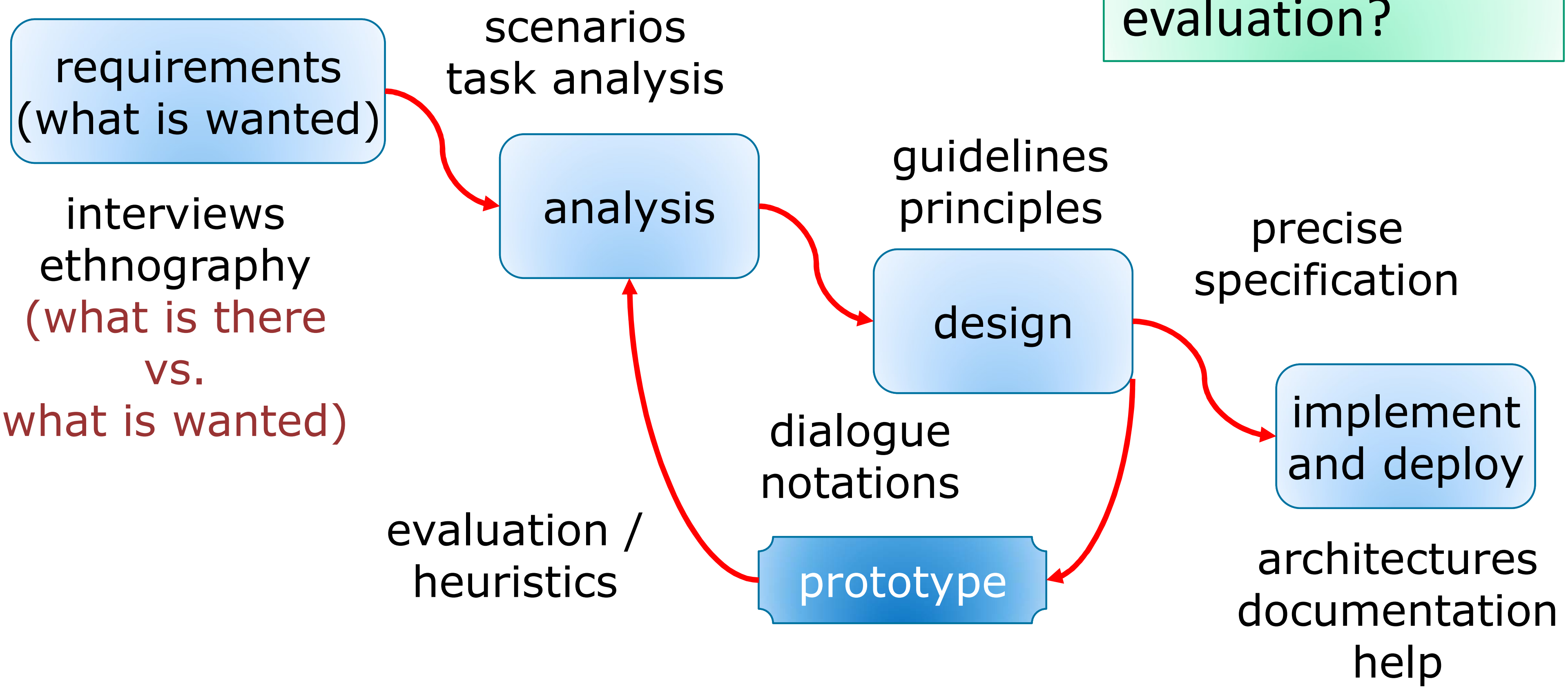
Goal #3: Identify Specific Problems

- The distinction between this goal and the other two goals is that:
  - This goal is about identifying specific problems that can be fixed
  - The first two goals test the functionality and user experience – these are often the reason the specific problem actually exists

The Design Process

Recall

When do we do evaluation?



