COMP210: Interfaces & Interaction
2: Expert Evaluation

Learning Outcomes

After this session you will be able to:

- **Describe** the difference between expert, automated & user testing.
- **Justify** the choice of heuristics for various situations
- **Implement** a heuristic evaluation.

Types of Usability Testing

According to Lazar et al., there are three different types of usability testing:

- **Expert** (today)
- Automated
- User

User Testing

`groups of representative users attempting a set of
representative tasks.' The results should feed back into the
design of the interface.

- ► Should happen throughout the entire design and
  development cycle and it is always easier to start
  earlier than later.
- ► Allows you to test assumptions about the user early.
- ► May produce negative results, do not take them
  personally.

Automated

'An automated usability test is a software application that inspects a series of interfaces to assess the level of usability. Often, this works by using a set of interface guidelines and having the software compare the guidelines to the interface.'

- ► They can be executed very quickly and often generate reports.
- ► They can detect a limited amount of issues.
- ► Focus on quantitative data such as the number of different fonts and sizes used, or depths of menu layers.

PAPER: Melody Y. Ivory and Marti A Hearst. 2001. The state of the art in automating usability evaluation of user interfaces. ACM Comput. Surv. 33, 4 (December 2001), 470-516.
DOI=http://dx.doi.org.ezproxy.falmouth.ac.uk/10.1145/503112.503114

Expert (Today)

`Expert-based tests are essentially structured inspections by interface experts.'

- ▶ To avoid bias, the expert should not have any involvement in the design and development of the interface.
- ▶ Can often be carried out without a deep understanding of in the interface.
- ▶ Expert testing can be used in conjunction with user testing but must be done first.

Types of Expert-Based Usability Testing

- ► Consistency Inspection
- ► Cognitive Walk-Through
- ► **Heuristic Evaluation**
- ► Most tests that draw upon expert experience to derive insight.

Definition: Heuristic

adjective

1. serving to indicate or point out; stimulating interest as a means of furthering investigation.

2. encouraging a person to learn, discover, understand, or solve problems on his or her own, as by experimenting, evaluating possible answers or solutions, or by trial and error: a heuristic teaching method.

3. of, relating to, or based on experimentation, evaluation, or trial-and-error methods.

4. Computers, Mathematics. pertaining to a trial-and-error method of problem solving used when an algorithmic approach is impractical.

http://www.dictionary.com/browse/heuristic

Nielsen Norman Group

Nielsen Norman Group

'Heuristic evaluation involves having a small set of
evaluators examine the interface and judge its
compliance with recognised usability principles (the
"heuristics").'

Link

4 Steps

1. Training - Provide expert with the domain specific knowledge needed for a given test scenario
2. Experts carry out evaluation **(two passes minimum)** in isolation of other experts but usually with one or more observers (`experimenter´).
3. Expert assigns a severity rating to each issue found.
4. Short debrief to discuss the findings with the relevant stakeholders (usually the other experts and the design team).

What is tested?

- ► The whole product if time allows of the system is not big.
- ► Parts of the whole specific to individual tasks.

Experts should run through the experiment at least twice. The first pass allows the expert to familiarise themselves with the task. The second pass, should focus on identifying issues. Any further passes are at the observers discretion but are probably not needed.

When?

'Since the evaluators are not using the system as such (to perform a real task), it is possible to perform heuristic evaluation of user interfaces that exist on paper only and have not yet been implemented (Nielsen 1990). This makes heuristic evaluation suited for use early in the usability engineering lifecycle.' (Nielsen, 1995)
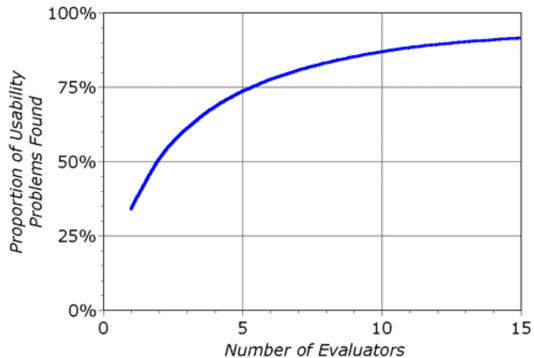
How many experts should evaluate an interface?

Figure 2
Curve showing the proportion of usability problems in an interface found by heuristic evaluation using various numbers of evaluators. The curve represents the average of six case studies of heuristic evaluation.
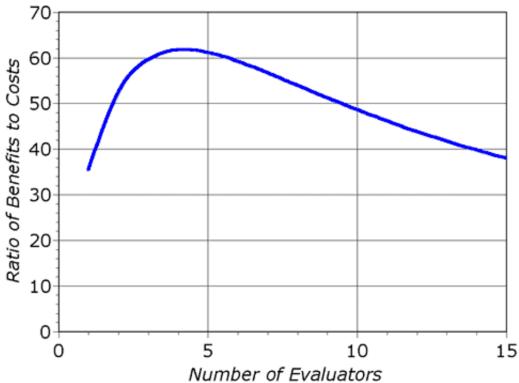
Figure 3
Curve showing how many times the benefits are greater than the costs for heuristic evaluation of a sample project using the assumptions discussed in the text. The optimal number of evaluators in this example is four, with benefits that are 62 times greater than the costs.

What Issues are we Looking for?

we are looking for issues in located in a dialogue in four different ways:

1. at a single location in the interface
2. at two or more locations that have to be compared to find the problem
3. as a problem with the overall structure of the interface,
4. and finally as something that ought to be included in the interface but is currently missing.

link
42% likely to identify major issues vs. 32% chance of spotting minor issues (Nielsen 1992).

The Heuristics - Nielson 1994

- **Visibility of system status:** The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

- **Match between system and the 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.

- **User control and 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.

- **Consistency and standards:** Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

- **Error prevention:** Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

- **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.

- **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.

- **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.

- **Help users recognise, 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.

Severity ratings

`can be used to allocate the most resources to fix the most serious problems and can also provide a rough estimate of the need for additional usability efforts. ´ Link

1. I don't agree that this is a usability problem at all
2. Cosmetic problem only: need not be fixed unless extra time is available on project
3. Minor usability problem: fixing this should be given low priority
4. Major usability problem: important to fix, so should be given high priority
5. Usability catastrophe: imperative to fix this before product can be released