

Käyttöliittymän heuristinen arviointi ääneenajattelua hyödyntäen

Tarkoituksena on toteuttaa vertaisarviointi toisen ryhmän tekemästä käyttöliittymän toimintopolusta hyödyntämällä ääneen ajattelua ja havainnointia heuristiikkojen avulla. Yksi ryhmäläinen on käyttäjä, joka seuraa toimintopolkua ja ajattelee ääneen kuvaten toimintopolun käytettävyyttä ja siinä mahdollisesti esiintyviä käytettävyyso ongelmia. Havainnoija kirjaa arviointilomakkeelle ääneen ajattelussa todetut käytettävyyso ngelmat.

Arviointi suoritetaan 2-3 hengen ryhmissä.

1. Alkutoimet

- Rooleista sopiminen
 - *Käyttäjä*: seuraa toimintopolkua ja suorittaa sen mukaisia tehtäviä etsien käytettävyyso ngelmia ja ajattelee samalla ääneen.
 - *Havainnoitsija*: kirjaa lomakkeelle ylös käytettävyyso ngelmia ja niihin liittyviä heuristiikkoja. Kolmen hengen ryhmissä voi olla 2 havainnoitsijaa.
 - **ÄÄNEENAJATTELUN AIKANA EI KESKINÄISTÄ KESKUSTELUA**

2. Käyttöliittymän vertaisarviointi toisen ryhmän suunnitelmaa käyttäen

- Käyttäjä arvioi käyttöliittymän toimintopolun onnistuneisuutta.
- Käyttäjä ajattelee ääneen ja keskittyy erityisesti tunnistamaan mahdollisia ongelmia, jotka rikkovat annettuja heuristiikkoja.
- Havainnoitsija havainnoi suoritusta ja kirjaa havainnointituloksia alustavasti joko paperilomakkeelle tai suoraan raportointilomakkeelle (liitteenä).
- Arviointitulosta täydennetään arvioinnin jälkeisellä keskustelulla, ks. seuraava kohta.

3. Keskustelu ja raportointi

- Ääneenajattelu -tilanteessa havaitut ongelmat käydään ryhmässä läpi ja niitä peilataan annettuihin heuristiikkoihin. Tarvittaessa toimintopolun seuraamiseen voidaan palata.
- Raportointiin käytetään arviointilomaketta.
- Vakavuusluokitus voidaan tehdä lopuksi kun kaikki ongelmat on kirjattu. Korjaus/parannusehdotuksia voidaan kirjata haluttaessa.

LIITE: Heuristiikat

1. **Visibility of system status.** Järjestelmän pitäisi aina antaa käyttäjille tietoa siitä, mitä on tekeillä ja antaa tarkoituksenmukaisen palautteen kohtuullisessa ajassa.
2. **Match between system and the real world.** Järjestelmän tulisi käyttää käyttäjien kieltä, sanoja, lauseita ja käyttää käyttäjille tuttuja käsitteitä mieluummin kuin siihen suuntautuneita termejä. Järjestelmän tulisi seurata realimailman käytänteitä, jolloin tiedot näkyvät luonnollisessa ja loogisessa järjestyksessä.
3. **User control and freedom.** Käyttäjät valitsevat usein järjestelmässä toimintoja vahingossa ja tarvitsevat selvästi merkityn "hätäuloskäynnin" lähtemiseen epätoivotusta tilasta ilman pitkää vuoropuhelua järjestelmän kanssa eli tuen kumo- ja tee uudelleen-toimintoihin.
4. **Consistency and standards.** Käyttäjien ei pitäisi miettiä, onko erilaisia sanoja, tilanteita tai toimia, jotka tarkoittavat samaa asiaa järjestelmässä. Järjestelmän tulisi seurata alustalla olevia yleisiä käytänteitä.
5. **Error prevention.** Hyvää virheilmoitusta parempi on, että huolellinen suunnittelu, joka estää ongelmien ilmenemisen. Järjestelmällä on mahdollista poistaa virhealttiutta aiheuttavia tiloja tai tarkistaa niitä ja näyttää käyttäjälle vahvistavan vaihtoehdon ennen kuin käyttäjä sitoutuu toimintaan.
6. **Recognition rather than recall.** Minimoida käyttäjän muistin kuormitusta luomalla kohteita, toimia tai vaihtoehtoja, joita on näkyvissä. Käyttäjän ei pitäisi muistaa informaatiota yhdestä osasta vuoropuhelusta järjestelmän kanssa toisessa osassa. Järjestelmän käyttöohjeet pitäisi olla näkyvissä tai helposti löydettävissä aina tarvittaessa.
7. **Flexibility and efficiency of use.** Järjestelmän tulisi voida nopeuttaa vuorovaikutustaan kokeneemman käyttäjän kanssa verrattuna kokemattoman käyttäjän kanssa. Järjestelmän tulisi sallia käyttäjien räätälöidä useita toimintoja itselleen sopivaksi.
8. **Aesthetic and minimalist design.** Järjestelmän ja käyttäjän kanssa ei pitäisi olla vuoropuheluita, jotka sisältävät turhia tietoja, jotka ovat merkityksettömiä tai harvoin tarpeen. Jokainen ylimääräinen tieto kilpailee järjestelmässä olevan tärkeän tiedon kanssa ja vähentää merkityksellisiä tietoa antavien kohteiden suhteellista näkyvyyttä.
9. **Help users recognize, diagnose, and recover from errors.** Virheilmoitukset tulee ilmaista yksinkertaisella kielellä (ei koodeja). Niiden tulisi osoittaa tarkalleen ongelma, ja rakentavasti ehdottaa ratkaisua siihen.
10. **Help and documentation.** Käyttäjälle olisi parempi, jos järjestelmää voidaan käyttää ilman asiakirjoja, kuitenkin voi olla tarpeellista antaa käyttäjälle apua ja dokumentaatiota. Tällaisten tietojen tulisi olla helposti haettavissa, keskittyä käyttäjien tehtäviin toimintoihin, luetella askel askeleelta, kuinka käytännöllisiä toimia toteutetaan. Näiden asiakirjojen ei tulisi olla liian suuria.

Käytettävyyssongelmien vakavuusluokitus

1. Kosmeettinen – ei välttämättä ongelma, korjaamista tulisi harkita
2. Vähäinen – ongelman korjaamista suositellaan
3. Häiritsevä – haittaa käyttöä, ongelma on korjattava
4. Vakava – vaikeuttaa käyttöä merkittävästi, ongelma on korjattava mahdollisimman pian
5. Katastrofaalinen – lähes käyttökelvoton tuote, tuotetta ei voida julkaista, korjattava välittömästi

Käytettävyyssongelman vakavuuteen vaikuttavia tekijöitä esim.:

- Esiintymistiheys – esiintyykö ongelma usein?
- Vaikutukset – onko ongelmatilanteesta helppo vai vaikea selvittää?
- Toistuvuus – häiritseekö sama ongelma aina, kun käyttäjä törmää siihen?

(Lähde: Nielsen, J. (1995). 10 Usability Heuristics for User Interface Design)