

# Käytettävyyystutkimuksen menetelmien vertailu

# 20

*Katri Kosonen*

Käytettävyyystutkimuksen menetelmiä on nykyään iso joukko. Mitä niistä sitten pitäisi käyttää? Tavoitteena olisi löytää menetelmä, joka paljastaa paljon suunnitteluvirheitä mutta on helppo ja nopea soveltaa. Käytettävyyden arvioinnissa esimerkiksi heuristisen arvioinnin avulla voidaan löytää paljon pieniä, kosmeettisia ongelmia, mutta on vaikea samalla löytää käyttäjän kannalta vaikeimmat käytettävyysskatastrofit. Tässä luvussa käydään läpi käytettävyyystutkimusmenetelmien vertailusta tehtyjä tutkimuksia. Tutkimusten kirjo on laaja ja mitään standardeja esimerkiksi menetelmien vertailun kriteereistä ei vielä ole olemassa. Menetelmien vertailussa pääkriteereinä voisivat olla kattavuus, pätevyys sekä hyödyllisyys. Menetelmien soveltamisella saatavien tulosten vertailu on vaikeaa ilman tietoa standardiongelmajoukosta, johon kuuluu kaikki arviointikohteena olevan sovelluksen ongelmat. Kriteerien ja standardiongelmajoukon avulla saadaan tietoja menetelmien ominaisuuksista ja tietojen perusteella voidaan verrata menetelmiä toisiinsa. Käytännön tuotekehityksessä asiantuntijoilla on omat suosikkimenetelmänsä. He ovat ehkä käytännössä nähneet, mitkä menetelmät toimivat, tai kokevat menetelmistä jotkut tietyt omille työtavoilleen läheisiksi. Heidän suosikkejaan ovat käytettävyytestaus, prototyypin tekeminen ja heuristinen arviointi. Käytännössä menetelmien vaatimat resurssit vaikuttavat suuresti menetelmän valintaan. Usein tuotekehityksessä toimivat käytettävyyssasiantuntijat joutuvatkin tekemään päätöksiä kulu-hyöty –akselilla, tietoisesti hyväksymään huonommat tulokset tai kalliin hinnan.

## 20.1. Johdanto

Käytettävyyystutkimuksen menetelmien vertailu on noussut melko tärkeään asemaan, koska käytettävyyystutkimusmenetelmiä alkaa olla melko paljon. Jotta näistä menetelmistä voitaisiin valita sopivimmat kuhunkin tarkoitukseen, pitää menetelmiä jollain tavalla verrata. Vertailuun on olemassa muutamia tapoja. Tähän mennessä tehtyjen vertailujen kirjo on melko laaja ja usein nämä tutkimukset eivät ole vertailtavissa keskenään niiden erilaisten kriteerien tai toteutustapojen takia.

Kun menetelmiä lähdetään vertailemaan, niin ensimmäisenä pitää päättää, minkälaisia menetelmiä vertaillaan. Voidaan vertailla sellaisia menetelmiä, jotka soveltuvat pelkästään suunnitteluvaiheeseen tai formatiivisiin testeihin, tai menetelmiä, joita voi käyttää yksin. Vaihtoehtoja on suuri määrä.

Kun menetelmiä vertaillaan, tarkoituksena on yleensä löytää paras menetelmä, mutta miten paras menetelmä tulisi määritellä? Halutaanko, että menetelmä olisi nopea, löytäisi paljon ongelmia tai vakavimmat ongelmat vai ettei menetelmän käyttäjän tarvitsisi omata erikoisia taitoja? Vaihtoehtoja on lukematon määrä ja eri henkilöt pitävät eri asioita tärkeinä. On suuri

ero siinä, kuka asettaa kriteerit. Tuotekehityksessä käytetyt kriteerit arviointimenetelmien vertailuun ovat erilaiset kuin tutkijoilla.

Useat tutkijat, kuten Hartson ja muut (2003), ovat sitä mieltä, että ongelmien löytäminen ja niiden korjaaminen tulee erottaa eri prosesseihin. Näiden tutkijoiden mielestä menetelmiä vertailtaessa tulee siis vertailla ainoastaan sitä, millä menetelmällä löydetään ongelmat. Wixon (2003) ehdottaa, että ongelmien löytämistä ja korjaamista pohdittaisiin yhdessä. Hän korostaa sitä, että pitäisi löytää menetelmä, jolla voidaan parantaa tuotetta pienimmällä työllä.

Kun on päästy siihen asti, että tiedetään menetelmät ja kriteerit, joita halutaan käyttää vertailussa, pitää päättää, miten itse vertailu suoritetaan. Tämä onkin avainkysymys menetelmien vertailussa.

Vertailun teossa osa tutkijoista vertaa menetelmiä toisiinsa. Joidenkin mielestä pitäisi pystyä löytämään todellinen tilanne, jolloin saataisiin tietää yksittäisen menetelmän kyvyt todellisessa tilanteessa. Toiset käyttävät testeissään todellisia sovellusalueen tuntijoita ja toiset turvautuvat opiskelijoihin testikäyttäjinä. Tällaiset erot tutkimusten toteutuksessa tekevät tutkimuksista vertailukelvottomia.

Kun sovelluksen käytettävyyttä arvioidaan, on arvioinnin tarkoituksena usein parantaa sovelluksen laatua. Tällöin tarkoitus olisi siis löytää suunnitteluvirheitä ja korjata niitä. Käytännössä on vaikeaa löytää käytettävyysongelma ja vielä korjata se niin, että sovelluksen ongelmien määrä vähenee. Kohdassa 20.2 kuvataan Johnin ja Marks (1997) esittämä tehokkuuspuu, visualisointi, joka on hyvä havainnollistus tuotekehityksen kulusta. Tehokkuuspuussa kuvataan askeleet tavoiteltuun vaihtoehtoon, joka siis on onnistunut todellisen käytettävyysongelman löytäminen ja korjaaminen oikein niin, että tuloksena on vähemmän ongelmia. Tavoitellun vaihtoehdon lisäksi puussa on 6 muuta mahdollista vaihtoehtoa, yhtä lailla seurauksia käytettävyyssiantuntijan tekemistä valinnoista. Menetelmien vertailulla voidaan yrittää löytää käytettävyystudkimusmenetelmistä ne, jotka helpottavat käytettävyyssiantuntijaa tässä tehtävässä.

Käytännön menetelmävalinnoissa, joita tekevät kaikki käytettävyyssiantuntijat, ei tarvita välttämättä testejä selvittämään menetelmien paremmuutta. Menetelmiä voidaan verrata myös yksinkertaisemmin listaamalla menetelmien ominaisuuksia ja pohtimalla minkälaiset menetelmät sopisivat kuhunkin tilanteeseen.

Kirjallisuudesta löytyy suhteellisen paljon tutkimuksia, jotka vertailevat käytettävyystudkimusmenetelmiä. Tutkimusten tulokset voivat olla kiistanalaisia, mutta ne antavat kuitenkin jonkinlaisen kuvan käytettävissä olevista menetelmistä. Tässä luvussa esitellään muutama tällainen tutkimus, joten lukija saa kuvan niistä.

Menetelmien vertailu on monimutkaista. Vertailun tekemisessä on monta asiaa, jotka pitää ottaa huomioon. Kun mitään standardeja ei ole vielä olemassa, pitää jokaisen tutkijan aina ajatella vertailun yksityiskohdat omista lähtökohdistaan. Toisaalta on olemassa myös tutkimuksia, jotka yrittävät luoda näitä standardeja. Työn loppupuolella kuvataankin Hartsonin ja muiden (2003) kehittämä vertailutapa, joka voisi viedä arviointimenetelmien vertailua oikeaan suuntaan. Ehkä näin voitaisiin saada standardeja menetelmien vertailuun.

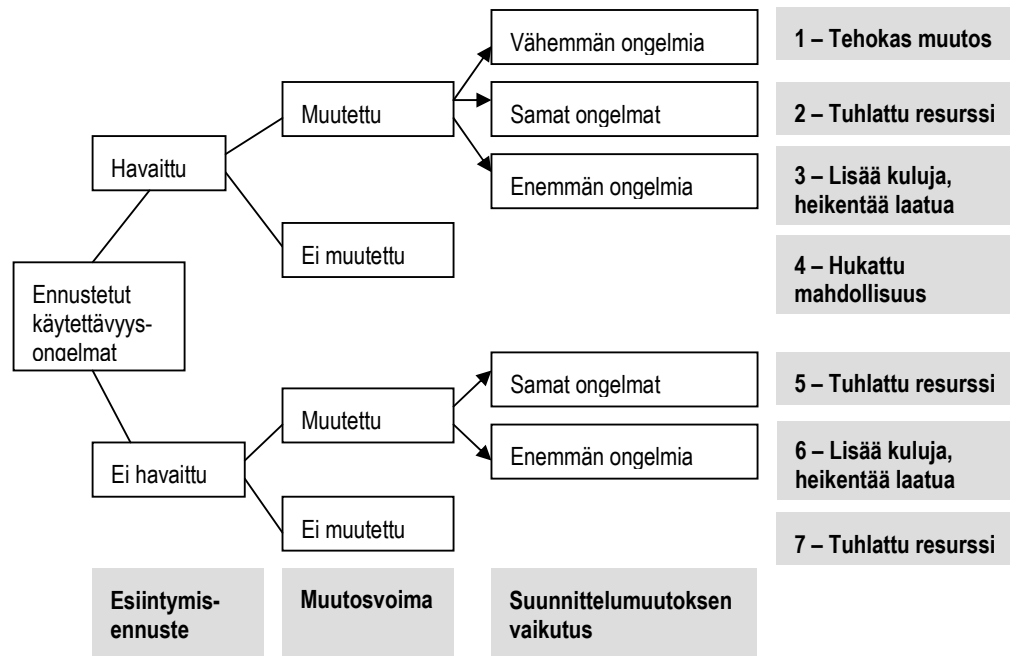
## 20.2. Tehokkuuspuu

*Tehokkuuspuu (Effectiveness Tree)* (kuva 1) on esitelty Johnin ja Marks (1997) tutkimuksessa. Käytettävyysongelmia voidaan ennustaa ennen kuin sovellusta on vielä toteutettu. Käyttäjä saattaa kohdata ennustetun ongelman, mutta aina näin ei kuitenkaan käy – esimerkiksi asiantuntija-arvioinnissa huomattua ongelmaa ei havaita käytettävyydestaustuksessa, jossa käyttäjä kokeilee sovellusta. Ennusteen osuvuutta oikeaan, todelliseen käyttäjän kokemaan käytettävyysongelmaan kutsutaan *esiintymisennusteeksi (predictive power)*. Tavallisesti tätä tietoa ei tiedetä ennustettavista ongelmista.

