

Käytettävyydestauksen ja heuristisen läpikäynnin vertailututkimus

Päivi Valtonen

Pro gradu – seminaarin kirjallinen alustus
Helsinki 27.02.2012

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Sisällys

1	Johdanto	1
2	Käytettävyys	1
2.1	Käytettävyyden määritelmä	1
2.2	Käytettävyyden standardit	3
2.3	Hyvän käyttöliittymän ominaisuuksia	4
3	Käytettävyystutkimus	8
3.1	Käytettävyystestaus	8
3.2	Käytettävyystestauksen suorittaminen	12
3.3	Variaatioita käytettävyystestausmenetelmistä	16
3.4	Heuristinen arviointi	17
3.5	Kognitiivinen läpikäynti	20
3.6	Heuristinen läpikäynti	22
3.7	Tekniikoiden vertailua	24
3.8	Validin datan kerääminen	26
4	Tietojenkäsittelytieteen laitoksen verkkosivujen käytettävyystutkimus	27
4.1	Määritelmä ja käytetyt tekniikat	27
4.2	Testitapaukset	27
4.3	Heuristinen läpikäynti	27
4.4	Käytettävyystestaus	27
5	Käytettävyystutkimuksen tulokset ja johtopäätökset	27
6	Yhteenveto	27
	Lähteet	28

1 Johdanto

Tutkielman tarkoituksena on esitellä muutama yleisin käytettävyyden tutkimusmenetelmä, tarkastella näistä kahta menetelmää tarkemmin, sekä käyttää valittua kahta menetelmää pieneen verkkosivujen käytettävyydestutkimukseen.

Verkkosivujen käytettävyyttä tutkitaan kahdella valitulla tutkimustekniikalla gradun soveltavassa osassa. Ensimmäisen käytettävyydestin suorittaa gradun tekijä itse. Tähän käytetään heuristista läpikäyntiä. Kun heuristinen läpikäynti on suoritettu gradun tekijän toimesta, verkkosivuston käytettävyyttä tutkitaan vapaaehtoisten testikäyttäjien voimin käytettävyydestaus -tekniikalla. Kyseisessä soveltavassa osassa tutkitaan Helsingin Yliopiston Tietojenkäsittelytieteen laitoksen verkkosivujen käytettävyyttä. Tarkasteltavaksi osaksi on valittu, filosofian maisterin tutkintoa suorittavien opiskelijoiden pakolliseen Pro gradu -tutkielmaan liittyvät ohjesivut.

Tutkielman soveltavasta osasta saadut tulokset liittyvät Pro gradu -ohjesivujen käytettävyyteen liittyviin mahdollisiin ongelmakohtiin. Samoin tutkimuksen tuloksena voidaan vertailla kahden eri käytetyn tutkimustekniikan tulosten eroavaisuuksia.

2 Käytettävyys

Seuraavissa aliluvuissa käsitellään käytettävyyden määritelmää ja standardeja sekä hyvän käytettävyyden ominaisuuksia.

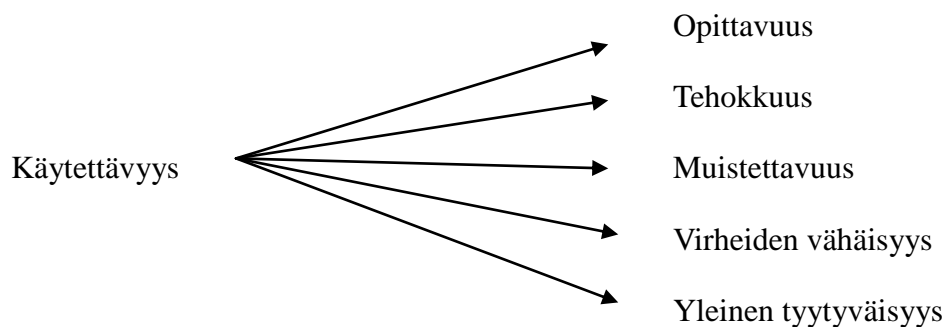
2.1 Käytettävyyden määritelmä

Käytettävyydellä tarkoitetaan ohjelmiston helppokäyttöisyyttä ja hyvää soveltuvuutta käyttöympäristöön [Nielsen93, luku 2]. Ohjelmistotuotannossa on lukuisia eri vaiheita, jotka osaltaan vaikuttavat siihen, kuinka onnistunut projektin lopputulos on. Vaatimusmäärittely, lähdekoodin selkeä kommentointi sekä ohjelmiston testaaminen ovat tärkeitä ohjelmistotuotannon vaiheita. On tärkeää, että lopputuloksena syntyvä ohjelmisto on käytettävyydeltään hyväksyttävällä tasolla.

