

USABILITY ENGINEERING

Néstor J. Rodríguez
Catedrático Asociado
Departamento de Ingeniería
Eléctrica y Computadoras

PRESENTATION OVERVIEW

- Why Usability Engineering?
- What is Usability?
- Usability Heuristics for User Interfaces
- Usability Testing
- Other Usability Assessment Methods

2

Your Best Guess is not Good Enough

- It is not possible to design an optimal user interface just by giving it your best try.
- Users have potential for unexpected misinterpretations of interface elements.
- Users have potential for performing their job in a different way than you imagine.
- Designers are not users

Prescription:

- understand the users and their tasks
- validate the design with usability assessment methods

3

The User is Always Right

- If users have problems with an aspect of the interface is not because:
 - ◆ they are stupid
 - ◆ they did not try a little harder

Prescription:

- do not blame the users for usability problems
- be humble, acknowledge the need to modify the original design to accommodate the user's problems

4

The User is not Always Right

- Users often do not know what is good for them.
- Users have a very hard time predicting how they will interact with systems with which they have no experience.
- Users will often have divergent opinions when asked about details of user interface design.

Prescription:

- do not design user interfaces exclusively based on what the users say they would like
- usability test user's preferences

5

Usability Has an Economic Impact

- Reduces training costs
- Products with good usability sell better
- Users can do more productive work per hour

A Very Good Example:

Changing a screen from full to half full saved a telephone company 40 million dollars per year.

6

USABILITY ATTRIBUTES

- Learnability
- Efficiency
- Memorability
- Errors
- Satisfaction

7

Learnability

- The system should be easy to learn
- Users should rapidly be able to do some productive work with the system

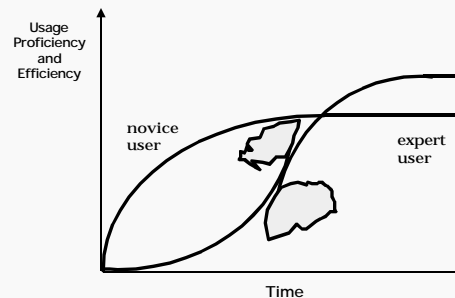
8

Efficiency

- The system should be efficient to use
- The user should eventually achieve a high level of productivity with the system

9

The Learning Curve



10

Memorability

- The system should be easy to remember
- Casual users should be able to use the system without having to learn everything all over again

11

Errors

- The system should have a low error rate
- Users should easily recover from errors
- Catastrophic errors should not occur

12

Satisfaction

- The system should be pleasant to use
- The system should be one that the users like to use

13

USABILITY ASSESSMENT METHODS

- Heuristic Evaluation
- User Testing
- Questionnaires and Interviews

14

HEURISTICS EVALUATION

- Based on well-known principles for user interface design
- Detects usability problems before user testing
- Does not provide solutions
- Discount usability assessment method

15

Jakob Nielsen's Usability Heuristics

- Simple and Natural Dialogue
- Speak the User Language
- Minimize User Memory Load
- Consistency
- Feedback
- Clearly Marked Exits
- Shortcuts
- Good Error Messages
- Prevent Errors
- Help and Documentation

16

SIMPLE AND NATURAL DIALOGUE

- User interfaces should be simplified as much as possible
- The ideal - present exactly the information the user needs (and no more) at exactly the right time and place
- User interfaces should match the users' task in as natural a way as possible.
- Minimize navigation through the interface

17

Mapping

Is the mapping between computer concepts and user concepts as simple as possible?

Is the navigation through the user interface minimized?

Is the information provided exactly the information the user needs (and no more) at exactly the time and place where it is needed?

Are the information objects and operations accessed in a sequence that matches the way users will most effectively and productively do things?

18

Graphic Design

Is the information that will be used together displayed close together or at least in the same screen?

Wisdom: Grouping could be achieved with dividing lines, boxes, colors, etc.

Do the most important dialogue elements stand out?

Wisdom: Blinking should only be used in extreme cases (it is distracting and annoying).

19

Colors

Does the interface look like an angry fruit salad of wildly contrasting, highly saturated colors?

Does the interface use more than 7 colors?

Could the interface be used without colors?

Wisdom: Colors should be used to categorize, differentiate and highlight.

20

Less is More

Is irrelevant information presented in the interface?

Is less important information left for auxiliary screens?

Are too many choices provided?

Wisdom: Use training wheels approach - provide multiple nested levels of increased complexity (for novice users to expert users).

21

SPEAK THE USERS' LANGUAGE

- The terminology should be based on the users' language

22

Verbal Language

Is the terminology used based on the users' language (not on system oriented terms)?