requirements (what is wanted)

interviews
ethnography
(what is there
vs.
what is wanted)

scenarios
task analysis

analysis

guidelines
principles

design

precise
specification

implement and deploy

architectures
documentation
help

dialogue
notations

prototype

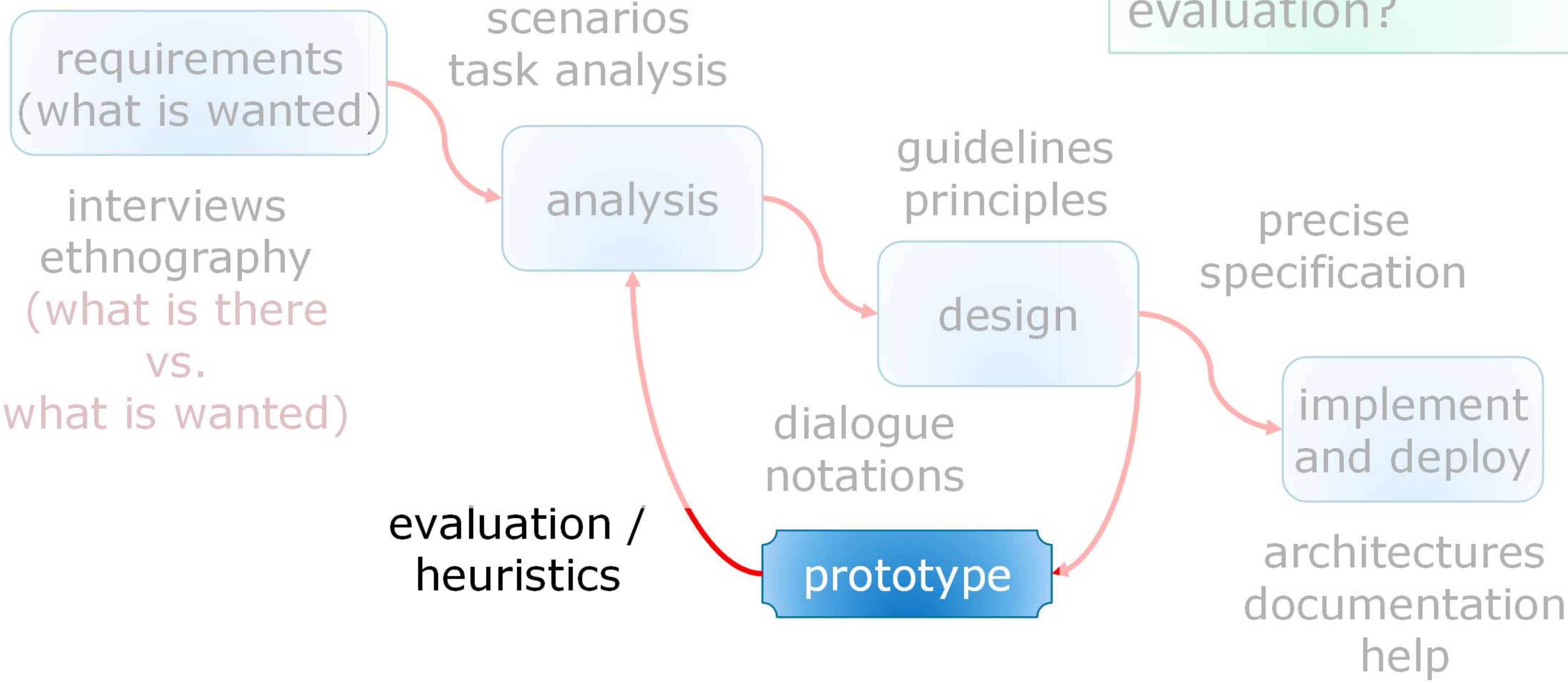
evaluation /
heuristics

(Source: Dix, Finley, Abowd, Beale, *“Human-Computer Interaction”)*

The Design Process

Recall

When do we do evaluation?



requirements (what is wanted)

interviews
ethnography
(what is there
vs.
what is wanted)

scenarios
task analysis

analysis

guidelines
principles

design

precise
specification

implement
and deploy

architectures
documentation
help

dialogue
notations

prototype

evaluation /
heuristics

(Source: Dix, Finley, Abowd, Beale, *"Human-Computer Interaction")*

Benyon text
Chapter 10

Evaluating HCI

EXPERT ANALYSIS

Discount Usability Engineering

- Expert Analysis techniques
  - Heuristic Evaluation
  - Cognitive Walkthrough
- Find problems, but not complete, nor ecologically valid
- Formative
  - Help inform design at early stage
- Not summative
  - Cannot make claims about usability of a finished product

