

Jukka S. Rannila

Lyhyt katsaus käytettävyyden testaamiseen

Lähtökohtia

Mikä on käytettävyyden asiantuntijoiden ideologia? Aloin pohtia tätä asiaa luettuani jonkin verran asiaa käytettävyydestä. Aikaisemmin olen todennut, että yksittäisellä ihmisellä on käytössään vain tietty määrä näkökulmia, joita hän osaa käyttää pohtiessaan eri asioita. Käytettävyys näyttää olevan yksi tietty näkökulma asioihin, joita on siis kaikenlaisia eri aihepiireihin liittyen.

Itse olen opiskellut käytettävyyttä suhteellisen vähän: eli vain parin kurssin verran. Kurssien nimet ovat seuraavat: graafisen käyttöliittymän suunnittelu ja arviointi (GRAKSA) ja käytettävyyden perusteet. Graafisen käyttöliittymän suunnittelu ja arviointi (GRAKSA) kurssina tapahtui pääasiassa Seinäjoella. Käytettävyyden perusteet kurssina tapahtui Tampereella.

Moni käytettävyyttä enemmän harrastanut henkilö on käynyt läpi useamman kurssin ja/tai on lukenut paljon enemmän kirjallisuutta käytettävyydestä.

Tämä kirjoitus on kuitenkin hyvin rajoitettu katsaus vaikeaan aiheeseen.

Lyhyt katsaus käytettävyyden testaamiseen

Peruskirja 1: Cooper (1999) / Lähestymistapa eri aiheisiin ja käytettävyyden testaamiseen

Cooper (1999) on lukemani peruskirja vuorovaikutussuunnittelusta. Cooper (1999) perusteella pitää todeta, että vuorovaikutussuunnittelu ja käytettävyyssuunnittelu tarkoittavat eri asioita. Tämän vuoksi käyn läpi Cooper (1999) perusteella eri asioita vuorovaikutussuunnittelun kannalta.

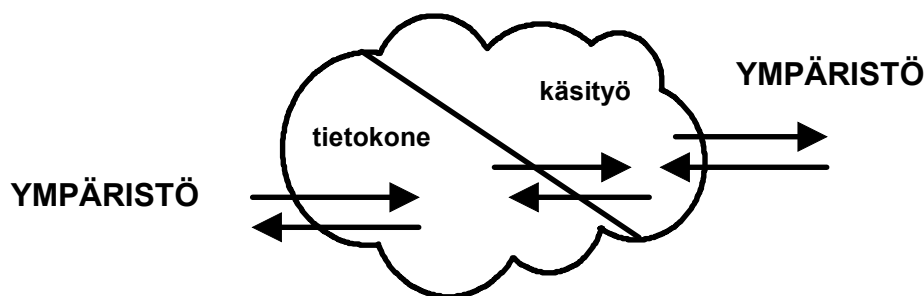
Engelman ratkaisun avain on vuorovaikutussuunnittelu. Me tarvitsemme uuden sukupolven vuorovaikutussuunnittelun ammattilaisia, jotka suunnittelevat sen miten ohjelmistot käyttäytyvät. Nykypäivän ohjelmoijat suunnittelevat tietoisesti ”koodit” ohjelmien sisällä, mutta suunnittelevat vain tahattomasti vuorovaikutuksen ihmisten kanssa. He suunnittelevat, kuinka se toimii, mutta eivät kuinka se käyttäytyy, kommunikoi ja tiedottaa. Sitä vastoin interaktiivisuuden suunnittelijat keskittyvät vain siihen, miten käyttäjät näkevät ja toimivat ohjelmistopohjaisten tuotteiden kanssa. Tämä vuorovaikutussuunnittelijoiden ala on uusi ja tuntematon ohjelmoijille, joten he päästävät – jos päästävät – interaktiivisuuden suunnittelijat paikalle vasta kun ohjelmointi on saatu päätökseen. Siinä vaiheessa on jo liian myöhäistä. (Cooper 1999, sivut 46-47)

Vuorovaikutussuunnittelu on siis täysin oma toiminnan lohko, ja Cooper (1999) esittelee kirjassaan vuorovaikutussuunnittelua eri tavoilla.

Pidän ”vuorovaikutussuunnittelusta” enemmän kuin ”käyttöliittymäsuunnittelusta”, koska ”käyttöliittymäsuunnittelu” antaa ymmärtää, että koodi on tässä ja ihmiset ovat tuossa, ja käyttöliittymä, joka välittää niiden välistä viestintää siinä välissä. Se antaa ymmärtää, että vain käyttöliittymä vastaa käyttäjän tarpeista. Suunnittelun eristäminen käyttöliittymätasolle antaa ohjelmoijille vapauden ajatella näin: ”Voin koodata mitä tahdon, koska käyttöliittymä tulee tämän päälle kun olen valmis”. Se lykkää suunnittelun ohjelmoinnin jälkeen, jolloin on liian myöhäistä. (Cooper 1999, sivu 54)

Seuraavassa kuvassa pohdin tietokoneen käyttöä ja käsityötä jossain ympäristössä. On totta, että tietokoneen ja ihmisen välille syntyy jokin rajapinta (interface), jolloin ihminen käyttää jotain tietokonetta rajapinnan avulla.

