

Tutorial Sheet 1: Introduction

A computer with internet access is required for this tutorial sheet.

Assignment 1: Assessing Maintainability

Download the source code of Jabref, an open source bibliography reference manager written in Java, from the following URL:

http://downloads.sourceforge.net/project/jabref/jabref/2.6/JabRef-2.6-src.zip?use_mirror=kent.

Open the class `net.sf.jabref.EntrySorter` in an editor of your choice and perform a review of the source code. Make a list of issues that in your opinion negatively affect the *maintainability* of the software.

Assignment 2: Assessing Usability

Jakob Nielsen developed the following ten principles for user interface design:

- *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 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.
- *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.

(Source: http://www.useit.com/papers/heuristic/heuristic_list.html)

1. Pick two of these heuristics and try to concretize them by giving positive and negative examples of aspects in user interfaces.
2. Name an example a software you use frequently that violates at least one of the principles and explain the violation.
3. Discuss advantages and disadvantages of design heuristics.