Expert Analysis

- Expert analysis techniques involve the use of experts to assess a system's user interface instead of using users
- The benefit of expert analysis is that it costs less than evaluations that involve users
  - User participation is typically more expensive
- The drawback of expert analysis is that we are not evaluating actual users interacting with the system

Find 80% of the problems before bringing in real stakeholders.

Cognitive walkthrough

- Walkthroughs are a technique you may have seen previously in software engineering
- In the context of software engineering quality assurance you learned about code reviews and code walkthroughs
- Cognitive walkthroughs apply the same principles but in a different context

"A cognitive walkthrough is [...] learning through exploration."

- Dix, Finlay, Abowd, Beale

Code vs. Cognitive walkthrough

Code Walkthrough

- conducted by expert (e.g., developer)
- reviews program code
- walks through an algorithm or portion of the program
- identifies problems based on style, correctness, etc.

Cognitive Walkthrough

- conducted by expert (e.g., usability expert)
- reviews the interface
- walks through the user interface actions necessary for a user to complete a task
- identifies problems related to the usability of the interface

Cognitive walkthrough

- Goal is to understand usability, including how easy a system is to learn
- People prefer to learn software by doing, rather than through training and manuals
- The checks that are made during the walkthrough ask questions that address exploratory learning
  - Review each step the participant did and provide a ‘story’ about why that step is or is not good for a new user

Terms

- Evaluator: the person leading the evaluation, often a member of the prototype design team
- Expert: the usability expert who is a *participant* in the evaluation (usually use 3-5 experts)

Cognitive walkthrough

Required Artifacts

- A system prototype (or specification)
  - Should be as complete as possible
- Description of the task being used for the walkthrough
- The complete, written list of actions necessary to complete the task
- User profile
  - Details about skill level, background, etc.

Cognitive walkthrough

How does an expert conduct a cognitive walkthrough?

- Once the artifacts are gathered by the evaluator they are given to the expert for review
- As the expert walks through each action and move towards completing a task a set of questions must be answered
- Throughout the process the expert will document the procedure and any problems encountered

Nuance

- The expert does not try to solve the task for him/herself … this would be a ‘think aloud’ study and it is usually conducted with *real stakeholders*
- The expert follows the ideal task completion sequence as outlined by the evaluators and comments directly on problems

Cognitive walkthrough

Questions

- Upon completing an action the expert asks himself/herself the following questions:
  - *Will the people using the system try to achieve the right effect? (or, "Is the effect of the action the same as the users' goal at that point?")*
  - *"Will users see that the action is available?"*
  - *"Once users have found the correct action, will they know it is the one they need?"*
  - *After the action is taken will users understand the feedback they get?"*