Lederman & Beynon-Davies (2018) perusteella pitää todeta, että on olemassa käsinkosketeltavia tietojärjestelmiä. On siis erilaisia yhdistelmiä käsinkosketeltavissa osien ja tietokoneistettujen osien välillä. Suunnittelussa pitää siis huomioida sekä käsinkosketeltavat osat että tietokoneistetut osat.



Lyhyt katsaus käytettävyyden testaamiseen

Cooper (1999, sivu 79) perusteella pitää todeta, että erilaisissa määrittelydokumenteissa voi olla sivukaupalla erilaisia toimintojen listoja, joita kehitettävän ohjelman pitäisi toteuttaa. Yleisesti Cooper (1999) perusteella pitää todeta, että kymmenien sivujen mittaiset toimintojen listat eivät ole paras tapaa esittää erilaisia määrittelyjä.

Cooper (1999, sivu 80) perusteella pitää todeta, että käyttäjät eivät juuri välitä toiminnoista, jolloin käyttäjät haluavat oikeasti päästä tavoitteisiinsa. Tässä on hyvin suuri ero: erilaiset toiminnot vai oikeat tavoitteet? Käyttäjän tavoitteiden oikea ymmärtäminen (siis oikein) pitäisi siis olla keskeinen osa vuorovaikutussuunnittelua.

Tässä kohtaa on hyvä todeta, että ohjelmoijat voivat haluta toimintojen suunnittelua. Tavoitteiden ymmärtäminen on paljon vaikeampaa, jolloin voidaan esitellä kasa toimintoja, joita ohjelmistoa käyttävät ihmiset eivät ymmärrä omien tavoitteidensa kannalta.

Pitäisikö tehdä prototyyppijä? Cooper (1999) selittää sivuilla 88-92, että monesti ohjelmistojen prototyyppit määrätään johdon puolelta tehtävän ohjelman perustaksi. Eli prototyypin kehittäminen voi olla huono lähestymistapa, jos epäselvä prototyyppi määrätäänkin johdon puolesta ohjelmoinnin perustaksi. On näköjään parempi tapa lähteä liikkeelle suunnittelusta kuin prototyypistä.

Suunnittelu ottaa tuotteen joka (a) voidaan rakentaa ja toimii hyvin, (b) voidaan levittää ja myydä voitolla, ja tekee siitä menestyksekkään (c) tekemällä siitä jotain, mitä ihmiset todella haluavat. Tämä kolmas vaihe tuo tasapainon ja muuttaa kiinnostavan teknisen saavutuksen pitkäaikaiseksi menestykseksi. (Cooper 1999, sivu 108)

Eli suunnittelussa pitäisi ymmärtää, että mitä ihmiset oikeasti haluavat. Ohjelmisto on yksi tapa ratkaista asioita, joita ihmiset oikeasti haluavat.

Cooper (1999) selittää sivulla 127 tarkemmin, että on valtava ero toiminnallisuudelle suunnittelemisen ja ihmisille suunnittelemisen välillä.

Cooper (1999) selittää sivulla 169 tarkemmin, että sana ”käyttäjä” aiheuttaa ongelmia. Monesti käyttäjän oletetaan sopeutuvan ohjelmistoon – ei siis päinvastoin. Monesti ihmiset joutuvat oikeasti joustamaan ohjelmistojen vuoksi. Tässä on siis iso ero.

Tavoitteet eivät ole sama kuin tehtävät. Tavoite on lopputila, kun taas tehtävä on välittävä prosessi tavoitteeseen pääsemiseksi. On hyvin tärkeää olla sekoittamatta tehtäviä tavoitteisiin, mutta on hyvin helppo erehtyä. (Cooper 1999, sivu 194)

Cooper (1999) on kuvannut kirjassaan eri kohdissa tietokoneohjelman ohjelmointia vuorovaikutussuunnittelun kannalta. Ohjelmointi on hyvin perusteltu näkökulma eri asioihin, ja oikeiden ohjelmoijien oikea johtaminen on hyvin vaativa johtamisen laji. Isoksi haasteeksi näyttää siis nousevan oikeiden ohjelmoijien ja oikeiden vuorovaikutussuunnittelijoiden hyvä yhteistyö.

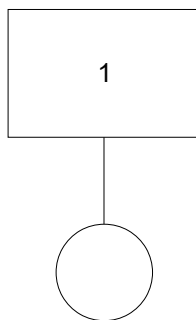
Sivuilla 230-234 Cooper erottelee seuraavat aiheet:

Lyhyt katsaus käytettävyyden testaamiseen

- ikuiset kohtalaiset
- ohjelmoijat suunnittelevat asiantuntijoille
- markkinoijat suunnittelevat aloittelijoille
- tämä aiheuttaa vastakkaiset käyrät.