Ennustetut ja havaitut ongelmat voivat motivoida sovelluksen kehittäjiä muuttamaan kohtia, joissa ongelmat esiintyvät. Kaikkia ongelmia ei kuitenkaan lähdetä muuttamaan. Tätä voimaa,

joka vaikuttaa muuttamispäätökseen, kutsutaan tässä *muutosvoimaksi* (*persuasive power*). Siihen tietenkin vaikuttavat myös projektin aikataulut, muutostyön suuruus, sovelluksen kehittäjien innostuneisuus ja moni muu asia. Kuvassa esiintymisennusteen ja muutosvoiman välisissä viivoissa ei esiinny nuolenpäitä, koska nämä kohdat eivät ole kausaalissa järjestyksessä. Tuotekehityksessä ei aina tehdä prototyypille käytettävyydestä tai muuten havainnoida sen käyttöä, ennen kuin päätetään mitkä ennustetuista käytettävyysongelmista korjataan. Menetelmän tulisi kuitenkin houkuttaa sovelluksen kehittäjiä korjaamaan ongelmallisia kohtia, sillä muuten monet käyttäjälle hankaluutta tuottavat suunnitteluvirheet jäisivät korjaamatta. (John & Marks, 1997)

Kun muutos on päätetty tehdä, pitää tietenkin päättää, miten ongelma korjataan. Korjaus voi poistaa ongelman, tilanne voi pysyä samanlaisena tai korjaus voi myös pahentaa tilannetta ja synnyttää uusia ongelmia. Tätä vaihetta kutsutaan *suunnittelumuutoksen vaikutukseksi* (*Design-Change effectiveness*). (John & Marks, 1997)



Kuva 1: Tehokkuuspuu (John & Marks, 1997).

Halutuin lopputulos on tietenkin kaikkein ylin polku puussa, joka päättyy vaihtoehtoon 1, tehokas muutos. Tässä tapauksessa käyttäjät todella kohtaisivat ongelman, jos sitä ei korjattaisi. Ongelma on tarpeeksi merkittävä saamaan sovelluksen kehittäjät korjaamaan sen ja korjaus onnistuu siten, että ongelmat vähenevät.

Tuhoisimmat seuraukset ovat poluilla 3 ja 6. Tällöin ongelman kohtaamisesta riippumatta lähdetään korjaamaan ongelmaa, mutta päädytään alkuperäistäkin huonompaan ratkaisuun. Korjausyritykseen on kulutettu rahaa ja aikaa, ja kaiken lisäksi laatu on huonontunut. Tämä vielä korostuu vaihtoehdossa, jossa käyttäjä ei olisi edes ikinä kohdannut ongelmaa.

Loput tulosvaihtoehdot eivät ole näin vakavia. Vaihtoehto 4 on hukattu mahdollisuus, koska ongelmalle ei tehdä mitään. Tällöin hukataan mahdollisuus korjata ongelma ja parantaa siten laatua. Tulokset 2, 5 ja 7 ovat tuhlattuja resursseja, sillä niihin on käytetty työtä. Kuitenkaan laatu ei ole huonontunut, joten se ei ole lopputuloksen kannalta haitallista.

Tehokkuuspuussa korostuu muutama hyvä huomio. Ongelmien todellisuus, eli se, kohtaa-atko käyttäjät ongelman sovellusta käyttäessään, on merkittävä tekijä arviointimenetelmien vertailussa. Ei ole kenenkään hyödyksi metsästä ja korjata virheitä, jotka eivät häiritse käyttäjiä. Kuten tehokkuuspuustakin näkee, tällaisten virheiden etsiminen ja korjaaminen on vain resurssien hukkausta. Pahimmassa tapauksessa virheiden metsästäys saattaa jopa huonontaa alkuperäistä sovellusta. Jos käytössä on menetelmä, joka antaa mahdollisimman

vähän ei-todellisia ongelmia, voidaan melkeinpä poistaa tehokkuuspuusta kolme alinta tulosta. Näin tehokkaiden muutosten todennäköisyys kasvaa.

Jos käytössä oleva arviointimenetelmä antaa pääosin todellisia ongelmia, kaikki löytyneet ongelmat kannattaisi yrittää korjata. Tätä voidaan vielä tehostaa sillä, että arviointimenetelmä löytää mahdollisimman monta todellisista käytettävyysongelmista. Tällöin hukattujen mahdollisuuksien määrä pienenee.

Kun mahdollisimman moni todellinen ongelma on tunnistettu, on sovelluskehittäjien vuoro tuottaa ongelmille korjausehdotukset. Totta kai sovelluksen kehittäjät tekevät parhaansa ja yrittävät löytää sen korjaavan oikean ratkaisun. Tätä voivat muutamat arviointimenetelmät auttaa ehdottamalla korjausvaihtoehtoja.

Tehokkuuspuu on hyvä havainnollistus siitä, miten monta erilaista mahdollista lopputulosta todellisuudessa on. On aina yllättävää, että se toivottu, paras ratkaisu, jossa virheet vähenevät, onkin vain yksi seitsemästä eri tuloksesta. Tietenkään kaikki tulokset eivät esiinny yhtä usein käytännössä, mutta arviointimenetelmien vertailun avulla voidaan yrittää löytää sellainen menetelmä, joka tukee sen oikean tuloksen saavuttamista.

### 20.3. Käytännön vertailutapoja

Käytännössä menetelmien vertailu on yleensä kevyempää kuin tähän lukuun kootuissa tutkimuksissa. Yksi tavallisimmin tehdyistä vertailutavoista on listata menetelmien hyviä ja huonoja puolia ja pohtia tämän perusteella, mikä menetelmä sopii mihinkin tilanteeseen. Näin saadaan esille eri menetelmien piirteet.

Esimerkkejä tällaisista listoista on Axupin (2004) tekemä vertailutaulukko, jossa ensin kuvataan menetelmä ja sen vaiheet. Tämän jälkeen on listattu menetelmän hyvät ja huonot piirteet. Tällaisessa taulukossa tai listauksessa on tärkeää esittää kunkin menetelmän kuvaus, jotta lukija tietää, mistä menetelmästä on kyse. Vielä tärkeämmäksi asia muuttuu, jos kyseisestä menetelmästä on olemassa monta eri variaatiota. Hyvien ja huonojen puolien listaus on tietenkin listaajan vastuulla. Listausta tulee kulloinkin vain sellaiset piirteet, mitä tämä kyseinen henkilö ajattelee hyvinä tai huonoina piirteinä. Ne eivät siis suinkaan ole ehdottomia listoja.

Menetelmiä voi myös verrata sen mukaan, mihin tilanteisiin ne sopivat. Tällaisessa vertailussa kriteereinä voivat olla muun muassa: tuotekehityksen vaihe, tarvittavat resurssit, menetelmän käyttäjältä vaaditut taidot, arviointiin tarvittava aika ja testikäyttäjien saatavuus. Tällaista vertailua on pohdittu johdantoluvussa. Vastaavia vertailuja on toteutettu muuallakin, esimerkiksi Usabilitynet-sivustossa<sup>1</sup> ja TRUMP-projektin sivuilla<sup>2</sup>.

Menetelmiä voidaan myös vertailla aitojen kokemusten pohjalta, joita tuotekehityksen ammattilaiset ovat saaneet työssään toimiessaan. Vredenburg ja muut (2002) tekivät kyselytutkimuksen käytettävyyssammattilaisten keskuudessa. Tämän alakohdan loppuosa pohjautuu kokonaisuudessaan heidän tutkimukseensa ja se kuvailee kyselystä saatuja tuloksia. Kyselyyn vastasi 100 alan ammattilaista. Vastaajilta kysyttiin, mitkä ovat heidän mielestään onnistuneimpia käyttäjäkeskeisen suunnitteluprosessin menetelmiä ja mitkä ovat myydyimpiä käytettävyyssammattilaisten menetelmiä. Kyselyssä otettiin huomioon kaikki käyttäjäkeskeisen suunnitteluprosessin menetelmät, siis muitakin kuin ne tutkimusmenetelmät, joita käsitellään tässä raportissa. Kolme suosituinta vastausta onnistuneimmiksi suunnitteluprosessin menetelmiksi olivat käytettävyyssammattilaisten (39 % vastaajista), prototyyppien tekeminen ja heuristinen arviointi. Kolme myydyintä olivat puolestaan asiakashaastattelut, paperi- tai muut prototyyppit sekä käytettävyyssammattilaisten.

Vastaajia pyydettiin mainitsemaan useita yleisimmin käyttämiään menetelmiä. Käytetyistä menetelmistä piti valita viisi menetelmää, joilla heidän mielestään on eniten vaikutusta tuotekehitykseen (esimerkiksi käyttäjätyytyväisyyteen, markkinoilla menestymiseen ja kustannus-

<sup>1</sup> <http://www.usabilitynet.org/>

<sup>2</sup> <http://www.usability.serco.com/trump/index.htm>

säästöihin). Tulokset on koottu taulukkoon 1. Menetelmän mainintakerrat suhteutettiin sille annettuihin tärkeysjärjestyssijoituksiin, joista laskettiin painotettu keskiarvo. Taulukon rivit on järjestetty painotetun keskiarvon mukaan. Mitä pienempi keskiarvo, sitä tärkeämpänä menetelmää pidetään käytännössä.

**Taulukko 1:** Yleisesti käytettyjen käyttäjäkeskeisten suunnitteluprosessien menetelmien arviointi niiden tärkeyden ja yleisyyden mukaan (Vredenburg et al., 2002).