Are words in nonstandard meanings used?

Are interaction messages stated from the users' perspective?

Are there restrictions on the names the users assign to objects (i.e. limits on the number of characters that can be specified)?

Wisdom: Use vote winning terms (words).

Wisdom: Allow for rich use of synonyms in interpreting command languages and in documentation indexes.

23

Graphical Language

Is there a good mapping between the computer display of information and the user's conceptual model of the information (the one million dollar question)?

Wisdom: When using metaphors care should be taken to present the metaphor as a simplified model of a more detailed conceptual model of the system and not as a direct representation of the system.

24

MINIMIZE THE USERS' MEMORY LOAD

- Users should not be forced to remember information or codes to interact with the computer
- Users should be allowed to choose from items on a dialogue element
 - menus
 - dialogue boxes options
 - palettes
 - default values

25

Minimizing Memorizing

Wisdom: In general, people have much easier time at recognizing than recalling.

Are dialogue elements displayed to the users so that they can choose from them or edit them (menus, palettes, dialogue boxes, default values)?

Are generic commands used?

Wisdom: The system should be based on a small number of pervasive rules that apply throughout the user interface (cut & paste, point & click, drag & drop, etc.)

26

CONSISTENCY

- Users should not have to wonder whether different words, situations or actions mean the same thing
- The same information should be presented in the same place on all screens and dialog boxes
- Follow the user interface standard

27

Maintaining Consistency

Wisdom: Users feel more confident in using the system if they know that the same command or the same action will always have the same effect.

Is the same information presented in the same location on all screens and dialogue boxes? Is it formatted in the same way?

Is the interface standard being followed?

28

FEEDBACK

- The system should continuously inform the user about what it is doing

29

Providing Good Feedback

Does the system continuously inform the user about what it is doing and how it is interpreting the user's input?

Is feedback expressed in abstract and general terms (like in command line interfaces)?

Wisdom: Feedback should restate and rephrase the user's input to indicate what is being done with it (i.e. when overwriting a file).

Is feedback provided in case of a system failure?

30

Feedback Persistence

Is the degree of persistence of the feedback appropriate?

Is feedback provided according to response time of operations?

Wisdom:

- ◆ No special feedback is needed for delays less than 1 second.
- ◆ Feedback for events lasting more than 10 seconds should have a high degree of persistence (use percent done indicators).
- ◆ Do not use percent done feedback for operations taking between 2 to 10 seconds.

31

CLEARLY MARKED EXITS

- The system should provide the user an easy way out of as many situations as possible
 - undo
 - cancel
 - ...

32

Getting out of Undesired Situations

Does the system provides the user an easy way out of as many situations as possible (cancel, undo, etc.).

Do the exit and undo mechanisms visible (not a special code or an obscure combination of keys)?

Wisdom: Users will make errors no matter what else is done to improve the interface, and one should therefore make it as easy as possible to recover from these errors.

Wisdom: In general, interfaces should show a high degree of responsiveness, to the extent that paying attention to the user new actions should get higher priority than finishing the user's old actions (like stopping a printer).

33

SHORTCUTS

- Provide features that allow experienced users to speed up interaction
 - function keys
 - double clicking
 - repeat operation
 - macros

34

Speeding up the Interaction

Are users allowed to jump directly to the desired location in large information spaces?

Are users allowed to reuse their interaction history (i.e. list of documents previously accessed)?

Wisdom: The following are shortcuts to consider:

- function keys
- templates
- wizards
- double clicking
- type-ahead
- click-ahead
- repeat last command
- repeat last operation
- working groups
- history list
- list files by date
- default values

35

GOOD ERROR MESSAGES

- Error messages should:
 - be phrase in clear language
 - avoid obscure codes
 - be precise
 - help the user solve the problem
 - not intimidate the user
 - not blame the user
 - avoid abusive terms such as fatal, illegal, etc.

36

PREVENT ERRORS

- Avoid putting the user in error prone situations

Are there mechanisms to prevent errors due to:

- spelling names of objects or operations
- case sensitive text
- operations with serious consequences
- use of modes
- other situations

37

HELP AND DOCUMENTATION

- Provide on-line help
- Provide useful manuals
- Provide good task-oriented search and look up tools

38

Good Help and Documentation

Wisdom:

The fundamental truth about documentation is that most users simply do not read the manuals.

The quality of help texts is far more important than the mechanisms by which those texts are accessed.

Does the system provides documentation?

Does the system provides online help?

Is a "minimal manual" provided (one that only gives whatever information is absolutely necessary in order for the users to get started using the system with common tasks)?