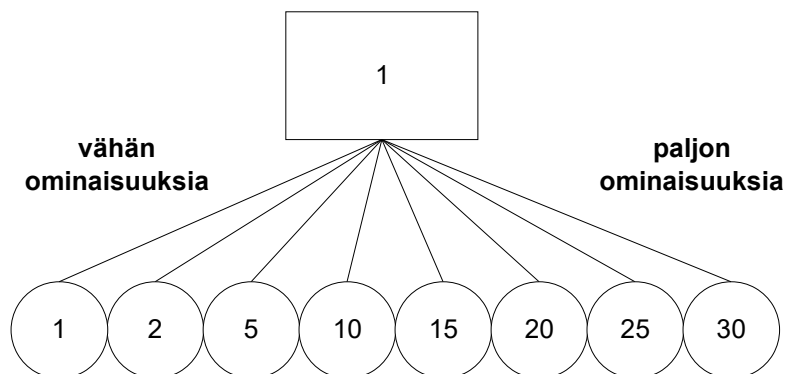
Ikuiset kohtalaiset ovat järjestelmää käyttäviä ihmisiä, jotka käyttävät järjestelmää sen verran paljon/usein, että he eivät ole enää aloittelijoita. Toisaalta ikuiset kohtalaiset käyttävät järjestelmää vähemmän kuin asiantuntija. Suurin osa käyttävistä ihmisistä on kuitenkin ikuisia kohtalaisia, jolloin suunnittelua pitäisi kohdistaa heihin. Ongelma siis on, että ohjelmoijat haluavat suunnitella asiantuntijoille, ja markkinoijat haluavat suunnitella aloittelijoille. Ikuiset kohtalaiset jäävät siis monesti tunnistamatta. Tässä on jälleen kerran yhden ristiriidan aihe (yksi monista) tietokoneohjelman kehityksen suhteen.

Itse olen todennut, että monesti päädymme kehittämään vain yhtä suurta käyttöliittymää suunnattuna samanaikaisesti kaikille käyttäjille. Tämä johtaa asiantuntijoiden kannalta lukuisien valintojen tekemiseen, jolloin asiantuntija voi toistaa samoja valintoja kymmeniä kertoja. Toisaalta ikuiset kohtalaiset eivät tarvitse enää niin paljon apua opittuaan tietokoneohjelman perusvalinnat.



Tarvitsemme useamman vaihtoehdon, jolloin huomioidaan ikuiset kohtalaiset, aloittelijat ja asiantuntijat. Asiantuntijat tarvitsevat nopeat oikopolut ja lyhyet valinnat moneen toimintoon, jolloin heille esitetään joukko yleisempiä valintoja. Aloittelijat voivat käyttää tietokoneohjelmaa avustavalla valinnoilla, jolloin heille selitetään eri toimintoja. Ikuiset kohtalaiset tarvitsevat jo vähemmän selitystä, koska he ovat jo kokeilleet eri toimintoja muutaman kerran.

Useamman vaihtoehdon mahdollisuuksia olen kuvannut seuraavaan kuvaan. Ominaisuuksien määrä vaihtelee eri vaihtoehtojen suhteen.



Sivuilla 235-236 Cooper (1999) esittelee sanastojen selventämisen tarpeen. Käytännössä käytämme erilaisia termejä, jotka kuvaavat eri asioita. Sanaston selventäminen on hyvin tärkeä vaihe suunnittelun vaihetta. Jonkin sanan (täydellinen?) väärinymmärrys voi johtaa ohjelmointia harhaan ennen jonkin sanan todellista oikein tehtyä määrittelyä.

Sanastosta voidaan käydä hyvinkin laajaa keskustelua eri vaiheissa. Tietysti ohjelmoijat tarvitsevat hyvin tarkat määrittelyt eri sanoille, jotta he voivat ohjelmoida oikeaa asiaa. Eri sidosryhmät voivat käydä laajaakin keskustelua sanastoista, joten sanaston kehittämiseen pitää kiinnittää tarpeeksi huomiota.

Vähän on enemmän! Edellä todetun perusteella voi todeta, että ohjelmoijat haluavat täyttää ohjelmiston erilaisilla toiminnoilla. Cooper (1999, sivu 250) toteaa, että vähän on enemmän.

Ohjelmoijan kannalta vähemmän ohjelmointi tarkoittaa vähemmän monimutkaisuutta, vähemmän vikoja, vähemmän mahdollisuuksia toimimattomalle vuorovaikutukselle ja helpompaa ylläpitoa (vrt. Cooper 1999, sivu 251). Oma havainto on, että voimme tehdä paljon vähemmän turhaa ohjelmiston koodia, ja paljon enemmän laadukasta ohjelmiston koodia.

Cooper (1999, sivu 251-252) toteaa, että eri yhtiöt ovat voineet tehdä käytettävyyssuunnittelua ja käytettävyyssuunnittelusta, mutta tulokset ovat voineet olla huonoja huolimatta kaikesta käytettävyyssuunnittelusta. Cooper (1999, sivu 282-283) toteaa, monesti paikalle kutsutaan käytettävyyssuunnittelija, mutta he tulevat monesti mukaan tuotteen ollessa suureksi osaksi rakennettu. Käytettävyyssuunnittelu alkaa siis monesti hyvin loppuvaiheessa, jolloin voi olla jo liian myöhäistä tehdä laajoja uudistuksia.

Yhteenveto: Cooper (1999)

Cooper (1999) on kirjoitettu oman aikanaan, mutta monet hänen mainitsema asiat vuorovaikutussuunnittelusta ovat edelleen hyvin ajankohtaisia virkkeen kirjoitushetkellä 2020-luvulla. Oman arvion mukaan erilaiset ohjelmistoyritykset ovat kyllä kehittäneet ohjelmistojaan, jolloin niihin on vähitellen osattu tehdä erilaisia parannuksia.