Menetelmä	Sijoitus tärkeysjärjestyksessä						Keskiarvo	Mainintoja
	1	2	3	4	5			
Kenttätutkimukset (sisältää tilannetutkimuksen)	12	6	5	2	1	2,00	28	
Käyttäjävaatimusten analyysi (User requirements analysis)	3	3	0	0	1	2,00	7	
Iteratiivinen suunnittelu (Iterative design)	17	21	9	5	2	2,15	65	
Käytettävyydestaus (Usability evaluation)	12	8	10	7	1	2,39	43	
Tehtäväanalyysi (Task analysis)	6	8	6	7	1	2,61	34	
Fokusryhmät (Focus groups)	5	2	2	1	4	2,79	16	
Muodollinen heuristinen arviointi (Formal heuristic evaluation)	3	2	5	2	2	2,86	15	
Käyttäjähaastattelu (User interview)	2	0	3	4	0	3,00	11	
Prototyyppi ilman käytettävyydestä (Prototype without user testing)	1	3	5	4	1	3,07	15	
Kyselyt (Surveys)	0	2	2	1	1	3,17	9	
Epämuodollinen asiantuntija-arvio (Informal expert review)	4	6	3	10	6	3,28	31	
Korttien lajittelu (Card sorting)	0	1	1	0	1	3,33	5	
Osallistava suunnittelu (Participatory design)	1	0	1	2	1	3,40	7	
Menetelmä, jota ei voida luokitella							64	

Viisi menetelmää arvioitiin usein käytetyiksi, koska vähintään noin kolmasosa vastaajista mainitsi ne. Nämä menetelmät olivat iteratiivinen suunnittelu (iterative design), käytettävyydestaus (usability evaluation), tehtäväanalyysi (task analysis), epämuodollinen asiantuntija-arvio (informal expert review) ja kenttätutkimukset (field studies). Näillä menetelmillä ajateltiin olevan eniten vaikutusta tuotekehitykseen, kuitenkin asiantuntija-arviota lukuun ottamatta. Asiantuntija-arviota käytettiin, koska se oli edullinen menetelmä, mutta sillä ei nähty olevan suurta merkitystä. Vastaavasti vaatimusanalyysiä (requirements analysis) pidettiin paljon resurssieja vaativana menetelmänä, eikä sitä siten käytetä paljoakaan. Tuotekehityksessä toimivat asiantuntijat kuitenkin arvostivat menetelmää ja arvioivat sen olevan tärkeä.

Vastaajia pyydettiin myös arvioimaan yleisesti käyttämiensä menetelmien hyviä ja huonoja puolia. Taulukko 2 näyttää tuotekehityksessä toimivien asiantuntijoiden useimmin esille nostamat hyvät ja huonot puolet menetelmistä. Ominaisuudet on jaoteltu kolmeen pääkategoriaan: menetelmän resurssitarpeisiin, prosessiin ja tuloksiin. Taulukon rivit on järjestetty kolmen kategorian sisällä niin, että useimmin mainitut piirteet ovat ylempinä. Miinusmerkit tarkoittavat taulukossa menetelmien heikkouksia ja plussat hyviä puolia. Numerot merkkien edessä kuvaavat piirteiden mainintojen lukumäärää. Taulukon yksinkertaistamiseksi jokaisen menetelmän yhteydessä esitetään vain kolme eniten mainintoja saanutta hyvää ominaisuutta ja heikkoutta (tai jos monta ominaisuutta sai yhtä paljon kannatusta, mainitaan ne kaikki). Esimerkiksi nopeus on hyvä piirre epämuodollisessa asiantuntija-arviossa, mutta heikkous iteratiivisessa suunnittelussa.

Halpaa hintaa, menetelmän käytön nopeutta ja tulosten pätevyyttä/laatua pidettiin tärkeimpinä menetelmien ominaisuuksina. Tulosten pätevyys/laatu sai kuitenkin ristiriitaisia tuloksia, mikä saattaa johtua sen kattaman aihepiirin laajuudesta (tulosten laatu riippuu havaittujen ongelmien määrän lisäksi myös muista asioista, kuten kerrotaan kohdassa 20.7).

Joitain ominaisuuksia kuten tiedon saatavuutta menetelmän avulla pidettiin hyvinä puolina melkein kaikille menetelmille, mutta noin puolet ominaisuuksista tuotiin esiin pääasiassa menetelmän heikkouksina (kaikki tulokset eivät näy taulukossa 2). Tällaisina heikkouksina tuotiin esiin johdon tuki, prototyypin saatavuus, asiantuntemuksen saatavuus, nopeus,

käyttäjän osallistuminen, suorituksen helppous, dokumentoinnin helppous, oikeiden käyttäjien löytäminen, tulosten pätevyys/laatu, tulosten uskottavuus, sekä tulosten oikea-aikaisuus. Tämä tulos heijastaa huolta menetelmien monimutkaisuudesta ja käytettävyydestä.

**Taulukko 2:** Kolme huonointa ja parasta ominaisuutta käyttäjäkeskeisen suunnittelun menetelmissä (Vredenburg et al., 2002)

	Muodollinen heuristinen arviointi	Epämuodollinen asiantuntija-arvio	Iteratiivinen suunnittelu	Prototyyppi ilman käytettävyydestä	Kenttätutkimukset	käytettävyydestä	käyttäjähastattelu	Fokusryhmät	Tehtäväanalyysi	osallistava suunnittelu	käyttäjävaahtusanalyysi	Kysely	Korttien lajittelu
<b>resurssitarve</b>													
halpa hinta	9+	12+	17+	7+		6-	3-	4+		2-		3+	2+
johdon tuki (management buy-in)				6+									
prototyypin saatavuus			16-	3-					2+				
asiantuntemuksen saatavuus	3-	4-											
tiedon saatavuus						3+	2+					3+	
<b>prosessi</b>													
nopeus	10+	22+	24-	3-	17-	3-	2-	7+	7-			2-	2+
osallistujien yhteistyö					3+		2-			3+			
käyttäjän osallistuminen	7-	10-					4+			4+		2+	
käytännön yhteensopivuus					3+	3-							
suorituksen helppous									4-				
monikäyttöisyys						4-							
dokumentoinnin helppous						3-	2-				2+		
oikeiden käyttäjien löytäminen													
<b>tulokset</b>													
tulosten pätevyys/laatu	6+ 10-	7+ 17-	24+ 20-	6+ 8-	9+ 8-	8+ 3-	2+ 4-	14+ 9-	3+ 8-	5- 4+		3- 2-	
asiayhteyden ymmärtäminen			19+		31+	7+	2+		10+	4+	3+		2+
tulosten uskottavuus					9-			5-				2-	
tulosten vaikutus													
tulosten oikea-aikaisuus													
<b>Yhteensä (+ / -)</b>	25+ 20-	41+ 31-	60+ 60-	19+ 14-	46+ 34-	18+ 22-	10+ 13-	25+ 14-	15+ 19-	11+ 7-	5+ 0-	8+ 7-	6+ 0-

Myös Wixon (2003) on ottanut kantaa siihen, mitä piirteitä tulisi korostaa menetelmien vertailussa. Hän listasi kolme ammattilaisten mielestä pätevää ohjetta menetelmien valinnalle:

- kyky tunnistaa käyttöliittymäongelmia,
- menetelmän käyttäjän tarvitsemat taidot, ja
- käyttöön tarvittavat resurssit.

Wixon toteaa myös, että pelkkä tehokkuuteen tuijottaminen ei riitä. Vaikka löydettäisiin menetelmä, joka löytää 100 % ongelmista yhdellä käyttäjällä, niin menetelmää ei käytetä tuotekehityksessä, jos se ei sovellu organisaation tuotekehitysprosessiin.

Menetelmien valinnassa joudutaan usein tekemään päätöksiä laadun ja hinnan välillä. Jos halutaan käyttää menetelmää, joka tuottaa paremmat tulokset, siitä on myös maksettava suuremmat kustannukset. Vastaavasti voidaan valita kevyempi menetelmä, jolla ei saavuteta yhtä hyviä tuloksia, mutta, jonka käyttäminen maksaa vähemmän.

## 20.4. Tehtyjä arviointimenetelmien vertailuja

Useissa tutkimuksissa verrataan erilaisia arviointimenetelmiä. Tässä esitellään muutama tällainen tutkimus tuloksineen. Tutkimukset on valittu siten, että ne koskevat yleisesti tunnettuja menetelmiä. Vaikka tässä esitelläänkin näiden tutkimusten tuloksia, niitä ei kuitenkaan kannata lukea kirjaimellisesti. Vertailututkimusten tekemisessä tulee ottaa huomioon monia seikkoja, jotta tuloksista saadaan luotettavia ja vertailukelpoisia. Useat tutkimukset sisältävätkin ongelmia, jotka saattavat johtaa tulosten vääristymiseen. Arviointimenetelmien vertailun vaikeutta käsitelläänkin raportissa myöhemmin.

### 20.4.1. Asiantuntija-arvio vs. käytettävyydestaus

Kohta pohjautuu kokonaisuudessaan Doubledayn ja muiden (1997) tekemään tutkimukseen, jossa arviointimenetelmien käytöllä yritetään saada tietoa sovelluksen käytettävyydestä, jotta sovellusta voitaisiin kehittää. Vertailun kohteena ovat siis asiantuntija-arviointi ja käytettävyydestaus. Käytettävyydasiantuntijat suorittivat asiantuntija-arvioinnin Nielsenin heuristisen tarkistuslistan pohjalta ja käytettävyydestestissä käyttäjien suoritukset videoitiin ja tehtäviin kuluneet ajat kirjattiin ylös. Testit suoritettiin kontrolloiduissa olosuhteissa. Tutkittavana sovelluksena oli tiedonhaku-ohjelma.

Asiantuntija-arvion teki viisi yliopiston käytettävyydasiantuntijaa. He eivät kuitenkaan olleet tiedonhaun asiantuntijoita. Asiantuntijoille esiteltiin ja selostettiin sovelluksen toimintaa ennen arviointia. He tekivät sovelluksella kaksi läpikäyntiä käyttäen hyväkseen seitsemää vähitellen vaikeutuvaa tehtävää, jotka tutkimuksen tekijät antoivat heille. Asiantuntijat etenivät omalla tahdillaan ja koko tapahtuma kesti hieman yli tunnin. Aikarajoituksia ei kuitenkaan ollut.

Käytettävyydestettiin osallistui 20 yliopisto-opiskelijaa. Ennen testiä he täyttivät kyselyn, joka koski heidän kokemustaan tietokoneen, Microsoftin tietokannan ja SQL:n käytössä. Testikäyttäjät tekivät joukon tehtäviä, joiden aikana heitä pyydettiin ajattelemaan ääneen. Heiltä kului tehtävien suorittamiseen aikaa maksimissaan noin 30 minuuttia. Testitilanne videoitiin myöhempää analysointia varten. Testin jälkeen testikäyttäjiä pyydettiin vastaamaan kyselyyn, joka koski tyytyväisyyttä joihinkin ohjelmiston piirteisiin. Kysymykset koskivat erilaisia käytettävyyden osa-alueita kuten joustavuus sekä virheiden välttämisen ja korjaamisen helppous. Testikäyttäjät arvioivat näitä ominaisuuksia asteikolla 1–7. Testin jälkeen haluttiin myös testata, kuinka hyvin käyttäjä muisti tiedonhaun työvaiheet. Testikäyttäjiä pyydettiin kuvaamaan vaiheet, jotka tarvitaan, kun käyttäjä lähettää haun tietokantaan. Vaiheiden kuvauksen aikana käyttäjä ei saanut katsoa sovellusta.

