Inf2: SEPP
Lecture 24: Usability and Interaction Design

Cristina Adriana Alexandru

School of Informatics
University of Edinburgh

In lecture 4 (Requirements Engineering)

We introduced Non-functional requirements (NFRs)

This lecture...

- Reminders: definition and classification of NFRs
- 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, use of colour
  - The interaction design process

Classification of NFRs

Concern the whole system, not just the software.

- Ways the system needs to be related to other systems and versions of itself:
  flexibility, maintainability, reusability, portability
- Properties of the system in use
  (including things the system must not do or allow to be done)
  usability
  dependability (safety, reliability, availability, resilience, . . .)
  efficiency (performance, resource usage, . . .)
  security (integrity, confidentiality, availability, . . .)
  scalability

Usability: Definitions

Many definitions available, but most frequently quoted are:

- The ISO 9241-11 standard definition: "*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.*"
- The definition provided by Jacob Nielsen: "*A quality attribute that assesses how easy user interfaces are to use*", and defined by 5 quality components: learnability, efficiency, memorability, errors, satisfaction

Branches of usability: accessibility (usability by people with all abilities), inclusivity.

Why is usability important?

Much software is found to be hard to use by the intended end users.

Many problems are described as “user errors”

Most user errors are actually interface design failures.

Why does this happen?

Human limitations

Humans have *very* limited short-term memory: 5–7 items.

Humans make mistakes, especially under stress.

Humans have widely varying capabilities, both physical and mental.

Humans have widely varying personal preferences.

Human brains organise their perceived world differently.

Principles of User Interface design

There are some fairly basic principles which, when followed, will reduce the chances of a design with poor usability.

- User familiarity
- Consistency
- Minimal surprise
- Recoverability
- User guidance
- User diversity

Resource: Sommerville online chapter 29 on *Interaction Design*.

User familiarity

The interface should 'look familiar' to the users – it should use concepts and entities from the existing experience of the users, and not implementation concepts and entities. For example:

1. An air traffic control system should present users with (representations of) aircraft, height, speed etc.; not anything to do with the implementation.
2. Familiarity is the guiding philosophy behind the desktop metaphor on PCs.

Consistency

Similar operations should be represented in similar ways (e.g., cancel as an 'x', drop down icon for download).

Consistency both within and across applications is important.

Windows, MacOS, Android, iOS strive for great consistency.

- E.g. see iOS Human Interface Guidelines

Traditional Unix doesn't (but some distributions more than others do e.g., Ubuntu).

What does 'similar' mean? Be careful about the different contexts!

Minimal surprise

Avoid situations where the user will be surprised by the behaviour of the system ('why on earth did it do that?').

Surprise is often caused by modal applications: same key may have different effects in different modes.

Recoverability

Allow the user to recover (easily) from errors.

Once upon a time, there were editors without an ‘undo’ command!

Get confirmation of irreversible actions (e.g. deletion). (But users learn to confirm automatically...)

Checkpointing/autosaving is a valuable technique.

User guidance

Interfaces should have built-in user assistance and help facilities, such as:

- Context-sensitive help which does not overwhelm users
- Tooltips to explain potentially confusing icons
- Meaningful error messages. *Segmentation fault* or *java.lang.NullPointerException* is not useful. Make sure error messages are written from the *user's* point of view, not the system's.

User diversity

Recognise and make the interface adaptable to different user needs. For example:

- Casual users (need guidance) vs power users (need shortcuts)
- Users with different levels of blindness may need different colours, larger text, buttons, replacing images with text, etc.
- Users with different levels of deafness may need replacing sound with text, captioning, sound controls, etc.

Wherever possible, provide choice of, and in, interfaces.

Legal obligations enshrined in Equality Act 2010 (superseding Disability Discrimination Act 1995), applying in particular to websites.

W3C has Accessibility Guidelines - Priority 1 probably required, Priority 2 EU recommended. There have been court cases in the UK and the US.

Web Accessibility Example

Some of W3C Priority 1 Web Content Accessibility Guidelines

- Provide a text equivalent for every non-text element
- For any time-based multimedia presentation (e.g., a movie or animation), synchronize equivalent alternatives (e.g., captions or auditory descriptions of the visual track) with the presentation
- Ensure that all information conveyed with color is also available without color
- Organize documents so they may be read without style sheets
- Ensure that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported

Interaction design

- Is about the design of any form of interaction between users and the system:
  - User interface design (focus of these slides)
  - Interface (i.e. front end) design in general

- Work of software engineers in many organisations, rarely (large organisations) of specialist interface designers.

- Essential as part of any software development process, due to importance of usability

Design considerations in interaction design

- User interaction
- Information presentation
  - Colour

User interaction

How do users interact with the system? Choice depends on task *and on user preferences*. For example [Shneiderman]:

- Direct manipulation, such as drag-and-drop.
- Menu selection, perhaps on a directly selected object.
- Form fill-in, as typically used for data entry.
- Command language, as used by traditional systems.
- Natural language, usually as a front end to a command language but now e.g., Siri.

Exercise: give one advantage and one disadvantage for each of these styles.

Information presentation

How should information be *presented* to the user?

N.B. This need not and should not depend on how information is *represented* internally!

Perhaps as text, graphs, tables, pie charts, colour coded maps, . . .

E.g. continuously varying information is best presented in an analogue representation, not as numbers.

Data visualization means presenting (usually large) amounts of data in an abstracted visual representation, possibly with virtual reality navigation. (E.g. network congestion.)

Presentation should depend on the audience, e.g., graphs vs. graphics

See work by Edward Tufte.

Colour

deserves a lecture in itself. A few of guidelines [first 5 from Shneiderman]:

- Use few colours, no more than four or five per context, no more than seven in total.
- Colour changes should signal something significant.
- Colour code to help users, not for the sake of it.
- Colour code consistently. (E.g. if red means error in most of the application, don't use red for a normal stop button.)
- Be careful about colour pairings. Don't do this.
- In general, vary colours along only one of the three dimensions (*hue*, *saturation*, *brightness*) to make a distinction.
- Know the output technology. Primary green is often unreadable on screen, but would be readable in print.
- Remember that around 10% of men are red–green colour-blind!

The interaction design process

Interfaces should be designed iteratively, regardless of the model for code design. *Only real end users* are good judges of the interface.

Core activities:

- User analysis: develop understanding of tasks that users do, their working environment, other systems that they use, interacting with other people etc. Techniques: task analysis, interviewing, ethnography.
- System prototyping: ideally evolving from paper mock-ups to increasingly sophisticated prototypes.
- Interface evaluation: collect information about user experience with the interface. Techniques: usability testing, inspection, questionnaire and survey methods (see Usability BOK)

In expensive or critical software, involve professionals from appropriate fields (e.g. ethnography).

Reading

Essential: Sommerville SE Chapter 29 on Interaction Design (https://software-engineering-book.com/downloads/)

Essential: Article on UK law about website accessibility from the Scottish Government: https://www.gov.uk/service-manual/helping-people-to-use-your-service/making-your-service-accessible-an-introduction

Recommended: Most popular articles at https://www.nngroup.com/articles/

Recommended: The W3C Accessibility guidelines: https://www.w3.org/WAI/standards-guidelines/wcag/

Recommended: Usability Body of Knowledge on usability evaluation methods: https://www.usabilitybok.org/usability-evaluation-methods