Vaihe I

Tutki käyttöliittymää käyttäen apuna tehtäväluetteloa. Voit käydä tehtäviä läpi missä järjestyksessä haluat. Kiinnitä erityistä huomiota tärkeisiin tehtäviin.

Jokaisen käyttäjän polulle osuvan toiminnon kohdalla käytä seuraavia tarkistuskysymyksiä mahdollisten ongelmakohtien paikantamiseksi:

A. **Will users know what they need to do next?** It is possible that they simply cannot figure out what to do next.

B. **Will users notice that there is a control (e.g. button, menu) available that will allow them to accomplish the next part of their task?** It is possible that the action is hidden or that the terminology does not match what users are looking for. In either case, the correct control exists but users cannot find it. The existence and quality of labels on controls and the number of controls on the screen influence the user's ability to find an appropriate control.

C. **Once users find the control, will they know how to use it** (e.g. click on it, double click, pull down menu)? For example, if the control is a pull-down menu but it looks like a normal button, users may not understand how to use it. Users may find the icon that corresponds to the desired action, but if it requires a triple-click they may never figure out how to use it.

D. **If users perform the correct action, will they see that progress is being made toward completing the task?** Does the system provide appropriate feedback? If not, users may not be sure that the action they just performed was correct.

Kirjaa ylös löytämäsi ongelmakohdat. Merkitse ylös ainakin

- tehtävä johon ongelmakohta liittyi
- missä vaiheessa/paikassa ongelma ilmeni
- mihin kysymykseen ongelma liittyi

Kun olet mielestäsi saanut riittävän kuvan järjestelmän laajuudesta ja käyttäjien näissä tehtävissä tarvitsemista toiminnoista, siirry vaiheeseen II.

Vaihe II

Tutki järjestelmää käytettävyysheuristiikkojen valossa. Kirjaa ylös löytyneet ongelmat samaan tapaan kuin I-vaiheessa.

Ten Usability Heuristics

by Jakob Nielsen

These are ten general principles for user interface design. They are called "heuristics" because they are more in the nature of rules of thumb than specific usability guidelines.

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

2. 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.

3. 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.

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

5. 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.

6. 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.

7. 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.

8. 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.

9. 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.

10. 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.