Chapter 10
Interactive Systems And Usability

Organizational
Requirements
Engineering

Prof. Dr. Armin B. Cremers
Sascha Alda
Overview

- Introduction: What is usability?
- Why is usability an important non-functional requirement?
- Ways to usability
- GUI Design for Interactive Systems
  - General
  - Design Guidelines
  - Forms
  - Navigation within an application
Interactive Systems

- An interactive system allows for a dialog between the computer and the user
- Examples:
  - Web browser
  - Automated Teller Machines (ATM)
  - Office tools
  - Booking terminals
  - Information desks
- Most typical software system that can be modeled by use cases (→ functional requirements)
- Non-functional requirements are also important
Usability... 
...as a non-functional requirement

- Typical non-functional requirements
  - reliability
  - maintainability
  - deployability
  - Configurability...

- **Usability** is of major importance
  - “When the interface is well designed, it is comprehensible, predictable, and controllable; users feel *competent*, satisfied, and responsible for their actions” (Shneidermann, 1996)
Usability Guidelines

- Many guidelines can be found

- **ISO 9241: Ergonomic requirements for office work with visual display terminals (1996-99)**
  - International standard for the interaction between humans and computer
  - Provides requirements and recommendations relating to the attributes of the hardware, software and environment that contribute to usability, and the ergonomic principles underlying them
  - Authoritative source of reference
  - Designers without usability experience have great difficulty applying these types of guidelines

- **ISO/IEC FCD 18021: Information Technology - User Interface for mobile tools (2001)**
  - Contains user interface specifications for PDA's with a data interchange capability with corresponding servers.
Usability...
...ISO 9241

Part 11
Usability in general

Part 10 / 12
Guides to design dialogues/
presentation of information

Part 13 - 17
Techniques to design dialogues
Product attributes

Analysis of Use Context

Design Guides

general special
Usability in ISO 9241

  - deals with general ergonomic principles which apply to the design of dialogues between humans and information systems:
    - Criteria: suitability for the task, suitability for learning, suitability for individualization, conformity with user expectations, self descriptiveness, controllability, and error tolerance.
- Part 12: Presentation of information (1998)
  - contains recommendations for presenting and representing information on visual displays.
- Part 14: Menu dialogues (1997)
- Part 15: Command dialogues (1997)
- Part 16: Direct manipulation dialogues (1999)
- Part 17: Form filling dialogues (1998)
Measuring Usability... 
...according to ISO 9241-11

Usability according to ISO 9241-11:
Usability can be measured by the extent to which users reach their goals dependent on values like effectiveness, efficiency, and satisfaction.

- **Effectiveness**
  - (Reaching a Goal)
  - accuracy and completeness with which users achieve specified goals.

- **Efficiency**
  - (Effort to reach the goal)
  - resources expended in relation to the accuracy and completeness with which users achieve goals.

- **Satisfaction**
  - (Positive attitude to the use of the product)
  - The comfort and acceptability of use

Usability is more than nice looking windows.

[http://www.klubok.net/pageid299.html](http://www.klubok.net/pageid299.html)
Usability... is context-dependent

- Usable products may be designed by incorporating product features and attributes known to benefit usability.
  - Not sufficient to ensure that the product will be effective, efficient and satisfying in use
- Necessary to measure the performance and satisfaction of users working with a product.
- Measurement of usability is particularly important in view of the complexity of the interactions between
  - the user
  - task characteristics
  - context of use
- Different levels of usability for the same product when used in different contexts
Usability...  
...is context-dependent

- **Context of use**: The users, goals, tasks, equipment (hardware, software and materials), and the physical and social environments in which a product is used.

- Environment
  - Large interfaces vs. smaller ones
  - Mobility concerns

- User (person who interacts with the product)
  - Skills
  - Former Systems (similar systems)
  - Expectations, Goals (an intended objective)
  - Tasks
  - Times of use: Examples, Online Help, Shortcuts
Usability framework for measuring usability

Intended objectives

Goals

Effectiveness

Efficiency

Satisfaction

Usability Measures

Outcome of Interaction

Actual Value

Target Value

Intended objectives defines

Usability framework for measuring usability

Context of Use

User

Task

Equipment

Environment

Product

Goals

Effectiveness

Efficiency

Satisfaction

Usability Measures

Actual Value relation Target Value

Intended objectives

Usability framework for measuring usability

Context of Use

User

Task

Equipment

Environment

Product

Goals

Effectiveness

Efficiency

Satisfaction

Usability Measures

Actual Value relation Target Value

Intended objectives defines
When specifying or measuring usability, the following information is needed:

- A description of the components of the context of use including users, equipment, goals
- Usability measures consisting of target and actual value of effectiveness, efficiency, and satisfaction for the intended context

Usability measures

- Provide at least one measure for each of effectiveness, efficiency, and satisfaction
- No objective measures of these values are possible
- Usage of subjective methods instead
Usability framework
Measure values (example)

- **Effectiveness**
  - Measures relate the goals or sub-goals of the user to the accuracy and completeness with which these goals can be achieved
  - Example:
    
