Lecture 1

• Topics Covered
  – Introduction to Graphical User Interfaces
General Software Development Phases

- Requirements Specification
- Analysis and Design
- Implementation
- Testing
- Maintenance
Interface

• What does this word mean?
• What are its different meanings in our field?
Human-Computer Interface (HCI)

- “The study, planning, and design of how people and computers work together so that a person’s needs are satisfied in the most effective way.” [Galitz]
- Things to consider in this area of software design
  - What people want
  - Limitations and abilities of the users, software, and hardware
Human Communication

- What are some examples of how humans “output”? 
- Which of these are most commonly used in software interfaces?
- Which are the most difficult to represent in software?
Exercise

• Draw a screen (form) for a car rental company to record a car reservation
Interface Design

• Why is it important?
• When would poor interface design be most costly?
• What are some examples of interfaces you have used that have
  – good interface design?
  – poor interface design?
Software Requirements

• Functional Requirements
  – The services required of a software system

• Non-Functional Requirements
  – Constraints or limitations placed upon the software system and often upon the software process
    • NFRs are often referred to as quality requirements and even “ilities” due to the spelling of NFR terms

• Let’s consider the NFR of Usability
Usability

- “Usability is a measure of how well a computer system... facilitates learning; helps learners remember what they’ve learned; reduces likelihood of errors; enables them to be efficient, and makes them satisfied with the system.” [Donahue]

- “Usability refers to the efficiency, comfort, safety and satisfaction with which a wide range of people and under a variety of conditions can perform their tasks with a product (i.e., a good or a service). It is much more than a measure of how easily a thing can be used, and it encompasses all aspects of the product and its use, including the hardware and software interfaces, the documentation, the packaging and even the services associated with the product. The meaning is similar to that of the term user-friendliness.” [The Linux Information Project]
Benefits of Usable Software

• Reduced cost for software developer when software is in use
  – Decreased upfront training
  – Decreased documentation
  – Decreased customer service throughout life of the product
  – Decreased maintenance

• Improved production for end user
  – Software is used correctly
  – Software is efficient (more efficient than manual operation)

• Increased profit for software developer
  – Increase in positive recommendations
  – Potential increase in positive media attention
  – Increase of sales
NFR Tradeoffs

• Often, non-functional requirements clash and trade-offs must be negotiated

• Let’s take a look at the chart that shows us how Usability relates to other NFRs
NFR Relationships [Weiger]

<table>
<thead>
<tr>
<th></th>
<th>Availability</th>
<th>Efficiency</th>
<th>Flexibility</th>
<th>Integrity</th>
<th>Interoperability</th>
<th>Maintainability</th>
<th>Portability</th>
<th>Reliability</th>
<th>Reusability</th>
<th>Robustness</th>
<th>Testability</th>
<th>Usability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Efficiency</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flexibility</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interoperability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maintainability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Portability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reliability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reusability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Robustness</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Testability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Progression of interfaces

- Command line
- Cryptic, ambiguous, cluttered fields (70s)
- Grouping and alignment, meaningful field captions (80s)
- Graphics, varying fonts, controls (buttons, check boxes, list boxes, etc.) (90s and beyond)
Interaction Styles

- Command Line
- Menu Selection
- Form fill-in
- Direct Manipulation
- Anthropomorphomic
Graphical Systems

• Advantages

• Disadvantages
What is really important?

- Good interface design
- Preferred interaction style of user
- Meaningful words/icons
- Content and organization of a graphic screen
- Following established principles of usability
General Principles for User Interface Design

- Accessibility
- Aesthetically Pleasing
- Availability
- Clarity
- Configurability
- Consistency
- Control
- Efficiency
- Familiarity
- Flexibility

- Forgiveness
- Obviousness
- Perceptibility
- Positive First Impression
- Predictability
- Recovery
- Responsiveness
- Simplicity
- Transparency
Source

- Wilbert O. Galitz was the source for many concepts presented today