Usability and Interaction Design

Inf-SEPP

Adriana Sejfia

School of Informatics, University of Edinburgh

This Lecture

- Reminders: NFR definitions and classification
- Usability
  - Definitions
  - Why is usability important?
  - Human limitations
  - Principles of user interface design
  - Web accessibility example
- Interaction design
  - Meaning
  - Design considerations: user interaction, information presentation, colour
  - The interaction design process

Classification of Non-Functional Requirements

- Concern the whole system, not just the software.

Classification of Non-Functional Requirements

- Concern the whole system, not just the software.
- Ways the system needs to relate to other systems / versions:
  - Flexibility, maintainability, reusability, portability

Classification of Non-Functional Requirements

- Concern the whole system, not just the software.
- Ways the system needs to relate to other systems / versions:
  - Flexibility, maintainability, reusability, portability
- Properties of the system in use:
  - Usability
  - Dependability (safety, reliability, availability, resilience…)
  - Efficiency (performance, resource usage…)
  - Security (integrity, confidentiality, availability…)
  - Scalability

Usability: Definitions

Usability: Definitions

- ISO 9241-11: "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use."

Usability: Definitions

- ISO 9241-11: "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use."
- Jacob Nielsen: "A quality attribute that assesses how easy user interfaces are to use", defined by 5 components:
  - Learnability
  - Efficiency
  - Memorability
  - Errors
  - Satisfaction

Usability: Definitions

- ISO 9241-11: "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use."
- Jacob Nielsen: "A quality attribute that assesses how easy user interfaces are to use", defined by 5 components:
  - Learnability
  - Efficiency
  - Memorability
  - Errors
  - Satisfaction
- Branches of usability: accessibility (usability by people with different abilities), inclusivity

Why Is Usability Important?

- Software is often found hard to use by its intended end users.
- Many problems are described as 'user errors'.
- Most user errors are actually interface design failures.

Human Limitations

- Limited short-term memory: only 5–7 items at a time

Human Limitations

- Limited short-term memory: only 5–7 items at a time
- Humans make mistakes, especially under stress

Human Limitations

- Limited short-term memory: only 5–7 items at a time
- Humans make mistakes, especially under stress
- Widely varying physical and mental capabilities

Human Limitations

- Limited short-term memory: only 5–7 items at a time
- Humans make mistakes, especially under stress
- Widely varying physical and mental capabilities
- Widely varying personal preferences

Human Limitations

- Limited short-term memory: only 5–7 items at a time
- Humans make mistakes, especially under stress
- Widely varying physical and mental capabilities
- Widely varying personal preferences
- Human brains organise their perceived world differently

Principles of User Interface Design

- User familiarity
- Consistency
- Minimal surprise
- Recoverability
- User guidance
- User diversity
- Resource: Sommerville online chapter 29 on Interaction Design

Principle: User Familiarity

- Interface should use concepts from users' existing experience, not implementation concepts.

Principle: User Familiarity

- Interface should use concepts from users' existing experience, not implementation concepts.
- Example 1: An air traffic control system should present aircraft, height, speed — not implementation details.

Principle: User Familiarity

- Interface should use concepts from users' existing experience, not implementation concepts.
- Example 1: An air traffic control system should present aircraft, height, speed — not implementation details.
- Example 2: The desktop metaphor on PCs is built around familiarity.

Principle: Consistency

- Similar operations should be represented in similar ways (e.g. cancel = ✕, download = ↓).

Principle: Consistency

- Similar operations should be represented in similar ways (e.g. cancel = ✕, download = ↓).
- Consistency within and across applications is important.
- Windows, macOS, Android, iOS all strive for great consistency.
  - Example: iOS Human Interface Guidelines
- Traditional Unix does not (though some distros, e.g. Ubuntu, do better).

Principle: Consistency

- Similar operations should be represented in similar ways (e.g. cancel = ✕, download = ↓).
- Consistency within and across applications is important.
- Windows, macOS, Android, iOS all strive for great consistency.
  - Example: iOS Human Interface Guidelines
- Traditional Unix does not (though some distros, e.g. Ubuntu, do better).
- Be careful: what counts as 'similar' depends on context.

