COS 436
Heuristic Evaluation Handout

Evaluating Severity
+
List & Discussion of Nielsen's Heuristics

Evaluating Severity

Adapted from http://www.nngroup.com/articles/how-to-rate-the-severity-of-usability-problems/

To assess severity of a usability problem in heuristic evaluation, consider the following factors:

- The **frequency** with which the problem occurs: Is it common or rare?
- The **impact** of the problem if it occurs: Will it be easy or difficult for the users to overcome?
- The **persistence** of the problem: Is it a one-time problem that users can overcome once they know about it or will users repeatedly be bothered by the problem?

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

Ten Usability Heuristics

by Jakob Nielsen

Additional information adapted from Scott Klemmer

H1. Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Show feedback about:

- Time (how much time remaining?)
- Space (how much space remaining, e.g. in your gmail account?)
- Change (e.g., the document has changed since your last save)
- Action (that the user is expected to perform some action next)
- System's actions (e.g., a notification message that an online transaction has been completed successfully, and the user should look for an email receipt.)
- Completion (e.g., a long system process has completed successfully)

H2. 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.

Examples of match:

- Print dialogs show a small thumbnail of the printed page, so you can see which pages you are printing, and whether in portrait or landscape
- Say “e-mail” rather than “electronic correspondence”
- Use familiar categories (e.g., states and zip codes for locations)

H3. 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.

Also support freedom to explore possible system functions/actions.

H4. Consistency and standards
Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Some more guidelines:

- Follow “look and feel” guidelines for a platform.
- Follow conventions regarding options (e.g., OK/Cancel buttons should function in the anticipated, common manner)
- This doesn’t mean that “OK/Cancel” is always the best choice. You may opt for clearer options instead (e.g., “keep .do” & “Use .pdf” example from class slides)

H5. 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.

Some more guidelines:

- Prevent data loss (e.g., prevent overwriting file in error)
- Prevent clutter (e.g., notify if user is importing a duplicate song into iTunes)
- Prevent misinterpretation (avoid “click OK to go back and save [your unsaved document], or select Cancel to continue”)
- Prevent bad input (e.g., use a graphical calendar widget for selecting dates)
- At the same time, prevent unnecessary constraints

H6. 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.

Some more guidelines:

- Use previews of files/data to enable efficient recognition and decision-making

H7. 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.

Some more guidelines:

- Save the user time by including defaults likely to cover most common cases (e.g., for a travel website, include radio buttons for most common travel destinations)
- Include appropriate ambient information that may help users make faster decisions without added overhead (e.g., show weather forecast information in background of a calendar app)
- Ensure that all information at the foreground of an interface is **relevant**

H8. 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.

Some more guidelines:

- For webpages, include most important information “above the fold” (i.e., user should not have to scroll to see the information they came to the site to see)
- Maximize the “signal-to-noise” ratio (e.g., don’t use lots of colors unnecessarily; noticeable color deviations should convey information; avoid visual clutter with thick borders that draw eye away from useful information, etc.)

H9. Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

H10. 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

Other suggestions:

- Show user examples to help him or her learn