Viisi asiantuntijaa löysi sovelluksesta 86 ongelmaa ja 20 testikäyttäjää kohtasi 38 ongelmaa. Tässä on kuitenkin huomattava, että ongelman määrittely ei ollut tarkkaa. Ongelmien rajaaminen oli hankalaa. Osa ongelmista saattoi sisältää useita seurauksia, jotka toinen asiantuntija olisi määritellyt useaksi eri ongelmaksi. Tämä piirre koskee sekä asiantuntija-arvion että käytettävyydestestin kautta löytyneitä ongelmia. Ongelmien päällekkäisyyksien poistaminen todennäköisesti pienentäisi ongelmien määrää. Tuloksista myös huomattiin, että käytettävyydestestissä löydetty 38 ongelmaa eivät kaikki sisältyneet asiantuntija-arvioinnin ongelmiin. Käytettävyydestestissä löydetystä ongelmista 39 % (15/38) ei löytynyt asiantuntija-arvioinnissa. Jos käytettävyydestestiä siis ei olisi tehty, moni ongelma olisi jäänyt huomaamatta.

Kun verrataan menetelmien vaatimia kustannuksia, tässä tutkimuksessa otettiin huomioon vain arviointimenetelmiin kuluneet ajat. Asiantuntija-arviointiin kului kokonaisuudessaan 33,5 tuntia. Tämä koostui 6,25 tunnista asiantuntijoiden arviointia, 6,25 tunnista muistiinpanojen tekemistä sekä 21 tunnista tulosten puhtaaksi kirjoittamista ja tulkitsemista.

Käytettävyydestiin kului kokonaisuudessaan aikaa 125 tuntia. Kahdenkymmenen testi-käyttäjän testeihin kului aikaa yhdeltä tutkijalta 25 tuntia. Kun testiä oli valvomassa kaksi tutkijaa, se tekee yhteensä 50 tuntia testeihin kulunutta aikaa. Tilastollisen analyysin tekemiseen kului vielä 75 tuntia.

Vaikka molempien arviointimenetelmien tarkoituksena on löytää käytettävyyso ongelmia, ne ovat kuitenkin melko erilaisia menetelmiä. Käytettävyydestissä saadaan selville minkälaisissa tilanteissa ja missä tehtävissä käyttäjällä on ongelmia. Käytettävyydestä ei kuitenkaan kerro, mistä nämä ongelmat johtuvat. Asiantuntija-arviolla taas saadaan paremmin selville, miksi jokin asia voi johtaa virheeseen ja mitä asian parantamiseksi voisi tehdä. Tämä kertoo siitä, että käytettävyydesti on hyvä menetelmä löytämään ongelmien seurauksia, kun taas asiantuntija-arvio pyrkii löytämään ongelmien aiheuttajia ja voi jopa auttaa ongelmien ratkaisuisissa. Doubledayn ja muiden (1997) mukaan asiantuntija-arvio onkin parempi menetelmä, koska se tuottaa ongelmien syitä ja ratkaisuja ongelmiin lyhyessä ajassa, kun taas käytettävyydestä tuottaa vähemmän ongelmia ja vie paljon enemmän resursseja kuin asiantuntija-arvio.

#### 20.4.2. Kuuden arviointimenetelmän vertailututkimus

John ja Marks (1997) tekivät tutkimuksen, jossa he vertailivat peräti kuutta eri arviointimenetelmää. Tämä kohta pohjautuu kokonaisuudessaan Johnin ja Marks (1997) tekemään tutkimukseen. Tavoitteena oli tutkia tehokkuuspuun avulla menetelmien muutosvoimaa. Tutkimukseen valittuja menetelmiä voidaan käyttää tuotteen iteratiivisessa suunnittelussa: kognitiivista läpikäyntiä (luku 9, Ranne), GOMSia (luku 12, Heimonen), heuristista arviointia (luku 8, Korvenranta), vaatimusanalyysiä, käyttäjän toimintojen kirjaamista, sekä pelkkää määrittelydokumentin (spesifikaation) lukemista. *Vaatimusanalyysi (claims analysis)* määriteltiin Kelloggin (1989) sekä Carrollin & Rossonin (1991) tutkimusten mukaan. Käyttäjän toimintojen kirjaamisen (*User Action Notation*) on tarkoitus tuottaa yksikäsitteinen määrittely sovelluksesta. Tämän menetelmän pohjana käytettiin tutkimuksia Siochi & Hartson (1989) sekä Hartson ja muut (1990).

Kaikki tutkimuksessa käytetyt menetelmät eivät ole tarkoitettuja puhtaasti käytettävyyden arviointiin ja käytettävyyso ongelmien etsimiseen. Käyttäjän toimintojen kirjaamisella löydetty ongelmat havaittiin lähinnä sivutuotteena menetelmän käytössä. Tämä onkin Johnin ja Marks (1997) mielestä luontevaa, sillä tämän menetelmän tarkoitus ei olekaan tunnistaa käytettävyyso ongelmia, vaan tuottaa yksikäsitteinen määrittely sovelluksesta.

Tutkimus toteutettiin siten, että kuusi yliopisto-opiskelijaa jakoi menetelmät keskenään. Kukin opiskelijoista perehtyi yhteen menetelmään ja arvioi käyttöliittymän tämän menetelmän mukaan. Arviointimenetelmiä käytettiin siis sovelluksen suunnitteluvaiheessa ja arviot tehtiin 35-sivuisen käyttöliittymäkuvauksen pohjalta. Kun ongelmat oli saatu selville, tutkimusryhmän kanssa työskennellyt ohjelmoija kävi ongelmat läpi ja luokitteli ne sen mukaan, mitkä olivat oikeita ongelmia sovelluksessa ja mitkä eivät. Hän teki kunkin menetelmän avulla löytyneiden mielestään todellisten ongelmien pohjalta uuden korjatun version sovelluksesta. Muodostetuille versioille tehtiin lopuksi käytettävyydestit, jotta saatiin selville niihin jääneet ongelmat.

Ohjelmoija oli työskennellyt arvioitavana olevan ohjelmiston kanssa jo melkein 10 vuotta. Vaikkei hän ollut määrittelyn kirjoittaja, hän oli osallistunut työhön ja ohjelmoinut osan sovelluksesta. Hänellä oli myös jonkin verran perustietoja tutkimuksessa käytetyistä menetelmistä, muttei juurikaan käytettävyyteen liittyvää kokemusta.

Menetelmien avulla löytyi kaiken kaikkiaan 323 ongelmaa; vaatimusanalyysillä 24, kognitiivisella läpikäynnillä 42, GOMSilla 44, heuristisella arvioinnilla 127, käyttäjän toimintojen kirjaamisella 17 ja määrittelyn läpikäynnillä 69. Tarkemmin eri menetelmillä havaittujen ongelmien määrät kuvataan taulukossa 3. Koska ryhmä oli tekemässä uusia versioita vanhasta ohjelmasta, oli löydettyjen ongelmien joukossa myös ongelmakuvauksia, jotka eivät koskeneet muokattavaa versiota sovelluksesta. Nämä ongelmakuvaukset karsittiin joukosta. Loput laaditut ongelmakuvaukset (284 kpl) annettiin ohjelmoijalle, joka perehtyi niihin tarkemmin.

Ohjelmoijan mielestä joukossa oli muutama kuvaus, jotka käytännössä sisälsivät useamman ongelman; nämä kuvaukset hän jakoi useaksi eri ongelmaksi. Osa ongelmakuvauksista koski sellaisia osia sovelluksesta, joita ei ollut vielä toteutettu. Tämän johdosta kuvauksista karsittiin myös ne, joiden olemassaoloa ei voitaisi testata käytettävyydestillä.

**Taulukko 3:** Käytettävyyso Ongelmien määrä eri vaiheissa. Muokattu ja suomennettu John ja Marks (1997).

Menetelmä	Kaikki löydetyt ongelmat	Ohjelmoijalle annetut ongelmat	Ohjelmoijan mielestä ongelmia	Kuvaukset, joita ei pidä ongelmina	Ei-ongelmien prosenttimäärä	Käytettävyyso Ongelmien määrä
Vaatimusanalyysi	24	24	25	6	24 %	19
Kognitiivinen läpikäynti	42	42	42	2	5 %	40
GOMS	44	44	45	6	13 %	39
Heuristinen arviointi	127	88	88	24	27 %	64
Käyttäjän toimintojen kirjaaminen	17	17	17	14	82 %	3
Määrittelyn lukeminen	69	69	69	4	6 %	65
<b>Yhteensä</b>	<b>323</b>	<b>284</b>	<b>286</b>	<b>56</b>	<b>20 %</b>	<b>230</b>

Ohjelmoija teki ongelmiin korjaus ehdotukset ja mietti jokaisen ongelman korjaustärkeyden. Ohjelmoijan mielestä 20 % hänelle annetuista ongelmakuvauksista ei ollut sovelluksen oikeita käytettävyyso Ongelmia. Ohjelmoijan tekemän karsinnan jälkeen jäljelle jäi noin 100 ongelmaa. Näistä hän korjasi ne virheet, joiden muuttamista piti tärkeänä (taulukko 4). Hän teki kunkin menetelmän avulla löytyneiden ongelmien pohjalta uuden korjatun version sovelluksesta. Nämä viisi versiota olivat: alkuperäinen sovellus sekä kognitiivisen läpikäynnin, GOMSin, heuristisen analyysin sekä määrittelyn läpikäynnin soveltamisella havaittujen ongelmien pohjalta tehdyt uudet versiot. Käyttäjän toimintojen kirjaamisessa havaittujen käytettävyyso Ongelmien joukossa ei ollut yhtään niin merkittävää ongelmaa, että sen perusteella olisi tehty muutoksia alkuperäiseen ohjelmaan.