- Benyon, p. 106

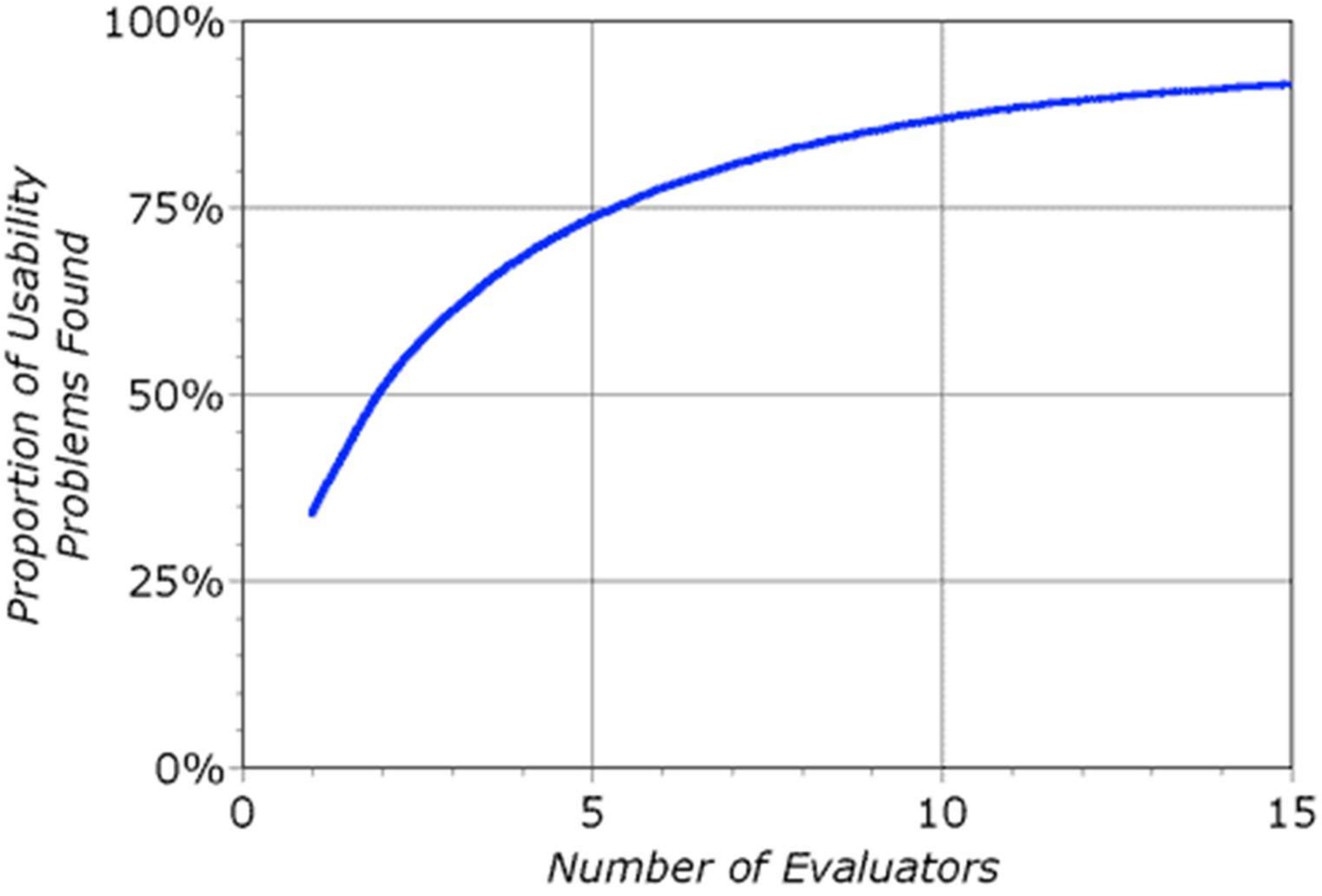
Saving A File

1. Click the file menu
2. Click “Save As..”
3. Select the directory “My Documents”
4. Enter a file name
5. Click “OK”

Recall: Heuristic evaluation

- A technique that can be applied to specifications, prototypes, user interface implementation – relatively low cost
- Heuristic evaluation involves a set of evaluators analyzing a user interface design
- Each evaluator works independently to identify problems with the user interface
- How many evaluators is enough?

Recall: Heuristic evaluation



Source: http://www.useit.com/papers/heuristic/heuristic_evaluation.html

Heuristic evaluation

- If the interface of a system does not conform to one of the heuristics it may indicate a usability problem
- Jakob Nielsen has identified 4 ways usability problems can be located:
  1. At a single interface location
  2. By comparing two interface locations
  3. In the interface structure
  4. As a missing part of the interface

Source: http://www.useit.com/papers/heuristic/usability_problems.html

Neilson’s Usability Heuristics

- 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

Severity Ratings

- Combination of three factors:
  - The frequency of problem occurrence
  - The impact of the problem if it occurs
  - The persistence of the problem
- Rating scale:

