HCI: HEURISTICS

Dr Kami Vaniea
Neilson’s 10 Heuristics

“Heuristics” are simple rules that can be easily applied and are true in most situations. Using the ten heuristics to the right we can detect a large percentage of usability issues.

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility an deficiency of use
8. Aesthetics and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation
Heuristic Evaluation

• Basic idea: Have one or more experts evaluate an interface based on a common set of criteria
• Heuristic Evaluation is very easy to do

• Pros
  • Can be done by even a single person
  • No ethics, recording, or other human-related problems
  • Minimal expense to find a large number of potentially expensive problems

• Cons
  • Experts are not the same as end users, they will miss some things
  • Heuristics are the most common types of problems but they do not represent all problems
Stages of a heuristic evaluation

- Briefing session to tell experts what to do
- Evaluation period of 1-2 hours where:
  - Each expert works separately
  - Take one pass to get a feel for the product
  - Take a second pass to focus on specific features
- Debrief session in which experts work together to prioritize problems.
No. of evaluators & problems

Figure 15.1 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.

Number of evaluators

- Nielsen suggests that on average 5 evaluators identify 75-80% of usability problems.
- Cockton and Woolrych (2001) point out that the number of users needed to find 75-80% of usability problems depends on the context and nature of the problems.
Neilson’s 10 Heuristics

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility an deficiency of use
8. Aesthetics and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation
Visibility of system status

Me adding the Q&A session to my Google calendar
Visibility of system status

Better add a reminder or I might forget to go
Is the reminder saved?
Visibility of system status

I click the back button without clicking “save” and get a warning.
Usability Aspect Reports (UAR)

- Similar to a bug report, but for usability issues
- Can be about good or bad features
- Should link to a heuristic
<table>
<thead>
<tr>
<th>HE-01</th>
<th>Problem/Good Aspect: Problem</th>
</tr>
</thead>
</table>

**Name:** Saved status not visible for calendar changes

**Evidence**

**Heuristic:** Visibility of status  
**Interface aspect:**

**Explanation:** When a calendar event element is changed it is not clear if it is automatically saved or not. As a result a user may try and leave the page when it is not saved.

**Severity or Benefit**

**Rating:** Low  
**Justification:** A warning box pops up preventing accidental loss of data  
**Frequency:** Medium  
**Impact:** Low  
**Persistence:** High (happens every time)  
**How I weight the factors:** The error is very recoverable and the warning is clear, so this may be an issue but it is a low importance one.

**Possible solution and/or tradeoff:** Automatic saving is possible, but that may lead to other issues

**Relationships:** None
Neilson’s 10 Heuristics

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility an deficiency of use
8. Aesthetics and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation
Match between system and the real world

- The interface should use concepts, language and real-world conventions that are familiar to the user.

Why

- The user already has knowledge from the outside world. A user interface can leverage that knowledge
- If the interface does not match the way the world typically works people will become confused
- Metaphors and affordances are useful here
User control and freedom

- Allow the user to have control of the interaction. Users should be able to undo actions, exit from any sequence of actions, and not be forced into a series of actions.

- Why
  - Users make errors sometimes
  - They need the ability to go back and correct the errors
Consistency and standards

- Information that is the same should appear to be the same
- Information that is different should be expressed differently
- Developers need to know the conventions being used in the software

Why
- Similar to the real world heuristic, people can leverage things they already know
- They will expect that something they learned will continue to be true
Error prevention

• If possible, prevent errors from happening in the first place
• Similar to visibility of system status, but specifically involves preventing an error from happening
• Example: if the user needs to select 3 things, don’t wait till the next screen to tell them that they have selected 4
• Why
  • Users are not machines, they do not always perceive all the information available and they can temporarily forget things
  • Computers are really good at using all the information available and remembering the last few things
Recognition rather than recall

- Show all the options available to the user rather than expecting them to remember them all
- Do not require users to remember information from one screen to the next
- Why
  - People are less good at remembering (recall) than they are at recognizing (recognition)
Flexibility and efficiency of use

• Experts should have a way to use the interface faster or more efficiently

• Design should have accelerators like keyboard shortcuts to allow skilled users to move faster

• Why
  • Using the mouse is MUCH slower than the keyboard. Users who know what they want should be able to find it quickly and efficiently
Aesthetics and minimalist design

- Get rid of clutter
- See most of the “Don’t make me think” book
- Why
  - The more things there are to look at, the harder it is for a user to process the data
Help users recognize, diagnose, and recover from errors

- Error messages should be clear, written in plain English, explain the problem, give constructive advice on how to solve the problem

- Why
  - Errors should only be shown to users when the system can no longer make a choice on their behalf. The error needs to be clear about what it is the user needs to do or provide input on
Help and documentation

- Unless the system is extremely simple, some people will need help documentation

- Why
  - People learn about things in different ways. Some people learn by playing around and pushing buttons, other people learn by reading. The system needs to support all people.
Heuristics for websites focus on key criteria (Budd, 2007)

- Clarity
- Minimize unnecessary complexity & cognitive load
- Provide users with context
- Promote positive & pleasurable user experience
Advantages and problems

• Few ethical & practical issues to consider because users not involved.

• Can be difficult & expensive to find experts.

• Best experts have knowledge of application domain & users.

• Biggest problems:
  – Important problems may get missed;
  – Many trivial problems are often identified;
  – Experts have biases.
Questions?