**Taulukko 4:** Ohjelmoijan tekemien muutosten määrät (John & Marks, 1997).

Menetelmä	Karsinnan jälkeen ongelmia	Muutokseen johtaneet ongelmat	Prosentti, joka johti muutokseen	Koodissa tehtyjen muutosten määrä
Vaatimusanalyysi	11	0	0	0
Kognitiivinen läpikäynti	15	11	73	5
GOMS	10	3	30	3
Heuristinen arviointi	41	7	17	2
Käyttäjän toimintojen kirjaaminen	0	0	0	0
Määrittelyn lukeminen	23	9	39	7
<b>Yhteensä</b>	<b>100</b>	<b>30</b>	<b>30</b>	

Seuraavaksi oli vuorossa käytettävyyso Ongelmien tekeminen, jotta saatiin tietää, mitkä ongelmat vielä esiintyivät sovelluksessa. Testeihin osallistui 20 käyttäjää, 4 kunkin version testaamiseen. Kaikki käytettävyyso Ongelmat videotettiin. Testikäyttäjille opetettiin ääneenajattelua ja he saivat harjoitella sitä ennen testiä pasianssia pelaamalla. Käyttäjille ei annettu mitään itse sovellusta koskevia ohjeita tai neuvoja.

Tässä vaiheessa ennustetuista 100 ongelmasta karsittiin pois sellaiset ongelmat, jotka eivät koskeneet itse käyttöliittymän käyttöä, vaan ennemminkin tekstin luettavuutta ja vastaavia asioita. Sen lisäksi seurattavia tapahtumia saatiin vielä karsittua 54:ään, koska jotkut ongelmat esiintyivät testikäyttäjien samanlaisena toimintana. Tämän karsinnan jälkeen

- vaatimusanalyysillä ennustettuja ongelmia oli jäljellä 10,
- kognitiivisella läpikäynnillä ennustettuja 8,
- GOMSilla ennustettuja 10,
- heuristisella arvioinnilla ennustettuja 15,

- käyttäjän toimintojen kirjaamisella ennustettuja 0, ja
- määrittelyn lukemisella ennustettuja 16.

Yhteensä ennustettiin, että ongelmia olisi jäljellä siis 54. Kaikki testeistä saadut videot käytiin läpi ja katsottiin esiintyvätkö mainitut ongelmat sovelluksen uusissa versioissa vai ei.

**Taulukko 5:** Käytettävyydestillä saadut tulokset. Koottu John ja Marks (1997).

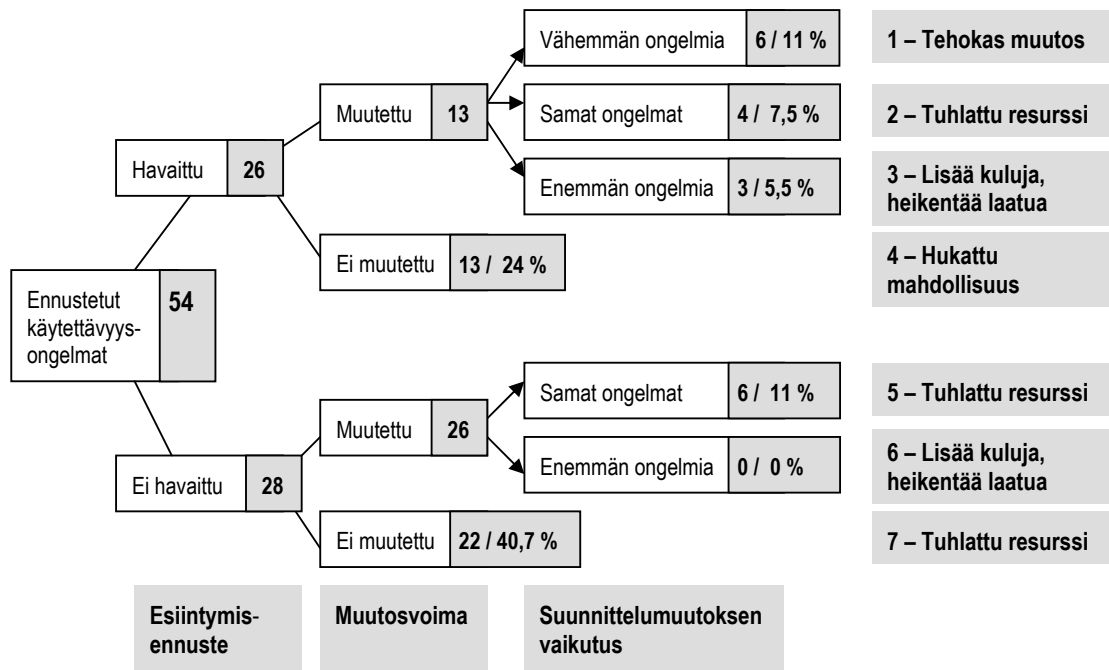
Sovelluksen versio	Toisen karsinnan jälkeen ennustettuja ongelmia	Käytettävyydestillä löydetyt ongelmat	Jäljelle jääneiden ongelmien prosenttimäärät
Kognitiivinen läpikäynti	8	5	63 %
GOMS	10	1	10 %
Heuristinen arviointi	15	2	13 %
Määrittelyn lukeminen	16	1	6 %
Alkuperäinen sovellus	54	26	48 %

Testeissä havaittiin, että alkuperäisessä muuttamattomassa sovelluksessa esiintyi 26 ongelmaa eli 48 % 54 ongelmasta. 48 % ennustetuista ongelmista oli siis todellisia ongelmia, jotka käyttäjät kohtasivat käytettävyydestissä. Kognitiivisen läpikäynnin versiossa oli korjausten jälkeenkin 5 ongelmaa kahdeksasta tällä menetelmällä ennustetusta ongelmasta. GOMS-versiossa esiintyi vielä 1 ongelma kymmenestä, heuristisen analyysin versiossa 2/15 ja määrittelyn lukemisen versiossa 1/16. Parhaiten korjaukset olivat onnistuneet määrittelyn lukemisen ennustamien ongelmien tapauksessa. Vain 6 % menetelmän ennustamista ongelmista havaittiin käytettävyydestissä korjausten jälkeen. Toisaalta tämä voi johtua siitäkin, että ennustetut ongelmat eivät olleet edes todellisia.

Hartson ja muut (2003) ovat tulkinneet tämän tutkimuksen tulokset siten, että eniten ongelmia löytää heuristinen arviointi, tämän jälkeen järjestyksessä tulevat määrittelyn läpikäynti, GOMS, kognitiivinen läpikäynti, vaatimusanalyysi ja viimeisenä käyttäjän toimintojen kirjaaminen. Paras menetelmä muutokseen johtavien ongelmien määrän näkökulmasta (taulukko 4) on kognitiivinen läpikäynti (73 %) sen jälkeen tulevat määrittelyn läpikäynti (39 %), GOMS (30 %), heuristinen arviointi (17 %) sekä viimeisellä tilalla vaatimusanalyysi (0 %) ja käyttäjän toimintojen kirjaaminen (0 %).

Tutkijat totesivatkin tutkimuksen tuloksena, että nämä tutkitut arviointimenetelmät eivät ole niin tehokkaita kuin toivoisi, mutta tilanne ei ole kuitenkaan toivoton. Tässä tutkimuksessa käytettiin tehokkuuspuuta tutkimusongelman asettelussa (kuva 2). Tarkastellaan seuraavaksi taulukon 5 tuloksia tehokkuuspuuhun sijoitettuna. Vain 11 % laadituista ongelmakuvauksista johti toivottuun tulokseen, eli virheiden vähenemiseen sovelluksessa. Suurimmassa osassa ongelmakuvauksista muutosvoima jäi liian pieneksi: 35 ongelmaa päätettiin jättää korjaamatta. Näistä 22 oli ongelmia, joita käyttäjät eivät edes kohdanneet, mutta yhteensä 13 niistä ainakin yksi käyttäjä kohtasi. 24 % ongelmista oli siis hukattuja mahdollisuuksia.

Korjaamatta jätetyistä ongelmakuvauksista 40,7 % oli tuhlatuja resursseja – eli tässä viitataan siihen työhön, jota tehtiin etsittäessä arviointimenetelmällä ongelmia, joita sitten ei päätettykään korjata. Onneksi ohjelmoija ei lähtenyt näitä muuttamaan. Hän olisi vain tuhlanut enemmän resursseja. Yhteensä muissa ei-toivotuissa vaihtoehdoissa tuhlatujen resurssien osuudeksi tulee 18,5 %, jos ei oteta huomioon viimeistä vaihtoehtoa 7. Suunnittelumuutoksen myötä lisää ongelmia aiheutettiin vaihtoehdossa 3 – yhteensä 3 uutta havaittua ongelmaa. Hyvä tulos on kuitenkin se, että resursseja eniten tuhlaava vaihtoehto 6 vältettiin kokonaan.



Kuva 2: Tehokkuuspuu ja tutkimuksen tulokset. (John & Marks, 1997).

### 20.4.3. Tutkimusten ongelmia

Menetelmien vertaaminen on vaikea tehtävä monestakin syystä. Ehkä kaikkein tärkein vertailujen tekemistä häiritsevä tekijä on standardoimattomat menetelmät, joilla käytettävyystutkimusta tehdään. Menetelmistä on olemassa liian monia erilaisia variaatioita. On vaikeaa tehdä vertailuja ja varsinkin lukea niistä tehtyjä raportteja, kun jokaisessa tutkimuksessa on käytetty hieman erilaista versiota. Tällöin vertailuja ei voi vertailla keskenään. Jotta menetelmiä voitaisiin tehokkaasti vertailla, pitäisi ensin menetelmät itsessään saada standardoituja.

Kohdassa 20.4.1 käsitellyssä tutkimuksessa käytettiin käytettävyystestiä toisena menetelmänä. Tässä tutkimuksessa käytettyyystestiin ajateltiin tarvittavan 20 testikäyttäjää, vaikka yleensä käytettyyystestiin riittää pienempikin testikäyttäjämäärä. He myös keräsivät kvantitatiivista tietoa testin aikana ja tekivät testin tuloksista tilastollisen analyysin. Myöskään tämä ei yleensä kuulu käytettyyystestiin. Tämä selkeästi huononsi käytettyyystestin asemaa tässä vertailussa, sillä tutkijat nimenomaan valittivat sitä, kuinka paljon resursseja tämä menetelmä vie. Tosin myöskään asiantuntija-arviota ei ollut tutkimuksessa käytetty niin kuin sitä yleensä käytetään.

Menetelmien käyttäjätkin – käytettyyysarviointin tekevät asiantuntijat – vaikuttavat merkittävästi vertailututkimuksen tuloksiin. Esimerkiksi jos asiantuntija-arvioinnin tekee 5 asiantuntijaa, joista yksikään ei ole sovellusalueen asiantuntija, ovat tulokset varmasti erilaisia kuin jos arvioinnin tekee 5 sovellusalan asiantuntijaa. Arvioijan vaikutusta tuloksiin käsitellään paremmin raportin luvussa 19 (Perälä).