On kuitenkin hyvä tehdä edelleen erottelu vuorovaikutussuunnittelun ja käyttöliittymäsuunnittelun välillä. Vuorovaikutussuunnittelussa pitäisi lähteä liikkeelle hyvissä ajoin ennen ohjelmointia. Ohjelmoinnin tuloksena on kyllä erilaisia käyttöliittymiä, joita voidaan sitten arvioida erilaisilla

Lyhyt katsaus käytettävyyden testaamiseen

tavoilla, mutta jo ennen käyttöliittymien ohjelmointia pitäisi tehdä oikeaa vuorovaikutussuunnittelua.

Cooper (1999, sivu 300) toteaa, että ohjelmistojen testaamisen työtä on alettu arvostaa ohjelmoijien keskuudessa, jolloin ohjelmoijien lisäksi ohjelmistohankkeissa on ohjelmistojen testaaajia. Monesti kokeneet testaajat voivat löytää enemmän virheitä kuin pelkästään ohjelmoijat. Cooper (1999) perusteella pitää kuitenkin todeta, että hän ei ole kuvannut laajasti käytettävyyden testaamista. Tosin hän sanoo kirjassaan edustavansa vuorovaikutussuunnittelua, joka on eri asia kuin käytettävyyden suunnittelu.

Mitä voisimme huomioida käytettävyyden testaamisen kannalta?

Peruskirja 2: Krug (2006)

Krug (2006) perusteella pitää todeta, että käytettävyyssuunnittelu on saanut paljon kannatusta eri vaiheissa, jolloin hän kirjoittaa (jo!) perinteiseksi muodostuneesta testauksesta. Krug (2006, sivu 137) esittelee uudenlaisen testauksen.

	Perinteinen testaus	Uudenlainen testaus
Käyttäjien määrä testiä kohti	Yleensä vähintään 8, koska alkukustannukset ovat suuret	3 tai 4
Koehenkilöiden etsiminen	Valitaan huolellisesti kohderyhmän mukaan	Melkein kuka tahansa Webiä käyttävä kelpaa
Testaustila	Käytettävyyslaboratorio, jossa on tutkimushuone ja peili-ikkuna	Mikä tahansa toimisto- tai neuvotteluhuone
Testaaja	Kokenut käytettävyyden ammattilainen	Kuka tahansa kohtuullisen kärsivällinen ihminen
Suunnittelu	Aikataulu on tehtävä viikkoja etukäteen, jotta voidaan varata laboratorio ja etsiä koehenkilöt	Testit voidaan tehdä milloin vain, joten tarkkoja aikatauluja ei tarvita
Valmistelut	Laaditaan testisuunnitelma, keskustellaan siitä ja tehdään muutoksia	Päätä, mitä haluat osoittaa
Mitä ja milloin testataan?	Ellei budjetti ole valtava, ota riski ja suorita testi, kun sivusto on melkein valmis	Tee pieniä testejä koko kehitysprosessin ajan
Kustannukset	5 000-15 000 euroa (tai enemmän)	Enintään 300 euroa (50-100 euron palkkio kullekin koehenkilölle)
Toimenpiteet	20-sivuinen kirjallinen raportti tulee viikon kuluttua, ja kehitystiimi kokoontuu päättämään muutoksista	Kehitystiimin jäsenet (ja muut asianosaiset) keskustelevat lounaalla samana päivänä

Lyhyt katsaus käytettävyyden testaamiseen

195

196 Yksi keskeinen ero perinteisellä testauksella ja uudenaikaisella testauksella on aikataulut. Krug (2006,
197 sivu 159 sisältää enemmän asiaa) ehdottaa tehtäväksi uudenaikaista testausta säännöllisesti kerran
198 kuukaudessa. Itse olen kutsunut tätä kevyeksi testaamiseksi, jolloin kaikki järjestelyt ovat hyvin
199 kevyitä verrattuna perinteiseen testaamiseen. Perinteisessä testaamisessa järjestelyt ovat aika
200 raskaita erillisten käytettävyydelaboratorioiden kanssa toteutettuna, jolloin testejä tehdään
201 harvemmin.

202

203 Krug (2006) perusteella pitää todeta, että uudenaikaisessa testauksessa voidaan aina yhdessä testissä
204 havaita kolme käytettävyyden suurinta ongelmaa, jotka pyritään ratkaisemaan testauksen jälkeen.
205 Tässä on eroa perinteiseen käytettävyyden testaamiseen, jossa yritetään kuvata useampia
206 mahdollisia ongelmia yhdellä kerralla.

207

208 3→3→3→ jne. Eli jokaisella testaamisen kerralla päätetään toimenpiteistä kolmen suurimman
209 ongelman havaitsemisen jälkeen.

210

211 Eli nopeat ja kevyet testit ilman laajoja kirjallisia raportteja voi olla hyvä lähestymistapa.

212

213 Omasta kokemuksesta pitää todeta, että yhdellä kurssilla (käytettävyyden perusteet Tampereella)
214 teimme (minä ja toinen opiskelija, nimimerkki HH) käytettävyydestestausta
215 käytettävyydelaboratoriossa kahdelle henkilölle kurssin vaatimusten mukaisesti. Näin jälkikäteen
216 ajatellen erillisen käytettävyydelaboratorion käyttö ja erilaiset tutkimusraportit käytettävyydestä
217 olivat melkoisen raskaita järjestelyitä.