    Goal: Reproduce 2-page document in a specified format
    
    \[
    \text{Accuracy} = \text{Number of spelling mistakes}
    \]
    
    \[
    \text{Completeness} = \frac{\text{Number of words in transcribed document}}{\text{Number of words in source document}}
    \]

- Additional measures may be required for particular desired properties of the product which contribute to usability.
## Usability framework

### Measure values

<table>
<thead>
<tr>
<th>Usability objective</th>
<th>Effectiveness measures</th>
<th>Efficiency measures</th>
<th>Satisfaction measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appropriate for trained users</td>
<td>Number of power tasks performed; Percentage of relevant functions used</td>
<td>Relative efficiency compared with an expert user</td>
<td>Rating scale for satisfaction with power features</td>
</tr>
<tr>
<td>Appropriate for walk up and use</td>
<td>Percentage of tasks completed successfully on first attempt</td>
<td>Time taken on first attempt; Relative efficiency on first attempt</td>
<td>Rate of voluntary use</td>
</tr>
<tr>
<td>Appropriate for infrequent or intermittent use</td>
<td></td>
<td>Time spent relearning functions; Number of persistent errors</td>
<td>Frequency of reuse</td>
</tr>
<tr>
<td>Minimisation of support requirements</td>
<td>Number of references to documentation; Number of calls to support; Number of accesses to help</td>
<td>Productive time; Time to learn to criterion</td>
<td></td>
</tr>
<tr>
<td>Learnability</td>
<td>Number of functions learned; Percentage of users able to learn to criterion</td>
<td>Time to learn to criterion; Time to relearn to criterion; Relative efficiency while learning</td>
<td></td>
</tr>
<tr>
<td>Error tolerance</td>
<td>Percentage of errors corrected or reported by the system; Number of user errors tolerated</td>
<td>Time spent on correcting errors</td>
<td>Rating scale for error handling</td>
</tr>
<tr>
<td>Legibility</td>
<td>Percentage of words read correctly at normal viewing distance</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Usability... principles according to Nielsen

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Recognition rather than recall
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

- Called “Nielsen’s heuristics of usability”, 1994
- http://www.useit.com/
Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

For example, pay attention to response time:
- 0.1 sec: no special indicators needed
- 1.0 sec: user tends to lose track of data
- 10 sec: max duration if user to stay focused on action

for longer delays, use percent-done progress bars
Usability…
...principles according to Nielsen  2-9

- **Match between system & real world**
  - The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms.
  - Follow real-world conventions, making information appear in a natural and logical order
- **Bad example (old) Mac and Amiga desktop**
  - Dragging disk to trash
  - Logically should delete it, not eject it
Usability...
...principles according to Nielsen  3-9

- **User control & freedom**
  - Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue.
  - Support undo and redo

- **Wizards**
  - “Surf” between states
  - For infrequent tasks (e.g., modem configuration.)
  - Not for common tasks
  - Good for beginners
    - have 2 versions (WinZip), no fixed paths
Usability...
...principles according to Nielsen  4-9

- **Consistency and Standards**
  - Users should not have to wonder whether different words, situations, or actions mean the same thing.
  - Follow platform conventions
  - Use the same elements always at the same place

- **User Dialogs:**
  - Example: OK always should mean a confirmation statement
  - Usage of standard frameworks (MFC, Java, Qt)
Usability…
…principles according to Nielsen 5-9

- **Recognition rather than recall**
  - Minimize the user's memory load by making objects, actions, and options visible.
  - The user should not have to remember information from one part of the dialogue to another.
  - Instructions for use of the system should be visible or easily retrievable whenever appropriate.

- **Menus, Icon etc. vs. Command line**

- **When using icons in a site's navigation, use icons that are easily recognizable**
Flexibility and efficiency of use

- Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users.

- Allow users to tailor frequent actions (e.g., macros)
Aesthetic and minimalist design

- Dialogues should not contain information which is irrelevant or rarely needed.
- Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
- Aesthetics of display considered

![Example of a form with various design elements including background color, text color, form title, and submit button text.](image-url)
Help users recognize, diagnose, and recover from errors

- Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
Help and documentation

- Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation.
- Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.
Usability... 
...how to get it? (Gould & Lewis, 1985) 1-2

- The design principles for usability (Gould & Lewis, 1985)
- Early focus on the users
  - direct contact between design team and the users
  - get the user involved
- Empirical measurement
  - Usage of prototypes and simulations to carry out real work
  - Performance and reactions should be observed, recorded, and analyzed
    - learnability
    - usability
  - testing of
    - appropriate tasks or concepts
    - time to learn the function keys (novices are different form experts!)
  - collect the users’ thoughts
  - collect the users’ mistakes
  - collect the users’ attitudes

Usability...
how to get it? (Gould & Lewis, 1985) 2-2

- Iterative design
  - incorporate the results from the tests into the next prototype
  - set goals for the system
  - evaluation criteria
    - easy to use
    - user friendly
    - easy to operate
    - simple
    - responsive
    - flexible

- Integrative design (help, training, documentation, etc in parallel).
  - build help, training, documentation, process modules at the same time
Eight Golden Rules of Interface Design (Shneiderman, Designing the User Interface)

- **1 Strive for consistency.**
  - Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent commands should be employed throughout.