Arvioijan vaikutuksen ongelma sisältyy kohdassa 20.4.2 käsitelyyn tutkimukseen. Kyseisessä tutkimuksessa menetelmiä käyttivät opiskelijat, jotka eivät aikaisemmin olleet tutustuneet niihin. He opettelivat käyttämään menetelmää samalla kun arvioivat sovellusta kyseisen menetelmän avulla. Tulokset poikkeavat varmasti sellaisista tuloksista mitä kokeneemmat tutkijat saisivat menetelmiä käyttämällä.

## 20.5. Tulosten vertailupohja

Kuten edellä nähtiin, menetelmiä verrataan usein toisiinsa niiden avulla löydettyjen ongelmien lukumäärien avulla ja näin voidaan saada tietoa ainoastaan menetelmien keskinäisestä paremmuudesta. Joskus voidaan kuitenkin tarvita tietoa myös jonkin menetelmän absoluuttisesta suorituskyvystä. Standardiongelmajoukko on mahdollisimman täydellinen listaus tutkimuksessa käytettävän sovelluksen käytettävyysongelmista. Se sisältää siis kaikki sovelluksessa olemassa olevat todelliset ongelmat. Standardiongelmajoukko on erittäin tärkeä osa arviointimenetelmien vertailua. Sen löytäminen tai tuottaminen on vaikeaa. Hartsonin ja muiden (2003) mielestä standardiongelmajoukon löytämiseen on neljä erilaista menettelytapaa:

- tunnettujen käytettävyysongelmien kylväminen,
- laboratoriopohjainen käytettävyystudkimus,
- asymptoottinen laboratoriopohjainen käytettävyystudkimus sekä
- vertailtavilla menetelmillä löydettyjen ongelmien yhdistäminen.

### 20.5.1. Käytettävyysongelmien kylväminen

Tunnettujen käytettävyysongelmien kylväminen (seeding with usability problems) perustuu oletukseen, että käytössä on ohjelma, jossa ei ole käytettävyysogelmia. Tähän tutkijat lisäävät haluamansa käytettävyysogelmat. Hartsonin ja muiden (2003) mielestä tämä on yksi harvoista menettelytavoista, joiden avulla tutkijat tietävät kaikki käytettävyysogelmat, jotka esiintyvät ohjelmassa. Kuitenkin tämä pitää paikkansa vain jos oletetaan, ettei alkuperäisessä ohjelmassa ollut ongelmia ennen kylvämistä.

Ongelmien kylvämisessä piilee vaara, että kun tutkija kylvää yhden ongelman tutkittavaan sovellukseen, tästä yhdestä ongelmasta tulee useita ongelmia testikäyttäjälle. Tutkija ei siis voi kuitenkaan olla aivan varma mitä ongelmia aiheutuu kylvetystä ongelmasta. Esimerkiksi linkin nimen muuttaminen toiseksi voi vaikuttaa koko sovelluksen käytettävyyteen ennalta arvaamattomalla tavalla. Käyttäjät eivät tyypillisesti arvioi linkkien nimiä irrallaan toisistaan, vaan osana koko sovellusta. Jos yksi linkki on nimetty toisin, voi käyttäjä joutua ihan uusille teille ja törmätä aivan uusiin ongelmiin yhden ongelman kylvämisen seurauksena.

Osa arviointimenetelmien tutkijoista vierastaa käytettävyysongelmien kylvämistä. Arviointimenetelmien tutkimuksen tulos saattaa riippua liikaa tutkijan taidosta kylvää aidonkaltaisia ongelmia. Vaikka ongelmat lisäävä tutkija olisikin kokenut, niin asetellut ongelmat eivät kuitenkaan korvaa oikeiden ongelmien vaihtelevuutta, yllätyksellisyyttä ja todellisuutta. (Hartson et al., 2003)

### 20.5.2. Laboratoriopohjainen käytettävyystestaus

Standardiongelmajoukon muodostukseen käytetään kaikkein eniten laboratoriopohjaista käytettävyystestausta (laboratory-based usability testing). Testit vievät paljon resursseja ja niillä on myös muita heikkouksia. Esimerkiksi tehtävien valinta vaikuttaa suuresti siihen, mitkä ongelmat nousevat testeissä esille (Koskinen, luku 13). Myös laboratorioympäristö saattaa vaikuttaa testikäyttäjiin. Testeistä saadaan tulokseksi joukko ongelmia, jotka testihenkilöt kohtasivat.

Laboratoriotestienkin havainnoista joudutaan joskus miettimään, ovatko ne todellisia ongelmia, vaikka ne ovat tulleet esille testeissä. Tässä voidaan ottaa avuksi asiantuntija-arvio. Näin voidaan rajata pois osa ongelmista, joiden ei ajatella olevan todellisia, käyttäjän oikeassa käyttökontekstissa kohtamia ongelmia. Tämä parantaa standardiongelmajoukon tarkkuutta ja laatua. (Hartson et al., 2003)

### 20.5.3. Asymptoottinen laboratoriopohjainen käytettävyystudkimus

Käytettävyydesteihin osallistuvien henkilöiden määrä tulee ongelmaksi myös standardiongelmajoukkoa kootessa. Monien testien tekeminen vie paljon resursseja ja tulee kalliiksi. Lopullinen löydettyjen ongelmien määrä on pääteltävissä asymptoottisesti, kun testeissä käyneiden käyttäjien määrä kasvaa. Asymptootilla tarkoitetaan siis suoraa, jota ongelmien määrä jatkuvasti lähestyy, kun käyttäjien määrä kasvaa. Asymptootin tasoa voidaan pitää hyvänä arviona ongelmien kokonaismäärästä. Sillä pyritään siis ennustamaan lopullista ongelmien määrää, kun testikäyttäjiä ei ole tarpeeksi. (Hartson et al., 2003)

### 20.5.4. Yhdistetty käytettävyyso Ongelmien joukko

Yhdistetty käytettävyyso Ongelmien joukko tarkoittaa eri menetelmien havaintojen yhdistämistä. Standardiongelmajoukkoon otetaan kaikki ongelmat, jotka löydetään, kun sovelletaan verrattavia arviointimenetelmiä toisistaan riippumatta. Tämän menettelytavan hyvä puoli on siinä, ettei tarvita erikseen resursseja standardijoukon etsimiseen. Kun tutkimusta tehdään tällä tavalla, ei voida kuitenkaan sanoa mitään arviointimenetelmien pätevyvyydestä, eli siitä kuinka tehokkaasti menetelmä löytää todellisia virheitä. (Hartson et al., 2003)

## 20.6. Arviointimenetelmien tarkka vertailu

Seuraavaksi esitellään tapa, jolla menetelmien vertailuun voitaisiin tuoda standardoituja piirteitä. Tässä keskitytään vain formatiivisten arviointimenetelmien vertailuun. Tutkijat perustelevat valintaa sillä, että suuri osa menetelmistä pyrkii nimenomaan tuotekehitysprosessin aikana löytämään suunnitteluvirheitä, jotta niitä voitaisiin sitten korjata (Hartson et al., 2003).

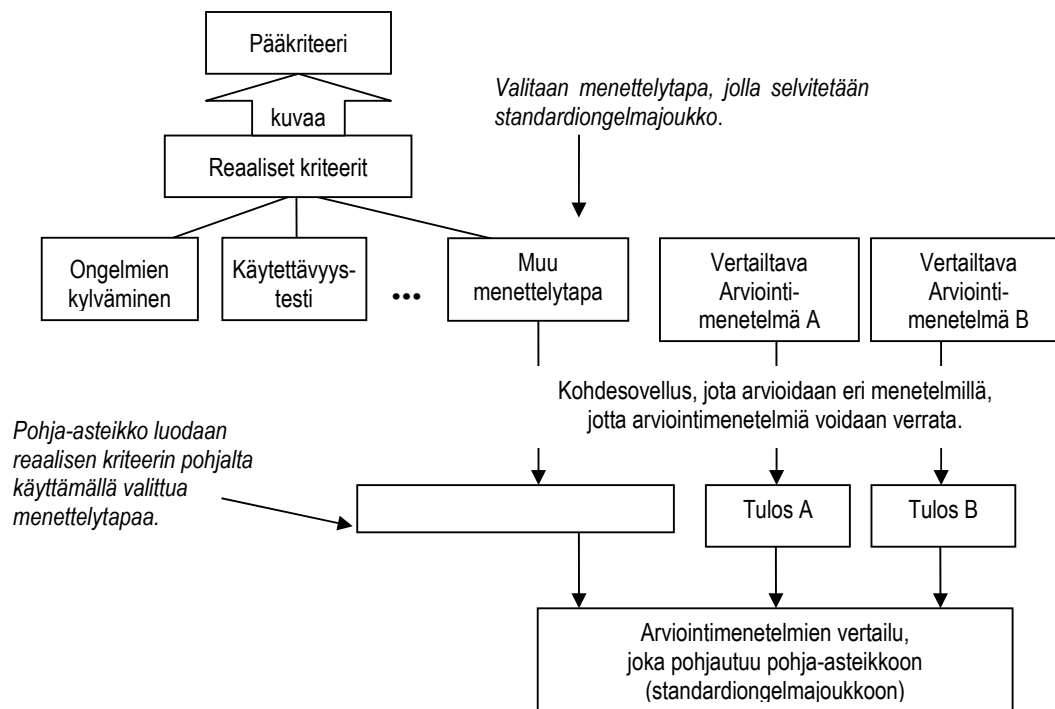
Tarkastellaan ensin arviointimenetelmien vertailua isossa mittakaavassa. Kuvassa 3 esitetään Hartsonin ja muiden (2003) julkaiseman käsittekaavion mukaelma. Tähän versioon käsittekaaviosta on lisätty muutamia reaalisen kriteerin käyttömetodeja. Myös kaavion lukemista on yritetty helpottaa antamalla enemmän tietoa prosessin kulusta.

Ensimmäisenä vaiheena, kun halutaan verrata arviointimenetelmiä, pitää päättää, mikä on *pääkriteeri* (*ultimate criterion*), jonka suhteen menetelmiä halutaan verrata. Pääkriteerin pitäisi vastata toiminnan päämäärää. (Hartson et al., 2003) Pääkriteerinä voisi olla esimerkiksi ”kyky parantaa sovellusta mahdollisimman nopeasti” tai ”kyky saada aikaa mahdollisimman ongelmattomia sovelluksia”.