39

USABILITY TESTING WITH USERS

- It is the most fundamental usability method
- It is irreplaceable
- It provides direct information about how people use computers
- It detects interaction problems with the interface

40

TESTING ASPECTS

- Test Goals
- Test Plans
- Test Tasks
- Test Users
- Ethical Aspects of Tests with Human Subjects

41

STAGES OF A TEST

- Preparation
- Introduction
- The test itself
- Debriefing

42

Test Plan

- Purpose
- Test objectives
- User profile
- Methodology
- Task list
- Test environment
- Data to be collected
- Results report

Based on Jeffrey Rubin's Handbook of Usability Testing

43

Purpose

- Describe in general terms the reason for conducting the test

44

Objectives

- Indicate what you want to accomplish with the test

Good examples:

- Compare the effectiveness of novice users vs. expert users of interface A.
- Compare the user effectiveness and user satisfaction between version A and version B of an interface.
- Determine navigation problems of interface B.
- Determine if the users of interface D are able to use buttons Y and X, and if they understand their functionality.

Bad example:

- Determine if a product is usable.

45

User Profile

- The users should be selected according to the objectives of the test.
 - computer experience
 - domain experience
 - demographics
- When testing an application the users should be representative of the eventual users of the application.
- The number of users needed for the test depends on:
 - degree of confidence in the results required
 - the available resources to conduct the test
 - the availability of the type of users required
- Most usability problems can be determine with four to eight users.

46

Methodology

Provide an overview of each facet of the test from the time the users arrive until the time they leave.

- greeting the user
- consent and background forms
- orientation script
- pretraining
- test design
- data to be collected
- post test questionnaires
- debriefing

47

Test Design

- Between-Subjects Design - each module of the application is tested with a unique set of users
- Within-Subjects Design - each part of the test is done with the same group of users
 - transfer of learning effect
 - the order of tasks need to be balanced out

48

Task List

- The tasks should satisfy the test objectives.
- The task list should indicate:
 - brief description of each task
 - materials and machine states (screens) required
 - description of successful completion of each task
 - minimum and maximum times for completing the tasks (optional)
- The first tasks should be easy to accomplish (hard ones should be last)

49

Test Environment

- Indicate the environmental conditions that need to be set up or simulated
- Indicate the equipment needed to conduct the test

50

Data to be Collected

- Performance measures
 - time to complete each task
 - number and percentage of tasks completed
 - count of object selections
 - count of error commission
 - count of incorrect menu choices
- Preference measures
 - ease of learning
 - ease of use overall
 - usefulness of application
 - how well the application matches expectations
 - one prototype vs. another prototype

51

Results Report

- Indicate how the collected data will be presented
- Indicate how you intend to communicate the results
 - informal meeting
 - formal report
 - formal presentation

52

Orientation

- Introduce yourself and any other observer. (Offer refreshments.)
- Explain why the users are here.
- Reaffirm the users that their participation will be remain anonymous.
- Describe the lab environment.
- Explain what is expected of the users.
- Assure the participants that they are not being tested.
- Explain any unusual requirement such as "thinking aloud".
- Mention that it is OK to ask questions.
- Ask is the participants have any question before starting the test.
- Refer to any forms that need to be completed.

53

Tips for Conducting the Test

- Treat each new participant as an individual.
- Be aware of the effect of your voice and body language.
- Don't make any comments to the participants about their performance on the test.
- If the situation arises, indicate the participants that there is no right or wrong response.
- Don't rescue the participants when they struggle. (Yet, don't let them struggle for too long.)
- Assist the participant only as a last resort (especially when frustration takes over).
- Avoid interruptions during the test.
- Observers should be out of sight and silent.
- Take notes (disimuladamente) on interesting situations. Talk about them during debriefing.

54

What NOT to Say to Participants (J. Rubin)

10. Saying, "Remember, we're not testing you," more than three times.
9. Are you familiar with the term "outlier"?
8. No one's ever done that before.
7. HA! HA! HA!
6. That's impossible! I didn't know it could go in upside down!
5. Could we stop for awhile-watching you struggle like this is making me tired.
4. I didn't really mean you could press any button.
3. Yes, it's very natural for observers to cry during a test.
2. Don't feel bad, many people take 15 or 16 tries.
1. Are you sure you have used computers before?

55

QUESTIONNAIRES AND INTERVIEWS

- Indirect methods of evaluating usability of interfaces
- Can reveal what features of an interface the users like or dislike
- Method mostly used after user testing

56

Other Issues

- Conduct a Pilot Test
- After the test, let the users know about unmentioned things such as "Wizard-of-Oz technique".

57