218

219 Perinteiseksi muodostunutta testausta voidaan siis arvioida eri tavoin. Erilaisissa peruskoulutuksissa
220 voidaan siis kouluttaa perinteistä testausta.

221

222 Peruskirja 3: Sinkkonen ym. (2006) → Liite: Käytettävyydestesti

223

224 Sinkkonen ym. (2006) kirjoittama liite käytettävyydestestistä on hyvin mielenkiintoinen.

225

226 Sinkkonen ym. (2006, sivu 277) toteavat, että käytettävyydestestissä testikäyttäjät tekevät (yleensä)
227 yksi kerrallaan testitarinan mukaisia, työtehtävänsä kaltaisia tehtäviä. Sinkkonen ym. (2006)
228 toteavat, että testit pitää tallentaa. Tämän jälkeen on mahdollista arvioida käyttöliittymässä olevat
229 ongelmat ja mahdollisesti suositellaan korjaustapa.

230

231 Sinkkonen ym. (2006, sivu 279) toteavat, että yhden järeän testin sijaan kannattaa tehdä monta
232 pientä testiä; eli saman budjetin kannattaa käyttää moneen peräkkäiseen testiin, joiden välissä
233 korjataan esille tulleet ongelmakohdat.

234

235 Sinkkonen ym. (2006, sivu 281) toteavat, että testin järjestäminen käsittää ainakin seuraavat
236 työvaiheet.

237

- 238 • Testin tavoitteiden selvittäminen.
- 239 • Käytettävyyksivaatimusten selvittäminen.
- 240 • Tuotteeseen tutustuminen.
- 241 • Testattavien toimintojen valinta.

Lyhyt katsaus käytettävyyden testaamiseen

- 242 • Testikäyttäjien määrä ja valinta.
- 243 • Testitarinan ja -tehtävien valinta.
- 244 • Testausmenetelmän valinta.
- 245 • Muita järjestelyasioita.
- 246 • Koulutustarve ennen testiä
- 247 • Käsikirja mukaan vai ei?
- 248 • Tarvitaanko testiin interaktiivinen tilanne (esimerkiksi joku näyttlee asiakasta ja soittaa
- 249 puhelimella
- 250 • Pilottitestin järjestäminen
- 251 • Testijärjestelyistä on hyvä tehdä testaussuunnitelma.

252

253 Sinkkonen ym. (2006, sivu 284) huomio testin suorittajien määrää. Vakavimmatkin virheet löytyvät
254 yleensä 3-4 käyttäjällä. Käytettävyydestin suorittajien määrän lisääminen lisää löydettyjen
255 käytettävyysongelmien määrää.

256

257 Käyttäjien valintaan vaikuttavat monet syyt kuten käyttäjien tuntemus testattavaan järjestelmään
258 nähden. (Sinkkonen ym. 2006, sivu 284)

259

260 Testausmenetelmiä ovat seuraavat (Sinkkonen ym. 2006, sivut 284-287):

261

- 262 • Ääneen ajattelu.
- 263 • Paritestit.
- 264 • Yhteisläpikäynti.
- 265 • Jälkikäteen haastattelu
- 266 • Pikkutestit, osatestit, ”kahden paperin” testit, käsitelistat.
- 267 • Ryhmäläpikäynti
- 268 • Vapaa läpikäynti.

269

270 Käytettävyydestin suorittajan pitää ilmeisesti valita paras menetelmä eri tilanteisiin → Jukka
271 Rannilan huomio.

272

273 Tyypillinen raportti käsittää seuraavat luvut: (Sinkkonen ym. 2006, sivut 292):

274

- 275 • Tuotteen käyttöliittymän ja mahdollisesti käyttötavan lyhyt kuvaus.
- 276 • Lyhyt kuvaus testaustavasta ja testikäyttäjistä.
- 277 • Testattavat toiminnot tai testitehtävät.
- 278 • Testin tulos
- 279 • Testin ohjaajan lausunto tuotteesta sekä yhteenveto testistä.

280

281 Raportin laajuus on 20-200 sivua testityypistä riippuen. Raporttiin kerätään havaitut ongelmat,
282 niiden syyt ja esiintymismäärä ongelmalajeittain. Testiraportti vaatii usein yhteisen läpikäynnin
283 suunnittelijoiden kanssa, jolloin voidaan varmistaa, että asiat on ymmärretty molemmin puolin ja
284 katsoa testin ”kootut tilanteet”. (Sinkkonen ym. 2006, sivun 292 perusteella)

285

286 Käytettävyydestiin voi kulua kalenteriaikaa muutamasta päivästä muutamaa kuukauteen riippuen
287 testaustavasta ja testin tavoitteista. Jos testataan samaa tuotetta eri tuotekehityksen vaiheessa,
288 testiajat lyhenevät merkittävästi. (Sinkkonen ym. 2006, sivun 293 perusteella)

Lyhyt katsaus käytettävyyden testaamiseen**Sinkkonen ym. (2006, siis liite): Yhteenveto**

Varmaankin voi todeta, että Sinkkonen ym. (2006, liite) kuvaa hyvin käytettävyydestä liittyvää ideologiaa. Sinkkonen ym. (2006, liite) kannattaa verrata Krug (2006) esittämään uudelleenlaiseen käytettävyydestä (taulukko Krug (2006) perusteella on esitelty aikaisemmin).