Koska tällainen pääkriteeri on vielä liian yleisellä tasolla, jotta arviointimenetelmiä voitaisiin oikeasti verrata sen suhteen, pitää kehittää pääkriteeriä kuvaava *reaalinen kriteeri* (*actual criterion*). Reaalisen kriteerin pitää olla niin muotoiltu, että sen avulla arviointimenetelmiä voi verrata. (Hartson et al., 2003) Esimerkkinä voitaisiin etsiä reaalisia kriteereitä edellisen esimerkin pääkriteerille, joka koski kykyä saada aikaa mahdollisimman ongelmaton sovellus. Reaalinen kriteeri voi olla esimerkiksi, että ”arviointimenetelmä löytää kaikki ongelmat, joita sovelluksessa on”. Tämän tasoisen kriteerin avulla voimme jo verrata menetelmiä toisiinsa.

Yleensä arviointimenetelmiä vertaillessa tarvitaan tietoa kaikkien ongelmien määrästä sovelluksessa. Tätä joukkoa kutsutaan standardiongelmajoukoksi. Standardiongelmajoukko toimii siis yleensä pohja-asteikkona arviointimenetelmiä vertailtaessa. (Hartson et al., 2003)

Seuraavaksi pitää selvittää pohja-asteikko, jota vasten arvioidaan verrattavia arviointimenetelmiä. Esimerkissä arviointimenetelmiä on tarkoitus verrata sen mukaan, löytävätkö ne kaikki ongelmat sovelluksesta. Pohja-asteikolla tarkoitetaan tässä esimerkissä niitä kaikkia ongelmia, joita sovelluksessa on. Oletetaan, että sovelluksessa on kaiken kaikkiaan 20 ongelmaa ja vertailtava arviointimenetelmä löytää 5 niistä eli 25 % ongelmista. Tässä tapauksessa 20 ongelmaa on siis pohja-asteikko, jota vasten arviointimenetelmien tuloksia vertaillaan.



**Kuva 3:** Käsitettä käytettävyysmenetelmien vertailulle. Muokattu ja käännetty suomeksi (Hartson et al., 2003).

Pohja-asteikko selvitetään käyttämällä jotakin menettelytapaa. Vaihtoehtoja menettelytapoiksi esiteltiin kohdassa 20.5. Esimerkkinä menettelytavasta voidaan mainita laboratoriopohjainen käytettävyystestaus, jolla esimerkkitapauksessamme yritetään selvittää kaikki sovelluksessa olevat ongelmat.

Jotta vertailu etenee, pitää selvittää reaalisen kriteerin arvo tekemällä käytettävyysarvio jokaisen vertailtavana olevan arviointimenetelmän avulla. Esimerkissä reaalisen kriteerin arvo on löydettyjen ongelmien lukumäärä. Tuloksia, jotka saatiin menetelmien soveltamisesta, verrataan pohja-asteikkoon.

Gray ja Salzman (1998) kiinnittivät huomion useamman kriteerin tarpeellisuuteen. Jos tutkijat vertaavat arviointimenetelmiä vain yhden reaalisen kriteerin perusteella, se voi johtaa pääkriteerin merkittävään puutteellisuuteen. Yksi reaalinen kriteeri ei välttämättä kuvaakaan pääkriteeriä tarpeeksi tarkasti, vaan tarvitaan useampia reaalisia kriteereitä. Tutkijoiden tulisi pyrkiä ottamaan huomioon useampia kriteereitä kuin yksi, mutta kaikkia ei tietenkään voi ottaa huomioon. Riippuu suuresti pääkriteeristä, mitkä reaaliset kriteerit tulisi ottaa huomioon.

## 20.7. Arviointikriteerit

Yleisimmin käytetyt reaaliset kriteerit arviointimenetelmien arvioimiseen ovat:

- *Kattavuus (Thoroughness)*: Menetelmän pitäisi löytää mahdollisimman monta sovelluksen ongelmista.
- *Pätevyys (Validity)*: Menetelmän pitäisi löytää ainoastaan oikeita vastauksia, eli ainoastaan todellisia käytettävyysongelmia.
- *Toistettavuus/Luotettavuus/Käyttövarmuus (Reliability)*: Menetelmän pitäisi toimia samalla tavalla ja antaa samanlaisia tuloksia käyttökerrasta riippumatta.

Hartson ja muut (2003) ovat ottaneet käyttöönsä vielä muutaman muunkin kriteerin:

- *Hyödyllisyys (Effectiveness)*: Tämä kriteeri yhdistää kattavuuden ja pätevyyden niin että havaitaan mahdollisimman paljon käyttäjälle vaikeita ongelmia.
- *Kustannustehokkuus (Cost effectiveness)*: Menetelmän kannattaisi saada mahdollisimman hyvät tulokset mahdollisimman vähillä resursseilla.

Seuraavassa käsitellään näistä kolmea: kattavuutta, pätevyyttä ja hyödyllisyyttä.

### 20.7.1. Kattavuus

Kattavuus tarkoittaa sitä, että menetelmä löytää mahdollisimman suuren osan sovelluksessa esiintyvistä ongelmista. Kattavuus on yksi houkuttelevimmista kriteereistä. Totta kai halutaan, että arviointimenetelmä löytäisi niin monta ongelmaa kuin on mahdollista. Hartsonin ja muiden (2003) mukaan Sears (1997) määritteli kattavuuden arviointimenetelmän löytämien ongelmien osuudeksi todellisista sovelluksessa olevista ongelmista, kuten kaavassa (1).

$$\text{Kattavuus} = \frac{\text{Löydettyjen todellisten ongelmien määrä}}{\text{Olemassa olevien todellisten ongelmien määrä}} \quad (1)$$

Jos esimerkiksi tutkittu arviointimenetelmä löysi 10 ongelmaa 20 todellisesta olemassa olevasta ongelmasta, voidaan kattavuuden sanoa olevan  $10/20=0,5$ . Kattavuuskriteeri voi saada siis arvoja 0 ja 1 väliltä. Ne arviointimenetelmät, joilla kattavuus on pieni, eli kattavuusarvo lähentelee nollaa, eivät löydä läheskään kaikkia käytettävyyso ongelmia, joita tutkimuksen kohteena olevassa sovelluksessa on todettu.

Aina ei kuitenkaan riitä se, että saadaan tietää kuinka suuren osan ongelmista arviointimenetelmä löytää. Voi olla tärkeämpää, että menetelmä löytää varmasti vakavimmat ongelmat ja sen lisäksi mielellään muita ongelmia. Kattavuutta voidaan muokata hieman siten, että painotetaan kattavuutta ongelmien vakavuudella.

$$\text{Painotettu kattavuus} = \frac{\sum \text{Vakavuus(Löydetty todellinen ongelma}_i)}{\sum \text{Vakavuus(Olemassa oleva todellinen ongelma}_i)} \quad (2)$$

Kaavassa (2) Vakavuus(i) tarkoittaa käytettävyyso ngelman i vakavuutta. Painotettu kattavuuden laskutapa antaa enemmän pisteitä sellaisille arviointimenetelmille, jotka löytävät vakavampia virheitä, ja jättää vähemmille pisteille vähemmän tärkeitä ongelmia löytävät menetelmät.

Toisaalta, jos ollaan erittäin tarkkoja vakavien ongelmien löytymisestä, saatetaan tarvita vielä erilainen tapa tarkastella kattavuutta. Joillekin tutkijoille voi olla hyödyllisempää tarkastella kattavuutta eri vakavuusasteilla erikseen, esimerkiksi kaavan (3) avulla.

$$\text{Kattavuus}(v) = \frac{\text{Löydettyjen todellisten ongelmien määrä vakavuustasolla } v}{\text{Olemassa olevien todellisten ongelmien määrä vakavuustasolla } v} \quad (3)$$

Käytännön tuotekehityksessä ollaan kiinnostuneempia vakavimmista ongelmista ja jätetään pienemmät ongelmat vähemmälle huomiolle. Tuotekehityksessä voidaan olla kiinnostuneita myös arviointimenetelmän löytämien ongelmien vakavuuskeskiarvosta.

$$\text{Vakavuuskeskiarvo(arviointimenetelmä)} = \frac{\sum \text{Vakavuus(Löydetty todellinen ongelma}_i)}{\text{Arviointimenetelmällä löydettyjen todellisten ongelmien määrä}} \quad (4)$$

Kaavan (4) lukua voitaisiin verrata toisista arviointimenetelmistä saatuihin vastaaviin lukuihin tai kaikista sovelluksessa olemassa olevista ongelmista laskettuun vakavuuskeskiarvoon..

$$\text{Vakavuuskeskiarvo(kaikki)} = \frac{\sum \text{Vakavuus(Olemassa oleva todellinen ongelma}_i)}{\text{Olemassa olevien todellisten ongelmien määrä}} \quad (5)$$

Myös kaava (5) antaa korkeita arvoja sellaisille arviointimenetelmille, jotka löytävät kaikkein vakavimpia ongelmia, vaikka nämä menetelmät eivät saisikaan hyviä pisteitä kokonais-kattavuudesta.

Pitää kuitenkin huomata, että ongelmilla voi olla erilaisia vakavuusasteita eri ihmisille. Toiset saattavat kokea jonkin ongelman ylitsepäsemättömänä, eivätkä pysty jatkamaan tehtävänsä, kun toisille saman ongelman ohittaminen ja työn jatkaminen voi olla helppo juttu. Ongelmista johtuvien seurauksien vakavuus tälle kyseiselle käyttäjälle riippuu hänen taidoistaan ja taustoistaan. Kokeneempi käyttäjä luultavasti selviää helpommin ongelmien aiheuttamista seurauksista. Jos käytetään menetelmää, jossa testikäyttäjää ei tavoiteta, vakavuuden arviointi perustuu vain arvioijan käsityksiin; tätä tarkastellaan enemmän luvussa 19 (Perälä).

## 20.7.2. Pätevyys

Pätevyys tarkoittaa tässä sitä, kuinka hyvin menetelmä tekee sen, mitä sen on tarkoitus tehdä. Arviointimenetelmien vertailussa sen voisi muotoilla esimerkiksi, kuinka hyvin menetelmä löytää todelliset ongelmat. Hartsonin ja muiden (2003) mukaan Sears (1997) määritteli pätevyuden kaavan (6) avulla.

$$\text{Pätevyys} = \frac{\text{Löydettyjen todellisten ongelmien määrä}}{\text{Kohtien määrä, jotka on määritelty ongelmiksi}} \quad (6)$$

Pätevyys ja kattavuus voidaan laskea samoista tiedoista. Näiden laskemiseksi tarvitaan tieto kaikista todellisista ongelmista, jotka esiintyvät standardiongelmajoukossa. Esimerkiksi: arviointimenetelmä löysi 20 ongelmaa ja standardiongelmajoukossa on 15 todellista, olemassa olevaa ongelmaa. Todellisista ongelmista menetelmällä löydettiin ainoastaan viisi. Menetelmän pätevyys saisi arvon  $5/20=0,25$ . Kattavuus saisi tässä esimerkissä arvon  $5/15=0,33$ . Arviointimenetelmät, jotka saavat pieniä arvoja pätevydestä, löytävät paljon kohtia, jotka eivät ole todellisia ongelmia. Tämä taas tuhlaa tutkijoiden ja kehittäjien aikaa, jos aletaan korjata sellaisia kohtia, jotka eivät oikeasti ole ongelmia tai joihin käyttäjät eivät ikinä törmää.

