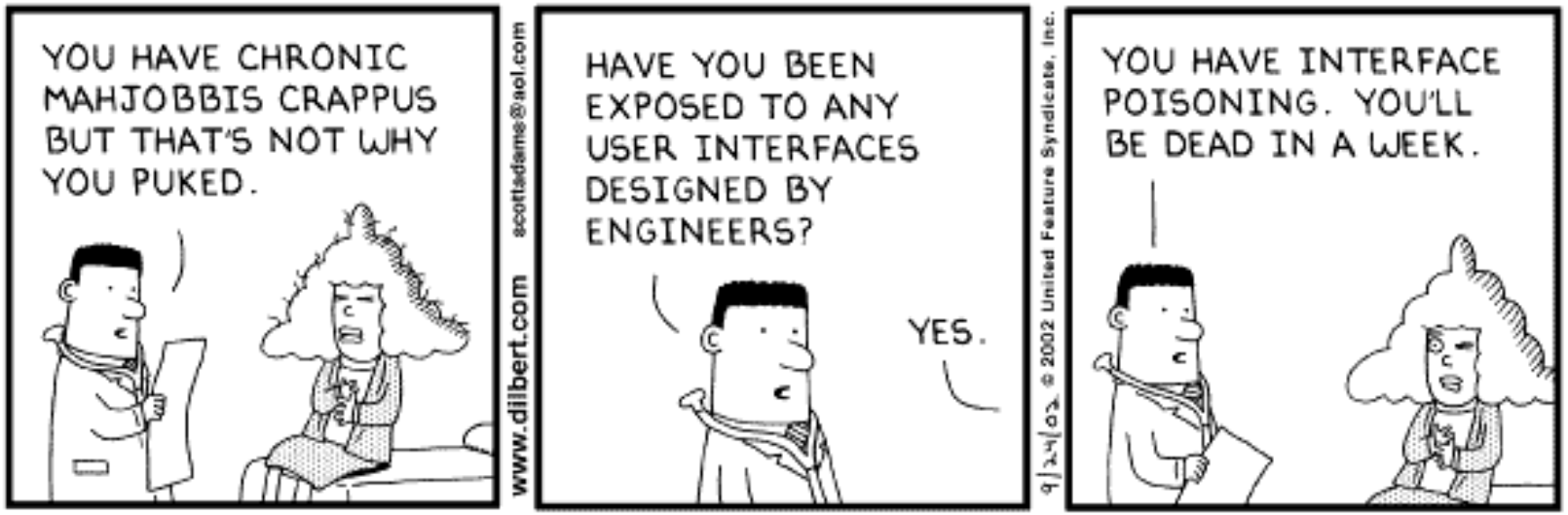
# Usability Testing



# Usability and Correctness

***Usability***
How easy is the system to use? How learnable is the system?

***Correctness***
Does the system do what it says it will do?

- Usability and correctness are two different criteria. One does not imply the other.
  - An app can be usable but not be correct.
  - An app can be correct but not usable.

# *About Face* (1995)
# Alan Cooper

- Conflict of interest
  - the people who build software are also the people who design it.
- Analogy:
  - Houses are designed by architects and built by carpenters (at least in part)
  - If carpenters designed houses, they would certainly be easier or more interesting to build, but not necessarily better to live in.
- The architect, besides being trained in the art of what works and what doesn't, is an **advocate** for the client, for the user.

# *About Face* (1995)
# Alan Cooper

- Separation of UI design and implementation in software design is currently considered a luxury
  - at least by those who haven't realized the fiscal and marketing advantages that come with professional interface design.
- Prediction:
  - a bifurcation in the industry
    - designers will design the software and engineers will build it

# Why Evaluate?

- How else to determine whether design goals have been reached?
  - In order to evaluate, criteria are needed
  - These criteria should already be established
  - Derived from the requirement analysis stage