Sinkkonen ym. (2006, liite) kuvaa hyvin käytettävyydestä suorittamista, jolloin tuotetaan erilaisia raportteja käytettävyydestä käytettävyydestä perusteella. Krug (2006) toteaa uudelleenlaista käytettävyydestä, että erillisiä käytettävyyden testauksen raportteja ei tarvita. Tässä on hyvinkin merkittävä ero.

Vuorovaikutussuunnittelua vai käytettävyyssuunnittelua?

Joissain lähteissä Cooper (1999) on mainittu, mutta mielestäni lainaukset ovat olleet hyvin ylimalkaisia. Cooper (1999) esittelee vuorovaikutussuunnittelua, joka on mielestäni hyvin erilaista kuin jo perinteeksi muodostunut käytettävyyden suunnittelu ja käytettävyyden testaus.

Cooper (1999) ei kuitenkaan esittele laajasti erilaisten liittymien testaamista, jolloin Krug (2006) esittämä uudelleenlaista testaus voisi olla hyvä lähtökohta erilaisiin asiayhteyksiin.

Tämä esitelmä on hyvin rajoittunut!

Tässä esitelmässä oli siis kolme peruslähdetä: Cooper (1999); Krug (2006); Sinkkonen ym. (2006).

Alkuperäisen esitelmän ideassa oli tarkoituksena tehdä katsaus käytettävyyteen asiaan liittyvän suomenkielisen kirjallisuuden perusteella. Muun kirjallisuuden selaamisen perusteella päädyin kuitenkin pohtimaan erityisesti käytettävyyden testausta. Eri lähteissä kyllä kuvataan käytettävyyttä eri tavoin, mutta käytettävyyden testaamisen kuvaus jää melko vähäiseksi. Krug (2006) sekä Sinkkonen ym. (2006) kuvaavat käytettävyyden testaamista enemmän verrattuna muihin lähteisiin.

Yksi yhteenveto: Mielipidekirjoitus 44

Yleisenä yhteenvetona pitää todeta kirjoittamani mielipidekirjoitus 44.

http://www.jukkarannila.fi/mielipidekirjoitukset.html#nro_44

44 / Yhden liittymän uskon ylittäminen?**ILKKA - 12. syyskuuta 2012**

Kari Hokkanen ihmetteli kolumnissa (Ilkka 8.9.), että Suomen kaltaisessa tietoteknologian kärkimaassa julkisen hallinnon tietojärjestelmät ovat jälkeensä.

Vasta 10.6.2011 on annettu laki julkisen hallinnon tietohallinnon ohjauksesta (tietohallintolaki). Tietohallintolaki vaatii laatimaan kokonaisarkkitehtuurin muutaman vuoden sisällä. Vähitellen saadaan julkisen hallinnon järjestelmien kehittämiselle vastuut.

Lyhyt katsaus käytettävyyden testaamiseen

Julkiseen tietohallintoon voidaan pakottaa yhdet standardit, joilla saadaan yhteensopivuus eri järjestelmiin. Järjestelmiä on kehitetty eri tavoin erilaisilla standardeilla.

Yksi harmistus on terveydenhuollon tietojärjestelmien viidakko, jota esimerkiksi Lääkäriliitto on tutkimuksissa arvostellut. Järjestelmiä on paljon, ja yhtenäistä näkemystä potilaasta ei saa välittömästi, ja joudutaan usean järjestelmän käyttöliittymän suohon.

Ihmeenä on pidetty Viron järjestelmää, ja Suomestakin on käyty tutustumassa järjestelmään. Lääkärilehti uutisoi, että Viron järjestelmässä "klikkausten ja nappuloiden määrä on ohjelmassa minimoitu" ja "sairaalassa voidaan itse laatia ns. dynaamisia dokumentteja, jolloin työstä johtuviin muutoksiin ei aina tarvita ohjelmistotalon apua".

Omassa tutkimuksessa tulini samaan tulokseen: tehokäyttö vaatii riisuttuja käyttöliittymiä ja toisaalta erilaisia riisuttuja käyttöliittymiä pitää ajaa eri käyttäjäryhmille.

Järjestelmäkehityksessä ideologia on päinvastainen. Ideologiana on kehittää yksi iso liittymä kaikille ryhmille samalla käyttöasteella. Tosiasiassa väki uupuu kymmenien klikkausten suohon.

Ideologiana ovat tietotekniikka-asiantuntijoiden kehittämät liittymät.

Tietotekniikka-asiantuntijat eivät koskaan pysty selvittämään kaikkien käyttäjäryhmien ajatusta käyttöliittymille, jolloin Viron esimerkin mukaisesti pitää olla mahdollisuus käyttäjien kehittämiin riisuttuihin käyttöliittymiin.