0 = There is no usability problem at all
1 = Cosmetic problem only
2 = Minor usability problem
3 = Major usability problem
4 = Usability catastrophe: imperative to fix

http://www.useit.com/papers/heuristic/severityrating.html

CW Documentation

CW Report

- Action forms for each step
  - Answers to each of the questions asked

CW & Heuristic Reports

- Cover form
  - The name of the expert
  - Date and time
  - The artifacts used in the walkthrough
- Usability problem report sheets
  - The name of expert
  - Date and time
  - Include problem description and severity rating

NOTE: Each problem report sheet corresponds to a negative answer to a question on an action form

Previous studies as evidence

- One way to reduce the cost of evaluation is to use existing studies to support design choices
- Previous studies in the area of HCI may be relevant to the user interface being designed
- **Example:** A study regarding the interaction styles for a particular group of users may indicate that a command-line interface is the most appropriate choice

Previous studies as evidence

- It is important to realize that not all previous studies are relevant!
  - Do the results of the study apply to the domain we are designing an interface for?
  - Are the participant in the study similar to the users of the interface under design?
  - etc.
- It is best if an expert is used to assess the relevance and applicability

Benyon text
chapter 10
pp. 232-246

Evaluating HCI

PARTICIPANT-BASED EVALUATION

Evaluation with Real Participants

- We can engage real primary stakeholders as participants in a variety of evaluation techniques
- These techniques, including laboratory studies and field studies, deployment studies, use qualitative and quantitative methods
- Can reveal usability and usefulness problems with early prototypes (paper) through to late-stage software (alpha/beta release)

User Participation

- Both laboratory and field studies can be costly
  - Field studies involve interruption to work day and onsite evaluation
  - Laboratory studies also interrupt participants work day and require them to travel to the lab
- Deployment studies (A/B testing, software logging, crowd source) are cheaper but difficult to control
- There are benefits and drawbacks to both which we will now discuss

Validity

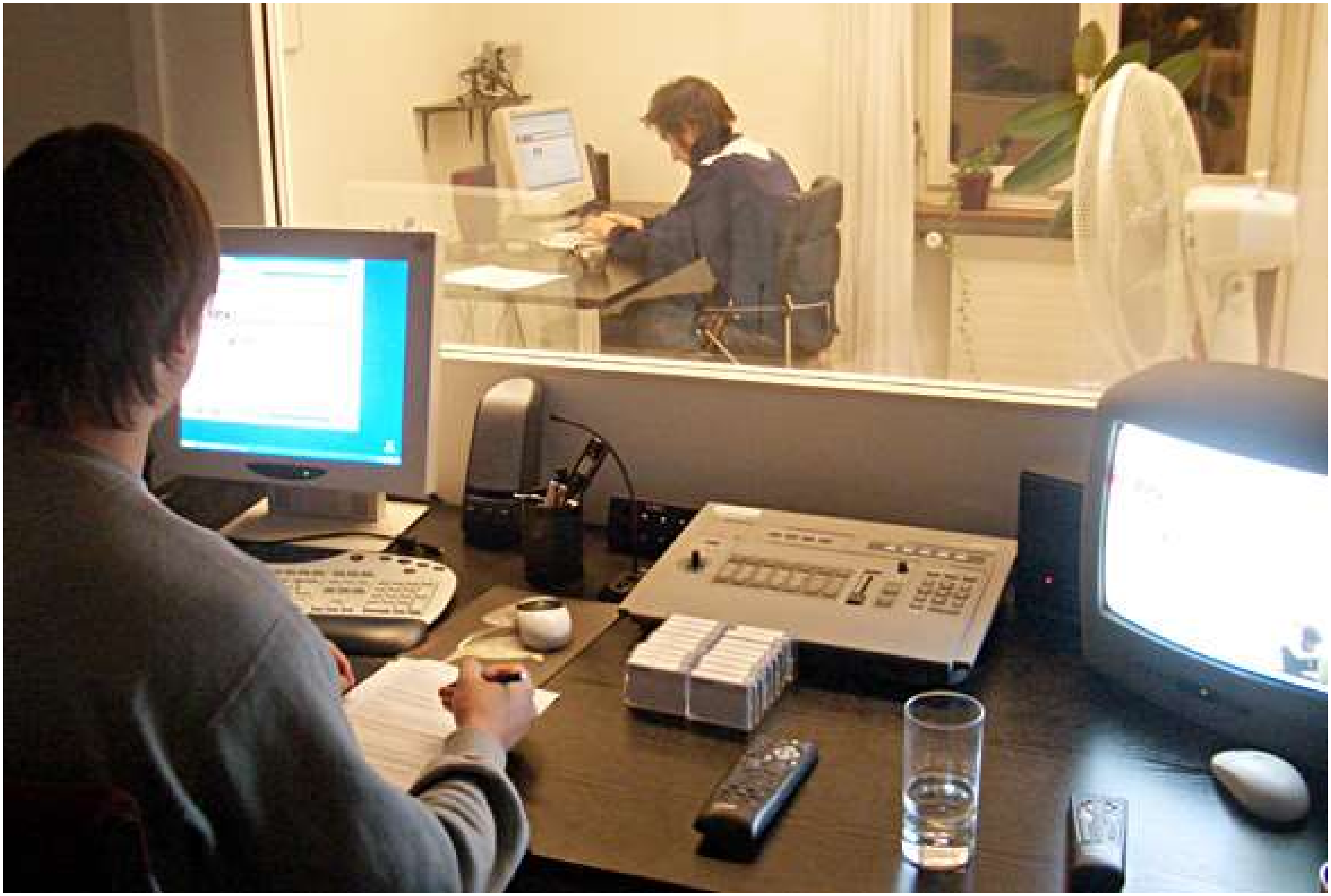
Memorize!