Principle: Minimal Surprise

- Avoid situations where the user will be surprised by the system's behaviour.

Principle: Minimal Surprise

- Avoid situations where the user will be surprised by the system's behaviour.
- Surprise is often caused by modal applications:
  - Same key may have different effects in different modes
  - Users ask: 'Why on earth did it do that?'

Principle: Recoverability

- Allow users to recover easily from errors.

Principle: Recoverability

- Allow users to recover easily from errors.
- Editors without an Undo command were once common — now considered unacceptable.

Principle: Recoverability

- Allow users to recover easily from errors.
- Editors without an Undo command were once common — now considered unacceptable.
- Get confirmation before irreversible actions (e.g. deletion).
  - Caveat: users learn to confirm automatically, reducing the value.

Principle: Recoverability

- Allow users to recover easily from errors.
- Editors without an Undo command were once common — now considered unacceptable.
- Get confirmation before irreversible actions (e.g. deletion).
  - Caveat: users learn to confirm automatically, reducing the value.
- Checkpointing / autosaving is a valuable technique.

Principle: User Guidance

- Interfaces should have built-in assistance and help facilities.

Principle: User Guidance

- Interfaces should have built-in assistance and help facilities.
  - Context-sensitive help that does not overwhelm users

Principle: User Guidance

- Interfaces should have built-in assistance and help facilities.
  - Context-sensitive help that does not overwhelm users
  - Tooltips to explain potentially confusing icons

Principle: User Guidance

- Interfaces should have built-in assistance and help facilities.
  - Context-sensitive help that does not overwhelm users
  - Tooltips to explain potentially confusing icons
  - Meaningful error messages written from the user's point of view, not the system's
    - Bad example: Segmentation fault or java.lang.NullPointerException

Principle: User Diversity

- Recognise and accommodate different user needs:

Principle: User Diversity

- Recognise and accommodate different user needs:
  - Casual users (need guidance) vs. power users (need shortcuts)

Principle: User Diversity

- Recognise and accommodate different user needs:
  - Casual users (need guidance) vs. power users (need shortcuts)
  - Users with varying levels of blindness (colours, larger text, image alt-text)
  - Users with varying levels of deafness (captions, sound controls, text alternatives)

Principle: User Diversity

- Recognise and accommodate different user needs:
  - Casual users (need guidance) vs. power users (need shortcuts)
  - Users with varying levels of blindness (colours, larger text, image alt-text)
  - Users with varying levels of deafness (captions, sound controls, text alternatives)
- Wherever possible, provide choice of interfaces.

Principle: User Diversity

- Recognise and accommodate different user needs:
  - Casual users (need guidance) vs. power users (need shortcuts)
  - Users with varying levels of blindness (colours, larger text, image alt-text)
  - Users with varying levels of deafness (captions, sound controls, text alternatives)
- Wherever possible, provide choice of interfaces.
- Legal obligations: Equality Act 2010 (UK), replacing Disability Discrimination Act 1995
- Applies especially to websites; there have been court cases in the UK and the US.
- W3C Accessibility Guidelines: Priority 1 probably required; Priority 2 EU recommended.

Web Accessibility:
W3C Priority 1 Guidelines (Examples)

- Provide a text equivalent for every non-text element.

Web Accessibility:
W3C Priority 1 Guidelines (Examples)

- Provide a text equivalent for every non-text element.
- For time-based multimedia, synchronise captions / auditory descriptions.

Web Accessibility:
W3C Priority 1 Guidelines (Examples)

- Provide a text equivalent for every non-text element.
- For time-based multimedia, synchronise captions / auditory descriptions.
- Ensure all information conveyed with colour is also available without colour.

Web Accessibility:
W3C Priority 1 Guidelines (Examples)

- Provide a text equivalent for every non-text element.
- For time-based multimedia, synchronise captions / auditory descriptions.
- Ensure all information conveyed with colour is also available without colour.
- Organise documents so they can be read without style sheets.

Web Accessibility:
W3C Priority 1 Guidelines (Examples)

- Provide a text equivalent for every non-text element.
- For time-based multimedia, synchronise captions / auditory descriptions.
- Ensure all information conveyed with colour is also available without colour.
- Organise documents so they can be read without style sheets.
- Ensure pages are usable when scripts / applets / objects are off or unsupported.