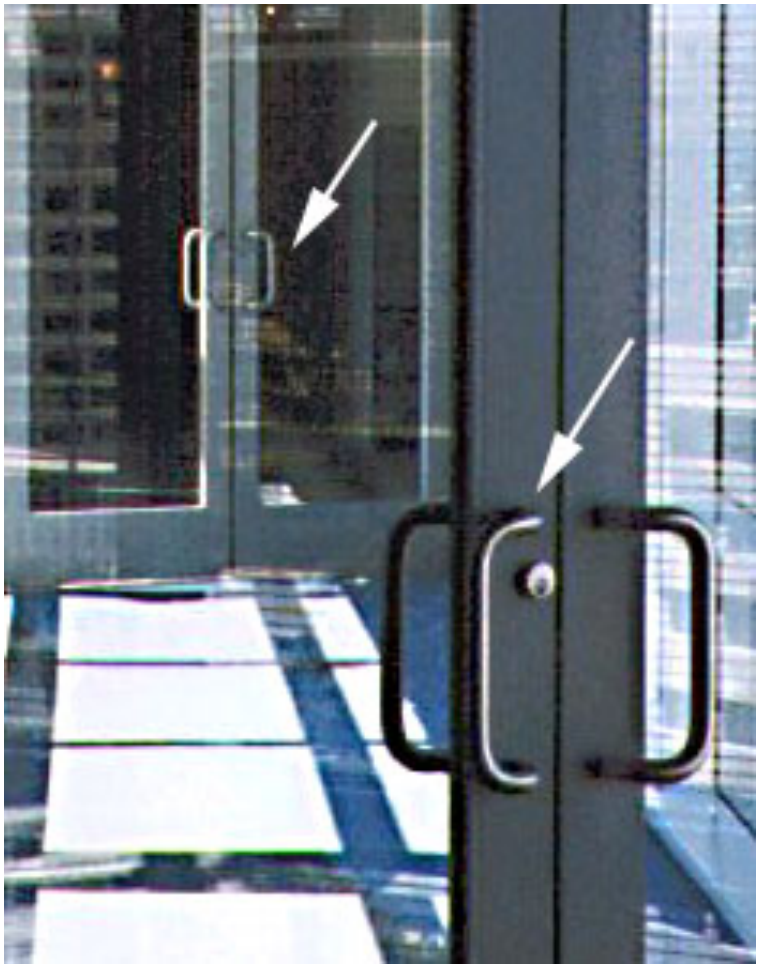
# Human interaction

- involves embodiment
  - requires use of one's own body
  - think about diversity of bodies and function
- involves cognitive processes
  - involves mental states
  - planning towards goals, intention, etc
- evaluation focuses
  - not on the software
  - not on the human
  - but on the **human using the software** (in conjoined action)

# Affordances

- "the perceived and actual properties of the thing, primarily those fundamental properties that determine just how the thing could possibly be used"

# Handles are for grabbing and pulling



# Affordances

- Some debate over its definition
- The prevalent definition in cognitive psychology is as follows:
  - all "action possibilities" latent in the environment, objectively measurable and independent of the individual's ability to recognize them, but always in relation to the actor and therefore dependent on their capabilities.
- This definiton developed by
  - Gibson, The Theory of Affordances, 1977; The Ecological Approach to Visual Perception, 1979

# Affordances

- Affordances always exist in relation to the actor and therefore are dependent on their capabilities.
  - the term is subjective
  - E.g., a set of steps which rises four feet high does not afford the act of climbing if the actor is a crawling infant.
- Norman appropriated this notion; made it dependent on the actor's goals, plans, etc
  - the term is subjective and relational
  - to afford: to provide/make available -vs- to suggest/invite

# Affordances

- perceived and/or actual properties of a thing
- features that provide clues to the operation of things
  - door plates are for pushing (pushability)
  - knobs are for turning/pulling ...etc (turnability)
- what the feature “is for” (its affordance)
- it's *ability

# Usability Goals

Preece, Rogers, and Sharp (2002) state the goals of usability are:

- effective to use (effectiveness)
- efficient to use (efficiency)
- safe to use (safety)
- have good utility (utility)
- easy to learn (learnability)
- easy to remember how to use (memorability)

Thought question:
Are these goals necessarily compatible with one another?

# Types of Evaluation

- Formative
  - done at different stages of development
  - done to provide feedback for design
- Summative
  - done to evaluate the final product

# Why Evaluate (Formative)?

- Identify problems before application is deployed, not after
  - "deployed" can mean shipped for commercial products or otherwise made available (for other products)
- Characterize problems
  - correct misconceptions by developers (developers' assessment of the problems may differ from the actual problems that exist)
  - prioritize problems and focus subsequent time and energy

# What is Usability?

Definition #1 `http://whatis.techtarget.com/`

- Usability is the measure of a product's potential to accomplish the goals of the user.
  - term used in relation to software applications and Web sites
  - term can also be used in relation to any product that is employed to accomplish a task (for example, a toaster, a car dashboard, or an alarm clock).
  - Some factors used in determining product usability are ease-of-use, visual consistency, and a clear, defined process for evolution.