Ideologiana yksi iso liittymä on tarttunut oppilaitosten seiniin. Yhden liittymän uskon ylittäminen on vallankumouksellista ja koko tietotekniikka-alan ja asiakkaiden työjako muuttuisi täydellisesti.

Tietotekniikka-asiantuntijoiden pitäisi luopua yhden ison käyttöliittymän uskosta. Historia osoittaa uskonpuhdistuksen vievän paljon aikaa.

Ennen tietotekniikka-alan uskonpuhdistusta saamme vielä pitkään yhden ison käyttöliittymän raskaasti käytettäviä järjestelmiä - valitettavasti.

Jukka Rannila
Jalasjärvi

Yhden liittymän uskon ylittäminen?

Loppujen lopuksi pitää todeta, että vain yhden ison liittymän kehittäminen johonkin järjestelmään on hyvin yleinen toimintapa. Olisi kuitenkin mahdollista kehittää useampi liittymä erilaisille ihmisille, jotka käyttävät jotain järjestelmää.

Käytettävyyden ideologiaa jonkin verran

Lyhyt katsaus käytettävyyden testaamiseen

382 Alkuperäisen ajatus oli pohtia käytettävyyden ideologiaa. Olen aikaisemmin lukenut Norman
383 (2013). Tätä esitelmää varten selasin Nielsen (1993), mutta en lukenut sitä läpi. Kummatkin kirjat
384 taitavat olla peruskirjoja käytettävyyden asian suhteen.

385

386 Ylijoki (1998) olen viitannut erilaisissa asiayhteyksissä. Oman havainnon mukaan ihmiset käyvät
387 koulutuksen aikana läpi erilaisia sosiaalisia (vrt. Ylijoki 1998) myllytyksiä, jolloin he omaksuvat
388 erilaisia näkökulmia. Loppujen lopuksi oma näkökulma voikin jäykistyä, vaikka eritasoisten
389 koulutusten virallisena tavoitteena on näkökulmien laajentaminen.

390

391 Näyttäisi siltä, että käytettävyysskin on täysin oma ideologiansa. Oman opiskelun perusteella pitää
392 todeta, että en ole käynyt laajasti läpi käytettävyyden peruslähteitä ja tehnyt paljon käytettävyyden
393 (työ)tehtäviä. Käytettävyysskin tarkoittaa erilaisia sosiaalisia myllytyksiä, joiden perusteella ihmiset
394 oppivat käytettävyyden erilaiset näkökulmat.

395

396 Yleinen yhteenveto: käytettävyyssuunnittelun ja vuorovaikutussuunnittelun eroavaisuus

397

398 Loppujen lopuksi pitää todeta, että käytettävyyssuunnittelu ja vuorovaikutussuunnittelu eroavat
399 toisistaan. Eli tässäkin kohtaa tulee vastaan kahden erilaisen ideologian eroavaisuudet.

400

401 Tämä kirjoitus tarjosi vain hyvin rajattuja näkökulmia hyvin vaikeaan asiaan.

402

403 Tämän tekstin perusteella käytettävyyssuunnittelun ja vuorovaikutussuunnittelun asiaa pitää seurata
404 muiden asioiden seurannan ohessa.

405

406

407 Lähteitä

408

409 Cooper, A. (1999). Nörttien valtakunta: Miksi korkeateknologiatuotteet saavat meidät sekaisin ja
410 kuinka palauttaa järki (R. Parkkonen, Käänt.). Helsinki: Suomen Atk-kustannus.

411

412 Krug, S. (2006). Älä pakota minua ajattelemaan! Tervejärkinen käsitys web-käytettävyydestä (2.
413 laitos; V.-P. Ketola, Käänt.). Helsinki: Readme.fi.

414

415 Lederman, R., & Beynon-Davies, P. (2018). Understanding the Tangible within Information
416 Systems. ICIS 2018 Proceedings, 1. Noudettu osoitteesta

417 <https://aisel.aisnet.org/icis2018/general/Presentations/1/>

418

419 Nielsen, J. (1993). Usability engineering. Boston: AP Professional.

420

421 Norman, D. A. (2013). The Design of Everyday Things: Revised and Expanded Edition. New York:
422 Basic Books.

423

424 Sinkkonen, I., Kuoppala, H., Parkkinen, J., & Vastamäki, R. (2006). Käytettävyyden psykologia (3.
425 uudistettu painos). Helsinki: Edita, IT Press.

426

427 Ylijoki, O.-H. (1998). Akateemiset heimokulttuurit ja noviisien sosialisatio. Tampere: Vastapaino.

428

429 **Liite 1: Lisenssi**430 **Nimeä-Epäkaupallinen-Ei muutoksia 4.0 Kansainvälinen**

431

432 Avoimesti lisensoitu teos

433

434 Tämä teos on lisensoitu Nimeä-Epäkaupallinen-Ei muutoksia 4.0 Kansainvälinen -lisenssillä.

435 Teoksen uudelleen käytön yhteydessä pitää mainita kirjoittaja. Valittu lisenssi tarkoittaa, että

436 teoksen sisältö on vapaasti käytettävissä, kunhan alkuperäislähteeseen viitataan.