- **2 Enable frequent users to use shortcuts**
  - As the frequency of use increases, so do the user’s desires to reduce the number of interactions and to increase the pace of interaction. Abbreviations, function keys, hidden commands, and macro facilities are very helpful to an expert user.

- **3 Offer informative feedback.**
  - For every operator action, there should be some system feedback. For frequent and minor actions, the response can be modest, while for infrequent and major actions, the response should be more substantial.

- **4 Design dialog to yield closure.**
  - Sequences of actions should be organized into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and an indication that the way is clear to prepare for the next group of actions.
Eight Golden Rules of Interface Design
(Shneiderman, Designing the User Interface)

- **5 Offer simple error handling.**
  - As much as possible, design the system so the user cannot make a serious error. If an error is made, the system should be able to detect the error and offer simple, comprehensible mechanisms for handling the error.

- **6 Permit easy reversal of actions.**
  - This feature relieves anxiety, since the user knows that errors can be undone; it thus encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data entry, or a complete group of actions.

- **7 Support internal focus of control.**
  - Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Design the system to make users the initiators of actions rather than the responders.

- **8 Reduce short-term memory load.**
  - The limitation of human information processing in short-term memory requires that displays be kept simple, multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequences of actions.
Interaction with the system: Designing a form

- In most cases only interaction via graphical user interface
- has to be designed very carefully

- Remember the usability heuristics (Nielsen), especially
  - Consistency
  - Visibility of system status
  - Match between system and the real world
  - Consistency and standards
  - Aesthetic and minimalist design
Basic GUI Elements

- **Labels**
  - Think of clear description
- **(Action) Buttons**
  - Actions
  - on which data is the action performed?
- **Input fields**
  - Selection or free text
    - Inputfield, Textarea
  - Selections by
    - Radio Buttons
    - Check Buttons
- **Tables**
  - Can hold many similar, information
- **Trees**
  - Structuring data
- **Menus**
  - Fast access to different functions
- **Help functions**
Basic GUI Elements
The Java Perspective

Basic Controls

Buttons

Combo box

List

Menu

Slider

Spinner

Text field or Formatted text field

http://java.sun.com/docs/books/tutorial/uiswing/components/components.html
Designing a form

- **HCI Perspective**
  - Organize items according to common topics
  - Leave space
  - Short and meaningful descriptions
  - Be clear
  - Follow design guidelines (manufacturers)
  - Keep in mind usability heuristics!!!

- **Software Architecture Perspective**
  - Model-based development
    - Look’n Feel
    - Use Layout Manager
  - MVC Pattern (Paradigm) eases development
    - Multiple Views are possible
Model View Controller

- Multiple views can exist at the same time
- New views can be added without any changes to the model
Different Layouts

LayoutManager

controls position and look’n feel within a visual container ➔ Strategy

- What is my optimal size?
- How can I place my children?

➔ can be ordered hierarchically
Different Layouts: FlowLayout and BoxLayout

- **FlowLayout**
  - Flow means text flow
  - Components as words
  - Properties: preferredSize, minimumSize
  - Children preserve their natural size

- **BoxLayout**
  - Components are aligned on an axis
  - horizontal:
    - Space is partitioned in an “intelligent way”: maximumSize and minimumSize are respected
    - Heights are adjusted

![Diagram of FlowLayout and BoxLayout with components A, B, C, D]
Different Layouts: BorderLayout and GridLayout

- **BorderLayout**
  - Changing Size:
    - NORTH & SOUTH are horizontally resized
    - EAST & WEST are vertically resized
    - CENTER in all directions
  - minimumSize is respected

- **GridLayout**
  - table with equally sized cells
  - each cell contains one component
  - respecting minimumSize
Interaction with the system: Structuring the application 1-2

- Application consists of more than one form
  - How are forms linked?
  - How are tasks performed?

Workflow
Interaction with the system: Structuring the application 2-2

- Think of Nielsen’s heuristics, especially
  - Match between system and the real world
  - User control and freedom
  - Help users recognize, diagnose, and recover from errors
Structuring an application: Some remarks

- **Workflow Structure**
  - Easy to handle, good for non-experienced users
  - Supports one special use case (sequential)
    - Must be known
    - Cannot be changed easily
  - Always show system status (user should be aware of work status)
  - Goal of work process is predefined (c.f. use case output)

- **Structured**
  - Higher degree of freedom during interaction, structure is helpful
  - Goal of work process is predefined (c.f. use case output)

- **Free**
  - Goal is not known
  - (Every) possible change to data can be done at any time
  - Highest degree of freedom, user experience is needed
Conclusions

- Usability – Why it is an important non-functional requirement
- Guidelines for usability
- Designing a form: standards, techniques
- Navigating within an application: Structures