- Note the use of the word *potential*
- What is the status of the software application?
  - It doesn't accomplish goals (i.e., it doesn't have *agency*); rather individuals use it to accomplish goals

# What is Usability?

Definition #2 `http://www.affectus.se/publicerat/usabilityinsp/`

- *Usability* basically refers to:
  - how easy it is for users to learn a system,
  - how efficiently they can use it once they have learned it, and
  - how pleasant it is to use.
  - Also, the frequency and seriousness of user errors are normally considered to be constituent parts of usability.

# Categorization of Usability Problems
## Critical Incident Approach

- Types of problems:
  - Relevance/Effectiveness/Usefulness
  - Efficiency
  - Attitudinal
  - Learnability
- Severity
- Frequency

# Categorization of Usability Problems

## Nielson's Usability Severity Ratings

**0** Not a usability problem at all

**1** Cosmetic problem only - need not be fixed unless extra time is available on project

**2** Minor usability problem - fixing this should be given low priority

**3** Major usability problem - important to fix, so should be given high priority

**4** Usability catastrophe - imperative to fix this before product can be released

# Categorization of Usability Problems

**Frequency Ratings**

**0** Problem did not occur

**1** This problem occurs rarely- only once/under very unusual circumstances

**2** This problem occurs occasionally - under less commonly occurring conditions

**3** This problem occurs often during common tasks

**4** Problem occurs every time under all tested conditions

# Aspects of Usability: REAL

- The acronym REAL summarizes four aspects of usability:
  - **R**elevance:
    - how well does the system serves the users' needs?
  - **E**fficiency:
    - how efficiently can the users carry out their tasks using the system?
  - **A**ttitude:
    - what are the users' subjective feelings towards the system?
  - **L**earnability:
    - how easy is the system is to learn initially?
    - how well do the users remember the skills over time?

# **R**elevance

- Measurement criteria:
  - Number of good and bad features recalled by users
  - Number of available commands not invoked by users
  - Number of available commands invoked by users
  - Number of times users need to work around a problem
  - Percentage of task completed

# Efficiency

- Measurement criteria:
  - Time to complete a task
  - Percentage of task completed
  - Percentage of task completed per unit time (speed metric)
  - Time spent in errors
  - Number of commands used
  - Frequency of help and documentation use
  - Time spent using help or documentation

# Efficiency

- Measurement criteria (con't):
  - Number of repetitions of failed commands
  - Number of runs of successes and of failures
  - Number of times interface misleads user
  - Number of times user needs to work around a problem
  - Number of times the help facilities solve the user's problem

# **A**ttitude

- Measurement criteria:
  - Percentage of favourable/unfavourable user comments
  - Number of good and bad features recalled by users
  - Number of users preferring the system
  - Number of times user loses control over the system
  - Number of times the user is disrupted from a work task
  - Number of times user expresses frustration or satisfaction

# Learnability

- Measurement criteria:
  - Ratio of success to failures (over time)
  - Time spent in errors
  - Percentage or number of errors
  - Number of commands used
  - Frequency of help and documentation use
  - Time spent using help or documentation
  - Number of repetitions of failed commands
  - Number of runs of successes and of failures

# Learnability

- Measurement criteria (con't):
  - Number of available commands not invoked by users
  - Number of features or commands that can be remembered after a test
  - Proportion of users using efficient strategies compared to those using less efficient strategies
  - Number of logical errors made

# Useful vs. Usable

- Useful – the quality of having use
  - the application can be used to accomplish something
- Usable – the quality of having potential to accomplish the goals of the user
  - the user's goals are the primary focus
  - the application serves (to varying degrees) to help the user accomplish those goals
- How can an application be useful but not usable?
- Does it make sense to say an application is usable but not useful?

# Low Fidelity Prototyping

- Low fidelity prototypes are
  - hand sketches and scenarios
    - e.g., storyboards, “scene by scene”
- They focus on the design, not on the interface mechanics

# Low Fidelity Prototyping

- Advantages of low fidelity prototypes:
  - can be used to conduct usability testing
    - thus, can perform usability testing even before any code has been written
  - can be used early in the development process
  - easy to modify
  - can be used to (re)define requirements
- Disadvantages:
  - _____

# The DECIDE framework

- The DECIDE framework provides the following steps to guide evaluation:
  - **D**etermine the overall goals that the evaluation addresses
  - **E**xplore the specific questions to be answered
  - **C**hoose the evaluation paradigm and techniques
  - **I**dentify practical issues
  - **D**ecide how to deal with the ethical issues
  - **E**valuate, interpret, and present the data

# The DECIDE framework
# **D**etermine the overall goals

- What are the high level goals of the evaluation?
- Examples:
  - Check that evaluators have understood the users' needs
  - Ensure that the final interface is consistent
  - Determine how to improve the usability of a user interface

# The DECIDE framework
# **E**xplore specific questions

- Break down overall goals into relevant questions
- Overall goal: Why do customers prefer paper tickets to e-tickets?
- Specific questions:
  - What is the customer's attitude?
  - Do they have adequate access to computers?
  - Are they concerned about security?
  - Does the electronic system have a bad reputation?
  - Is its user interface poor?