- External validity
  - confidence that results applies to real situations
  - usually good in natural settings
- Internal validity
  - confidence in our explanation of experimental results
  - usually good in experimental settings
- Trade-off: Natural vs Experimental
  - precision and direct control over experimental design versus
  - desire for maximum generalizability in real life situations

Laboratory Studies

Advantages

- Involves the user participating in a controlled environment – no distractions
- Can be equipped with monitoring equipment to record the study
  - Video, audio, keylogging, physiological



http://magazin.unic.com/wp-content/uploads/2011/06/SBB_Usability-Testing.jpg

Laboratory Studies

Disadvantages

- "Unnatural environment"
- Not as useful for group scenarios that involve collaboration
- Often involves non-typical people as participants (e.g. students stand in for real stakeholders)

Field Studies

Advantages

- “Natural” environment
- Can identify the effect of other work activities on use of system
- Can study long-term activities (spanning more than a few hours)

Field Studies

Disadvantages

- Environment is not controlled – constant distractions can affect the outcome
- **Example:** text messages, meetings, etc.

Lab ≠ Quant; Field ≠ Qual

- Lab:
  - Quantitative example: have participant click a button 20 times and measure speed and accuracy
  - Qualitative example: ask participant to complete a purchase on the new website and observe
- Field:
  - Quantitative example: observe through software logging the way visitors interact with wayfinding kiosk over time
  - Qualitative example: observe several individuals using the kiosk and approach them to participate in an interview

Quantitative Methods

- In an experiment it is important to:
  - – select the appropriate participants
  - – Define the independent and dependent variables
    - Independent = manipulated
    - Dependent = measured
  - – Have a clear hypothesis
  - – Have a well thought out experimental design
  - – Use appropriate statistical measures

Quantitative methods are also called empirical methods.

Confounds

- Confounding variables get in the way:
  - learning effects / order effects
  - the effects of different tasks,
  - the effects of different background knowledge, etc.
- You want to ensure a balanced and clear relationship between independent and dependent variables so that you can be sure you are looking at the relationship between them and nothing else.

Confounds are sometimes called “Threats to validity.”

Quantitative Methods

- Controlled experiments are appropriate where the designer is interested in particular features of a design, perhaps comparing one design to another to see which is better.
- In order to do this with any certainty the experiment needs to be carefully designed and run.

Summary

- Today we introduced:
  - Three goals of evaluation
  - Design principles to look for
  - Expert Evaluation
    - Cognitive Walkthrough
  - Overview of participant based studies