437

438 Lisenssin kansantajuinen esitys on seuraavalla www-sivulla:

439

440 <https://creativecommons.org/licenses/by-nc-nd/4.0/deed.fi>

441

442



443

444 NIMI:

445 Teoksen tekijä on ilmoitettava siten kuin tekijä tai teoksen lisensoija on sen määrännyt (mutta ei

446 siten että ilmoitus viittaisi lisenssinantajan tukevan lisenssinsaajaa tai teoksen käyttötapaa).

447

448 Ei muutettuja teoksia

449 Teosta ei saa muuttaa, muunnella tai käyttää toisen teoksen pohjana.

450

451 Epäkaupallinen

452 Lisenssi ei salli teoksen käyttöä ansiotarkoituksessa.

453

454 Lisenssin perusteellinen juridinen esitys on seuraavalla www-sivulla:

455

456 <https://creativecommons.org/licenses/by-nc-nd/4.0/legalcode.fi>

457

458

459

460

461 [jatkuu seuraavalla sivulla]

462

463

464 Liite 2: Vastuulausekkeet

465

466 Yksittäisen ihmisen yksittäistä tulkintaa yksittäisistä asioista / aiheista

467

468 Tämä teos on yksittäisen ihmisen tulkintaa eri asioista / aiheista, eikä edusta minkään (rekisteröidyn tai
469 rekisteröimättömän) yhteisön virallista tai epävirallista kantaa. Tässä teoksessa mainitut mielipiteet eivät ole
470 (lainopillisia) neuvoja, ja lukijoita kehoitetaan itse perehtymään huolellisesti tässä teoksessa mainittuihin asioihin /
471 aiheisiin.

472

473 Tämä teos ei kata tulevaisuuden kehittymistä, jolloin tässä teoksessa mainitut ennustukset voivat osoittautua
474 vääräksi. Vastaavalla tavalla tämä teos ei kata menneisyyden tulkintaa, jolloin tässä teoksessa tehdyt arviot
475 menneisyydestä voivat osoittautua vääräksi.

476

477 Poliittisia vastuulausekkeitä

478

479 Tämä teos käsittelee useita poliittisia mielipiteitä erilaisista asioista / aiheista. Nämä mielipiteet eivät kuitenkaan
480 ole virallisia neuvoja poliittisen päätöksen perustaksi. Teoksen lukijoita kehoitetaan lukemaan kukin mielipide
481 yksittäisen henkilön ajatuksena, koska esitetyt mielipiteet eivät ole minkään yksittäisen puolueen (rekisteröity tai
482 rekisteröimätön) virallisia mielipiteitä. Lisäksi esitetyt poliittiset mielipiteet eivät edusta minkään puoleen
483 (rekisteröity tai rekisteröimätön) jäsenjärjestön (rekisteröity tai rekisteröimätön) virallista kannanottoa.

484

485 Teoksessa esitetyt poliittiset mielipiteet eivät kata Suomen, Euroopan tai maailmanlaajuisen politiikan
486 menneisyyttä tai tulevaisuutta, ja ovat vain yksittäisen henkilön yksittäisiä mielipiteitä.

487

488 Teoksessa esitetyt poliittiset mielipiteet eivät ole tarkoitettu virallisen tai epävirallisen ehdokkuuden tukemiseksi
489 missään vaalissa millään tasolla, eli teoksessa esitetyt poliittiset mielipiteet eivät ole virallisiin vaaleihin
490 valmistautuvan virallisen tai epävirallisen ehdokkaan mielipiteitä. Mahdollisissa virallisissa vaaleissa (teoksen
491 julkaisun jälkeen) virallisena ehdokkaana esitetyt poliittiset mielipiteet ovat oma kokonaisuutensa, ja virallisissa
492 vaaleissa (teoksen julkaisun jälkeen) virallisen ehdokkaan julkiset mielipiteet ovat tämän teoksen ulkopuolella, ja
493 tämä teos ei ennakoit tulevia mahdollisia poliittisia mielipiteitä virallisissa vaaleissa (teoksen julkaisun jälkeen).

494

495 Viitattujen www-sivujen sisältö

496

497 Tässä teoksessa viitataan erilaisiin www-sivuihin. Viitattujen www-sivujen laillinen sisältö on tarkistettu tämän
498 teoksen julkaisuhetkellä, mutta monen viitatus www-sivu sisältö tulee mahdollisesti muuttumaan tämän teoksen
499 julkaisun jälkeen. Kaikki muutokset viitatuilla www-sivuilla ovat viitattujen www-sivujen omistajien / ylläpitäjien
500 vastuulla. Kaikki uusi laillinen ja/tai laitton sisältö viitatuilla www-sivuilla ei ole tämän teoksen kirjoittajan
501 vastuulla, ja tämän teoksen lukijoita kehoitetaan huolellisesti välttämään www-sivuilta ladattavien laittomien
502 sisältöjen käyttöä.

503

504 Kaupallinen sisältö / Yleishyödyllisyys

505

506 Tämä teos ei sisällä kaupallista sisältöä, eikä tätä teosta ole tarkoitettu kaupalliseksi sisällöksi, ja käytetyn
507 lisenssin mukaisesti tämä teos on tarkoitettu ei-kaupalliseksi sisällöksi. Tämä teos ei sisällä kaupallisen yhteisön
508 (rekisteröity tai rekisteröimätön) liike- tai ammattisalaisuuksia.

509