# The DECIDE framework
# **C**hoose paradigm and techniques

- Practical and ethical issues might be considered
- Factors:
  - Cost
  - Timeframe
  - Available equipment or expertise
- Compromises may have to be made

# The DECIDE framework
## Identify practical issues

- Important to do this before starting
- Find appropriate users
- Decide on the facilities and equipment to be used
- Schedule and budget constraints
- Prepare testing conditions
- Plan how to run the tests

# The DECIDE framework

## **D**ecide on ethical issues

- Studies involving humans must uphold a certain code
- Privacy of subjects must be protected
- Personal records must be kept confidential
- Exact description of the experiment must be submitted for approval

# The DECIDE framework

## **E**valuate the data

- Should quantitative data be treated statistically?
- How to analyze qualitative data?
- Issues to consider:
  - Reliability (consistency)
  - Validity
  - Biases
  - Scope
  - Ecological validity

# Four Paradigms for Evaluation

1. "Quick and Dirty"
2. Field studies
3. Predictive evaluation
   - Heuristic evaluation
   - Cognitive walkthroughs
4. Usability testing
   - Thinking Aloud Protocol
   - Co-discovery method

# "Quick and Dirty" Evaluation

- How it takes place:
  - Individual(s) use the application, either in a lab or in his or her natural environment
  - Evaluator is present, but has minimal control
- What is obtained:
  - Sketches
  - Quotes
  - Descriptive reports
- Feedback is incorporated into design process

# "Quick and Dirty" Evaluation

Issues:

- User might share opinions or impressions of appearance
  - How should these be interpreted?
- A relatively small set of interactions are elicited
  - Are they representative of the future interactions that will take place with that user (or others like him or her?)
  - Factors: evaluator is present; user's lack of familiarity
  - Are they representative of the future interactions that will take place for other users?
- What is being evaluated?
  - is it the user interface, or the interactions that the user interface affords?
- [Recall: concept of affordance]

# "Quick and Dirty" Evaluation

Advantages:

- User-centered
- Practical
- Provides feedback relatively quickly

Disadvantages:

- Users are expected to behave naturally
- Evaluators take minimum control

# Field Studies

- How it takes place:
  - analysts visit users in their natural environment.
  - analysts assess how the users' needs are being met and whether there are problems; try to identify design opportunities
- What is obtained:
  - Qualitative descriptions
  - Quotes
  - Sketches
  - Anecdotes
- Often used early in design to check that users' needs are met or to assess problems or design opportunities
- Conducted in the user's natural environment
- Evaluators try to develop relationships with users

# Predictive Evaluations

- How it takes place:
  - Expert evaluators use practical heuristics and practitioner expertise to predict usability problems
  - Usually conducted in a lab
  - Doesn't involve users
- What is obtained:
  - Reviewers provide a list of problems, often with suggested solutions

# Predictive Evaluations

- Two predictive evaluation techniques:
  - Heuristic evaluation
  - Cognitive walkthroughs

# Heuristic Evaluation

- Heuristic evaluation was developed by Jakob Nielsen
  - see: Nielson, Jakob (n.d.) "How to Conduct a Heuristic Evaluation. `http://www.useit.com/papers/heuristic/heuristic_evaluation.html`
- It is a technique in which experts evaluate whether user interface elements conform to a set of usability principles or not.
  - The usability principles are described as *heuristics*
  - Heuristics bear a close resemblance to design principles and guidelines

# Heuristic Evaluation

Ten usability heuristics:

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Help users recognize, diagnose, and recover from errors

# Heuristic Evaluation

Ten usability heuristics:

6. Error prevention
7. Recognition rather than recall
8. Flexibility and efficiency of use
9. Aesthetic and minimalist design
10. Help and documentation

# Cognitive walkthroughs

- "Cognitive walkthroughs involve simulating a user's problem-solving process at each step in the human-computer dialog, checking to see if the user's goals and memory for actions can be assumed to lead to the next correct action"
  (Nielsen and Mack, 1994).
- The focus is on evaluating design for ease of learning

# Cognitive walkthroughs

- Primary Steps:
  - Characteristics of typical users are identified
  - Designers and evaluators meet, walk through the action sequences for each task and try to answer the following questions:
    - Will the correct action be evident to the user?
    - Will the user notice that the correct action is available?
    - Will the user know from the feedback whether they made a correct choice?