Interaction Design: Meaning

- Design of any form of interaction between users and a system:
  - User interface design (focus of these slides)
  - Front-end / interface design in general

Interaction Design: Meaning

- Design of any form of interaction between users and a system:
  - User interface design (focus of these slides)
  - Front-end / interface design in general
- Usually done by software engineers; specialist interaction designers are rare outside large organisations.

Interaction Design: Meaning

- Design of any form of interaction between users and a system:
  - User interface design (focus of these slides)
  - Front-end / interface design in general
- Usually done by software engineers; specialist interaction designers are rare outside large organisations.
- Essential in any software development process due to the importance of usability.

Design Considerations in Interaction Design

- User interaction — how do users interact with the system?
- Information presentation — how is information shown to users?
- Colour — guidelines for effective, accessible colour use

User Interaction Styles [Shneiderman]

- Direct manipulation — e.g. drag-and-drop
- Menu selection — on a directly selected object
- Form fill-in — typical for data entry
- Command language — traditional systems
- Natural language — e.g. Siri; usually a front end to a command language

Exercise: give one advantage and one disadvantage for each style.

Information Presentation

- Presentation should NOT depend on internal representation of data.

Information Presentation

- Presentation should NOT depend on internal representation of data.
- Formats: text, graphs, tables, pie charts, colour-coded maps…

Information Presentation

- Presentation should NOT depend on internal representation of data.
- Formats: text, graphs, tables, pie charts, colour-coded maps…
- Continuously varying information is best shown as an analogue representation, not numbers.

Information Presentation

- Presentation should NOT depend on internal representation of data.
- Formats: text, graphs, tables, pie charts, colour-coded maps…
- Continuously varying information is best shown as an analogue representation, not numbers.
- Data visualisation: presenting large amounts of data in an abstracted visual form, possibly with VR navigation (e.g. network congestion).

Information Presentation

- Presentation should NOT depend on internal representation of data.
- Formats: text, graphs, tables, pie charts, colour-coded maps…
- Continuously varying information is best shown as an analogue representation, not numbers.
- Data visualisation: presenting large amounts of data in an abstracted visual form, possibly with VR navigation (e.g. network congestion).
- Presentation depends on the audience: graphs vs. graphics.

Information Presentation

- Presentation should NOT depend on internal representation of data.
- Formats: text, graphs, tables, pie charts, colour-coded maps…
- Continuously varying information is best shown as an analogue representation, not numbers.
- Data visualisation: presenting large amounts of data in an abstracted visual form, possibly with VR navigation (e.g. network congestion).
- Presentation depends on the audience: graphs vs. graphics.
- See work by Edward Tufte.

Colour Guidelines [Shneiderman + others]

- Use few colours: ≤4–5 per context, ≤7 in total.

Colour Guidelines [Shneiderman + others]

- Use few colours: ≤4–5 per context, ≤7 in total.
- Colour changes should signal something significant.

Colour Guidelines [Shneiderman + others]

- Use few colours: ≤4–5 per context, ≤7 in total.
- Colour changes should signal something significant.
- Colour code to help users, not for its own sake.

Colour Guidelines [Shneiderman + others]

- Use few colours: ≤4–5 per context, ≤7 in total.
- Colour changes should signal something significant.
- Colour code to help users, not for its own sake.
- Colour code consistently (e.g. if red = error, don't use red for a normal stop button).

Colour Guidelines [Shneiderman + others]

- Use few colours: ≤4–5 per context, ≤7 in total.
- Colour changes should signal something significant.
- Colour code to help users, not for its own sake.
- Colour code consistently (e.g. if red = error, don't use red for a normal stop button).
- Be careful about colour pairings and contrast.

Colour Guidelines [Shneiderman + others]

- Use few colours: ≤4–5 per context, ≤7 in total.
- Colour changes should signal something significant.
- Colour code to help users, not for its own sake.
- Colour code consistently (e.g. if red = error, don't use red for a normal stop button).
- Be careful about colour pairings and contrast.
- Vary colours along only one dimension (hue, saturation, or brightness) to make a distinction.