## 20.7.3. Hyödyllisyys

Hyödyllisyyskriteeri on Hartsonin ja muiden (2003) kehittämä. Heidän mielestään kattavuus tai pätevyys ei yksinään ole tarpeeksi tehokkaita kriteereitä, joilla arviointimenetelmiä voitaisiin vertailla. He yhdistivät nämä kaksi kriteeriä ja muodostivat niistä kriteerin hyödyllisyyden. Yksinkertaisimmillaan kriteeri saadaan muotoon:

$$\text{Hyödyllisyys} = \text{Kattavuus} \times \text{Pätevyys} \quad (7)$$

Hyödyllisyyden arvot vaihtelevat välillä nolasta yhteen, kuten kattavuuden ja pätevyuden arvot. Jos jompikumpi, kattavuus tai pätevyys, on heikko, saa myös hyödyllisyys heikon arvon. Jotta siis saadaan hyvä arvo hyödyllisyydelle, pitää molempien ominaisuuksien olla kunnossa.

Myös tässä kriteerissä voidaan painottaa sen eri osia. Joskus halutaan painottaa hieman enemmän esimerkiksi kattavuutta kuin pätevyttä. Usein tutkijat hyväksyvät sen, että he käyvät läpi mieluummin hieman useampia ongelmia, joita käyttäjät eivät välttämättä edes kohtaa, kunhan kattavuus saa tarpeeksi hyvän arvon.

Määrälliset ongelmien vertailut ovat hankalia ainakin sen vuoksi, että ongelmakuvaukset eivät ole yhteismitallisia, vakavuuden arviointi tapahtuu useimmiten asiantuntijavoimin perustamatta arviota todellisen käyttäjän havainnoinnille, ja standardiongelmajoukkoa ei tunneta.

## 20.8. Lopuksi

Käytettävyytutkimusmenetelmiä voidaan vertailla moneen eri tarkoitukseen. Yksi haluaa selvittää, mikä menetelmä olisi paras niillä resursseilla, mitä kulloinkin on käytettävissä. Toinen pitää tärkeämpänä, että menetelmä on nopea. Kolmas pitää menetelmän tehokkuutta tuotekehitysprosessissa tärkeimpänä. Vertailututkimuksia löytyy monenlaisia. Tämä onkin yksi vertailututkimusten vaikeuksista. Tutkimusten lukeminen on hankalaa, kun tutkijat käyttävät erilaisia kriteerejä, tutkivat eri variaatioita käytettävyytutkimusmenetelmistä ja suorittavat vertailunsa eri lailla. Tutkimusalueelta puuttuvat standardit.

Helpoin tapa vertailla menetelmiä on ajatella menetelmiä terveellä järjellä ja luokitella niitä sen mukaan, mitä kukin menetelmä vaatii ajalta, resursseilta ja sovelluksen tuotantovaiheelta. Jos ei ole väliä, löytääkö menetelmä esimerkiksi 50 % vai 55 % ongelmista, ei välttämättä tarvita suuria vertailututkimuksia. Tällaisilla käytännön toimintatavoilla voidaan päätellä, milloin mitään menetelmää mahdollisesti kannattaisi käyttää.

Myös tuotekehityksessä toimivat ammattilaiset ovat hyviä henkilöitä vertailemaan eri menetelmiä, koska nämä henkilöt ovat todennäköisesti käyttäneet eri menetelmiä melko laajasti. Heitä pyydettiin mainitsemaan eniten käyttämiään menetelmiä. Tuloksena saatiin iteratiivinen suunnittelu, käytettävyydestaus, tehtäväanalyysi, epämuodollinen asiantuntija-arvio ja kenttä-tutkimukset. Näitä menetelmiä käytetään tuotekehityksessä, koska niillä ajatellaan olevan eniten vaikutusta tuotekehitys prosessiin.

Jos käytettävyystudkimuksen menetelmiä halutaan verrata oikein tarkasti, niin että saadaan selville, mikä menetelmästä löytää suurimman osan virheistä tai löytää ne oikeasti käyttäjiä häiritsevät virheet, täytyy suorittaa paljon tarkempia tutkimuksia. Myöskään tällaisilla tutkimuksilla ei ole standardeja, joiden mukaan menetelmiä tulisi vertailla. Hartson ja muut (2003) esittelivät tutkimustavan, jota heidän mielestään tulisi käyttää menetelmävertailussa. He esittelivät joukon kriteerejä, joista tässä luvussa esiteltiin kolme: kattavuus, pätevyys ja hyödyllisyys. Kukin näistä kriteereistä on helppo havaita tärkeäksi tuotekehityksen kannalta, jossa pitää nimenomaan pyrkiä löytämään mahdollisimman suuri osa niistä oikeista käyttäjiä häiritsevistä ongelmista. Näin sovelluksen kanssa ei tehdä turhaa työtä ja resursseja säästyy.

Jotta menetelmien tehokkuutta voidaan arvioida tietyn sovelluksen yhteydessä, pitää vertailusta sovelluksesta selvittää sen standardiongelmajoukko. Millään menettelytavalla ei kuitenkaan saada määritettyä standardiongelmajoukkoa niin hyvin, että tulokset vastaisivat täysin todellisuutta.

Käytettävyystudkimuksen menetelmien vertailu on melko tuore tutkimusalue, eikä yleisesti hyväksyttyjä, luotettavia ja tehokkaita vertailutapoja menetelmäkehityksen tueksi ole vielä kehitetty. Toivottavasti tulevaisuudessa saadaan standardoituja käytettävyystudkimusmenetelmiä, joita voitaisiin tutkia ja vertailla standardoiduilla menetelmillä ja kriteereillä. Toisaalta menetelmien vertailu on inhimillistä toimintaa, joten inhimilliset erot vertailussa ja tutkimuksessa näkyvät aina. Jos vertailua saataisiin yhtenäistettyä, voitaisiin oppia lisää erilaisista käytettävyyssmenetelmistä ja oppia, milloin kutakin niistä tulisi käyttää. Jos opitaan tuntemaan nykyisten menetelmien hyvät puolet ja heikkoudet tarpeeksi hyvin, voidaan tulevaisuudessa ehkä myös osata kehittää yhä parempia käytettävyyden tutkimusmenetelmiä.



**Katri Kosonen**, Fil. yo. Olen opiskellut tietojenkäsittelytieteiden laitoksella vuodesta 1999. Työni laitoksella aloitin vuonna 2002. Toimin tuntopalautteen tutkijana tietojenkäsittelytieteiden laitoksella ja teen pro gradu -työtäni aiheesta. Ennen tietojenkäsittelytieteiden laitokselle siirtymistäni työskentelin opiskelujeni ohessa myös käyttöliittymäsuunnittelijana Solteq Oyj:llä.

**Ohjaaja: Saila Ovaska**

**Opponentit: Leena Parviainen ja Suvi Vuorela**

## Lähteet

Axup, J. (2004) Comparison of usability evaluation methods (UEMs). Available at: [http://www.userdesign.com/usability\\_uem.html](http://www.userdesign.com/usability_uem.html) (12.5.2004)

Carroll, J.M. & Rosson, M.B. (1991) Deliberated evolution:stalking the view matcher in design space. *Human-Computer Interaction* 6, 281–318.

Doubleday, A., Ryan, M., Springett, M. & Sutcliffe, A. (1997) A comparison of usability techniques for evaluating design. *Proc. of Designing Interactive Systems (DIS '97)*, ACM, 101-110.

Gray, W.D. & Salzman, M.C. (1998) Damaged merchandise? A review of experiments that compare usability evaluation methods. *Human-Computer Interaction* 13 (3), 203–261.

Hartson, H.R., Andre, T.S. & Williges, R.C. (2003) Criteria for evaluating usability evaluation methods. *International Journal of Human-Computer Interaction* 15 (1), 145–181.

Hartson, H.R., Siochi, A.C. & Hix, D. (1990) The UAN: a user-oriented representation for direct manipulation interface designs. *ACM Transactions on Information Systems*, 8, 181–203.

John, B.E. & Marks, S.J. (1997) Tracking the effectiveness of usability evaluation methods. *Behaviour & Information Technology* 16 (4/5), 188–202.

Kellogg, W.A. (1989) Qualitative artifact analysis. (Research report RC 15768), IBM T.J. Watson Research Center.

Sears, A. (1997) Heuristic walkthroughs: Finding the problems without the noise. *International Journal of Human-Computer Interaction* 9, 213–234.

Siochi, A. & Hartson, H.R. (1989), Task oriented representation of asynchronous user interfaces. *Proc. of Computer Human interaction (CHI'89)*, ACM 183 - 188.

Vredenburg, K., Mao, J.-Y., Smith, P.W. & Carey, T. (2002) A survey of user-centered design practice. *Proc. of Computer Human interaction (CHI'02)*, ACM 471 - 478.

Wixon, D. (2003) Evaluating usability methods: Why the current literature fails the practitioner. *interactions* 10 (4), July 2003.

## Kuvien lähteet

Kuva 1: Uudelleen piirretty ja käännetty suomeksi. Alkuperäinen kuva löytyy lähteestä: John, B. E. & Marks, S. J. (1997) Tracking the effectiveness of usability evaluation methods. *Behaviour & Information Technology* 16 (4/5), 188–202.

Kuva 2: Uudelleen piirretty ja käännetty suomeksi. Alkuperäinen kuva löytyy lähteestä: John, B. E. & Marks, S. J. (1997) Tracking the effectiveness of usability evaluation methods. *Behaviour & Information Technology* 16 (4/5), 188–202.

Kuva 3: Muokattu ja käännetty suomeksi. Alkuperäinen kuva löytyy lähteestä: Hartson, H. R., Andre, T. S. & Williges, R. C. (2003) Criteria for evaluating usability evaluation methods. *International Journal of Human-Computer Interaction* 15 (1), 145–181.