# Goals of Usability Testing

- Does the application meet the qualitative usability goals?
- Does the application meet the quantitative usability goals?

Adapted from Mayhew, Deborah J. (1999) The Usability Engineering Lifecycle

# Examples of Qualitative Goals

- E.g. 1: That the design support users working in a high-interrupt environment
  - One way to achieve this goal: that the design provide lots of context information on screen to remind users where they are when they get distracted
- E.g., 2: That the design support very infrequent users of a very complex task
  - One way to achieve this goal: that the design be self-explanatory, easy to learn and to remember.

Adapted from Mayhew, Deborah J. (1999) The Usability Engineering Lifecycle

# Examples of Quantitative Goals

- E.g. 1: That experienced users should take no longer than 15 seconds on average to address an email.
  - Experience defined as users who have performed the task five times in a training session

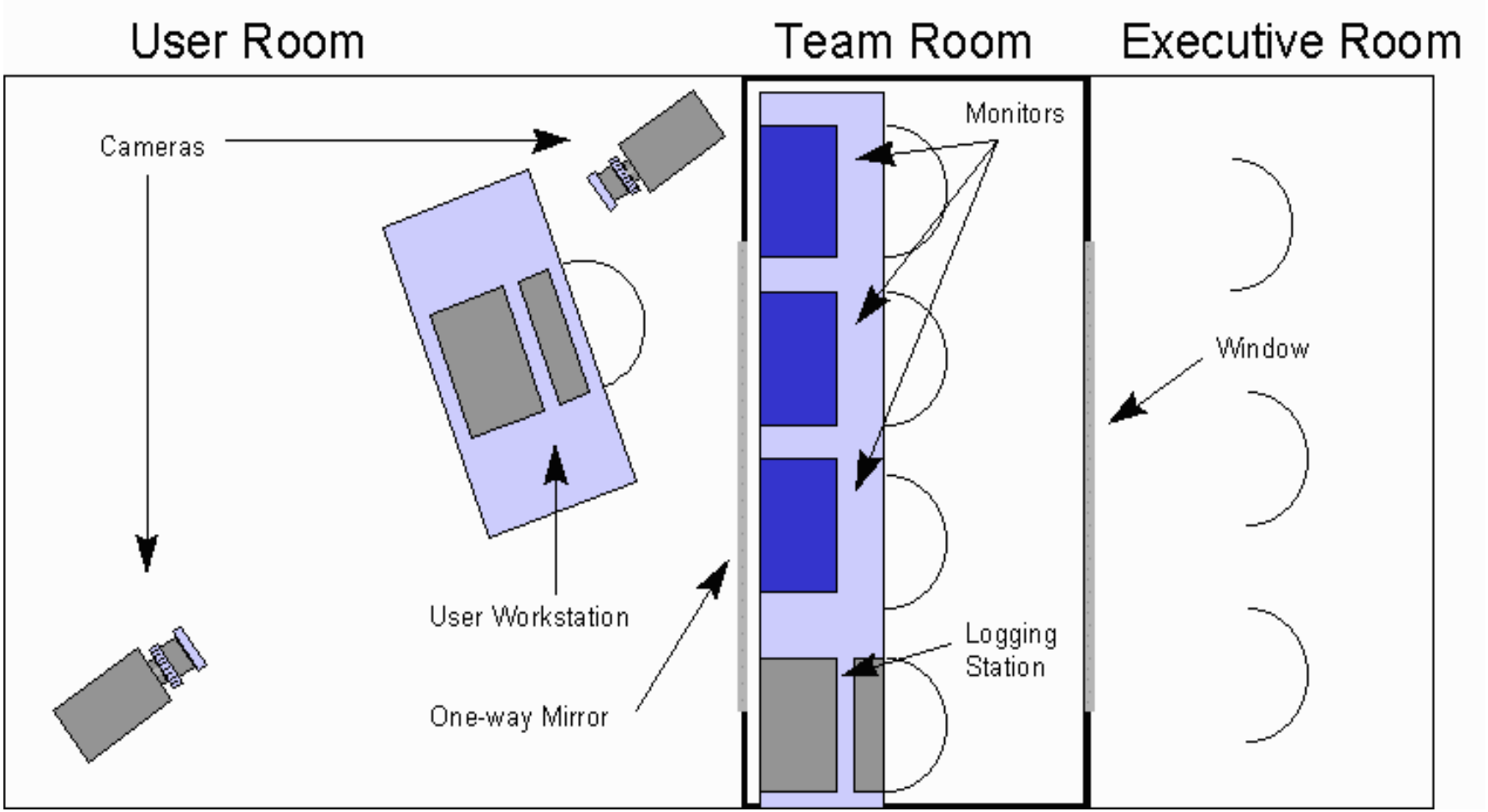
- E.g. 2: That novice users should take no longer than three minutes to complete the registration input form.
  - Novice defined as first-time users