Colour Guidelines [Shneiderman + others]

- Use few colours: ≤4–5 per context, ≤7 in total.
- Colour changes should signal something significant.
- Colour code to help users, not for its own sake.
- Colour code consistently (e.g. if red = error, don't use red for a normal stop button).
- Be careful about colour pairings and contrast.
- Vary colours along only one dimension (hue, saturation, or brightness) to make a distinction.
- Know the output technology (e.g. primary green can be unreadable on screen).

Colour Guidelines [Shneiderman + others]

- Use few colours: ≤4–5 per context, ≤7 in total.
- Colour changes should signal something significant.
- Colour code to help users, not for its own sake.
- Colour code consistently (e.g. if red = error, don't use red for a normal stop button).
- Be careful about colour pairings and contrast.
- Vary colours along only one dimension (hue, saturation, or brightness) to make a distinction.
- Know the output technology (e.g. primary green can be unreadable on screen).
- Remember: ~ red–green combination is difficult for colour-blind.

The Interaction Design Process

- Interfaces should be designed iteratively — only real end users are good judges.

The Interaction Design Process

- Interfaces should be designed iteratively — only real end users are good judges.
- Core activities:
  - User analysis: understand tasks, working environment, other systems. Techniques: task analysis, interviewing, ethnography.

The Interaction Design Process

- Interfaces should be designed iteratively — only real end users are good judges.
- Core activities:
  - User analysis: understand tasks, working environment, other systems. Techniques: task analysis, interviewing, ethnography.
  - System prototyping: evolving from paper mock-ups to increasingly sophisticated prototypes.

The Interaction Design Process

- Interfaces should be designed iteratively — only real end users are good judges.
- Core activities:
  - User analysis: understand tasks, working environment, other systems. Techniques: task analysis, interviewing, ethnography.
  - System prototyping: evolving from paper mock-ups to increasingly sophisticated prototypes.
  - Interface evaluation: collect information about user experience. Techniques: usability testing, inspection, questionnaire/survey methods.

The Interaction Design Process

- Interfaces should be designed iteratively — only real end users are good judges.
- Core activities:
  - User analysis: understand tasks, working environment, other systems. Techniques: task analysis, interviewing, ethnography.
  - System prototyping: evolving from paper mock-ups to increasingly sophisticated prototypes.
  - Interface evaluation: collect information about user experience. Techniques: usability testing, inspection, questionnaire/survey methods.
- For expensive or critical software, involve professionals from appropriate fields (e.g. ethnographers).

Example

An e-commerce site redesigns its checkout flow. The team uses red for promotional banners, red for 'item added to basket' notifications, and red for error messages when a required field is empty.

Example

A developer builds a command-line deployment tool for internal use. When a deployment fails, the tool outputs: \"java.lang.NullPointerException at DeployManager.java:247\". Junior engineers waste 20–30 minutes per incident trying to understand what went wrong.

Example

A local council commissions a new benefits-application website. The design team uses red and green to distinguish 'eligible' and 'ineligible' status indicators throughout the site. No other visual distinction (shape, text label, icon) is used.

Example

A product team is designing a mobile banking app. They debate whether to use direct manipulation (drag-to-pay), a form fill-in (enter amount, payee, confirm), or a natural language interface (type 'pay John £50'). The target users are a broad mix of ages and digital literacy levels.

Example

During usability testing of a new university admissions portal, a designer watches three applicants fail to find the 'Submit Application' button. The designer concludes: 'The users didn't read the instructions — it's a user error.'

Example

A hospital introduces a new patient-records system. Nurses report that the system frequently asks them to confirm routine data entries — even for minor edits — with a dialog box saying \"Are you sure?\". Within two weeks, most nurses are clicking \"Confirm\" automatically without reading the message.

Reading

- Essential: Sommerville SE Chapter 29 on Interaction Design (software-engineering-book.com/downloads/)
- Essential: UK government article on website accessibility (gov.uk/service-manual/…)
- Recommended: NN/g articles — nngroup.com/articles/
- Recommended: W3C Accessibility Guidelines — w3.org/WAI/standards-guidelines/wcag/
- Recommended: Usability Body of Knowledge on evaluation methods — usabilitybok.org/usability-evaluation-methods