Adapted from Mayhew, Deborah J. (1999) The Usability Engineering Lifecycle

# Usability Lab

http://www.sun.com/usability/

# Usability Lab




http://www.ulabs.com/images/lab.gif

# Usability Testing

- How it takes place:
  - Users carry out one or more pre-determined tasks in a usability lab
  - Evaluators present but not visible
- What is obtained:
  - Users' opinions (collected by questionnaire or interview)
  - Usability test participants are encouraged to think aloud and voice their every opinion.
  - Reports of performance measures: number and type of errors, time-to-completion and others
  - Qualitative; Quantitative
- Feedback is incorporated into design process
  - Changes are made to the application or site based on the findings of the usability tests.

# Usability Testing

- There should be:
  - specific questions that the usability testing is designed to answer.
  - ***Usability testing has specific objectives.***
  - Design issues to be studied need to be determined in advance.
- The tasks are chosen in order to measure specifically chosen attributes of the interaction.

Adopted from http://searchwebservices.techtarget.com/

# What to User Test?

The tasks are chosen in order to measure specific attributes:

- The product's ease-of-use
- Ease of system learning
- Retention of learning over time
- Time required to complete task(s); Speed of task completion
- The user's perception of the experience; Subjective user satisfaction
- Conformance with a requirement
- Conformance with guidelines for good design
- Error rates

Galitz, W. O., (2002) The Essential Guide to User Interface Design, 2nd Edition, Wiley Computer Publishing, New York, NY.

# Which Attributes to Evaluate?

- Depends on outcomes of previous evaluations
- Depends on which phase of the development cycle
  - What can be changed?
    - Fundamental design decisions (e.g., the conceptual model of the task)
    - The way the design decisions have been implemented (e.g., Layout, Fonts, etc)

# Usability Testing

Advantages:

- It is impossible to predict usability from appearance, just like it is impossible to judge a person's personality on appearance.
- Casual "feel good" feedback is inadequate.
- Formal testing is often the only way problems are identified pre-release. Problems found once a product is released are usually not fixed unless they are really severe.

# Usability Testing

Advantages:

- Users, designers, programmers have different models
- The designer's intuition is not always correct
- Design standards and guidelines are not sufficient
- Usability testing leads to competitive advantages and reduced support costs

# Usability Testing

Advantages:

- Applied approach based on experimentation
- Evaluators are strongly in control

Disadvantages:

- Time consuming, expensive
- Used when a prototype or a product is available
- Evaluators are strongly in control

# Usability Testing

What do evaluators say to subjects?

- Give a brief explanation that the participant's involvement is to solicit user feedback.
- Provide *instructions* as to the user's task but not *explanations* of the software.
- Reassurance that any problems are the fault of the software.
- In real-world situations, explain confidentiality agreement, liability legalities, and that participant is free to leave at any time (and still get paid).

# Thinking Aloud Protocol

- During a usability test, instruct participants to verbalize their thoughts.
- The evaluator's goal is:
  - to understand their mental model of the system and the tasks
  - to understand where the subjects have trouble in understanding and using the system
- Prompt participants by asking direct questions about the software

# When to ask questions?

- Need to balance:
  - interruption to the user's task flow
    - interruption may affect the flow of the task
  - short-term memory
    - users may forget
- In general:
  - ask the question after the completion of the task, unless you are more worried about the participant forgetting their current thought process (i.e., interrupt and ask right away).

# Co-discovery Method

- Two participants perform tasks together, but one participant is assigned the mouse and the other the keyboard.
- They are instructed to work together to accomplish a common goal using the product.

- The allocation of input devices increases the amount of communication between the subjects
- Their communication provides information about their thought processes