Ohjelmistoa suunnitellessa täytyy ottaa huomioon ohjelmiston käyttäjät, käyttäjien tavoitteet, käyttöympäristö ja tehokkuuden sekä tyytyväisyyden mittarit [Nielsen93, luku 4]. Tällöin täytyy siis jo tuotantovaiheessa miettiä minkälaiset alkuperäiset tiedot, taidot, koulutustausta, tavat, toiveet ja kykenevydet käyttäjillä on. Täytyy myös huomioida

da, onko tulevilla käyttäjillä joitain fyysisiä tai psyykkisiä esteitä jotka täytyy ottaa huomioon. On tärkeää selvittää miten käyttäjät tulevat ohjelmistoa ja sen eri toiminnallisuuksia käyttämään. Tämän takia ei pidä selvittää vain ohjelmiston toiminnallisia vaatimuksia, vaan myös niiden mahdollinen toisto (kuinka usein) ja kesto (kuinka nopeasti jokin tehtävä on tärkeää suorittaa) [Nielsen93, luku 6]. Ympäristöstä täytyy ottaa huomioon laitteet joilla ohjelmistoa käytetään, kuten myös käytettävät materiaalit ja tuleeko ohjelmisto hyvin sosiaaliseen ympäristöön vai rauhalliseen yksin suoritettavaan ympäristöön.

Jakob Nielsenin mukaan käytettävyys koostuu opittavuudesta, tehokkuudesta, muistettavuudesta, virheiden vähäisyydestä ja yleisestä tyytyväisyydestä [Nielsen93, luku 2].



Kuva 1. Malli käytettävyyden attribuuteista [Nielsen 93, s. 25]

Järjestelmän käyttö tulee olla mahdollisimman helppoa. Opittavuus tarkoittaa sitä, että järjestelmää on helppo oppia käyttämään ilman tarkkoja ohjeita. Paras mahdollinen tilanne olisi se, jos käyttäjä jo ensimmäisellä käyttökerralla osaa ja oppii käyttämään järjestelmää sujuvasti. Jos järjestelmää käytetään usein ja käyttäjä joutuu jokaisella käyttökerralla turvautumaan apuun tai lisäohjeisiin on järjestelmän käytettävyydessä parannettavaa. Järjestelmän tehokkuus tarkoittaa sitä, että järjestelmä suoriutuu sille asetetuista tehtävistä helposti ja nopeasti. Tämä tarkoittaa sitä, että järjestelmän ei ole hidasteena tai vaikeuta käyttäjän tekemisiä vaan on yksinkertainen ja tehokas. Järjestelmän muistettavuus liittyy opittavuuteen. Kun käyttäjä on kerran käyttänyt järjestelmän jotakin toimintoa, tulisi hänelle olla helppoa muistaa seuraavilla käyttökerroilla, miten kyseinen toiminnallisuus toimii. Ensimmäisellä käyttökerralla uuden toiminnallisuuden oppiminen voi toki olla hidasta, mutta jatkossa järjestelmän käytön tulisi olla niin suoraviivais-

ta että toiminnallisuuksien käyttäminen on tehokkaampaa ja käyttäjä suoriutuu niistä itsenäisesti ilman ohjeistusta. Virheiden vähäisyys on tärkeä osa järjestelmän käytettävyyttä. Virheet hankaloittavat järjestelmän toiminnallisuuksien suorittamista ja näin ollen vaikeuttavat järjestelmän käyttöä. Viimeisenä käytettävyyden osa-alueena on yleinen tyytyväisyys. Jos järjestelmän käyttäjä itse tuntee, että järjestelmää on miellyttävä käyttää on järjestelmä tällöin onnistunut olemaan käytettävyydeltään hyvä.

2.2 Käytettävyyden standardit

ISO 9241-11 [ISO11] ja ISO 13407 [ISO07]. ovat kaksi tärkeää käytettävyyteen liittyvää standardia. Ensimmäinen määrittelee käytettävyyden ja jälkeinen ohjeistaa käytettävyyssuunnittelua. Käytettävyyttä standardisoiva ISO 9241-11 selittää, kuinka tunnistaa se tieto, joka on tärkeää ottaa huomioon, kun mitataan käytettävyyttä suorituksen ja tyytyväisyyden osalta. ISO 13407 standardi tarjoaa ohjeistusta käyttäjakeskeiseen suunnitteluun.

ISO standardit ovat kansainvälisesti käytössä olevat yleiset toteutus- ja kirjoitusmenetelmät, joihin kuuluvat muun muassa sellaiset universaalisti tunnetut tunnuksien kuten eri kielten lyhenteet, valuuttojen koodit, ja päivämäärien muoto. ISO standardeissa on otettu kantaa myös tietotekniikkaan ja käytettävyyteen.

Käytettävyyden laatua mitataan järjestelmän käytön avulla, eikä sen avulla mitä ominaisuuksia järjestelmästä löytyy. Käytettävyyttä voidaan mitata esimerkiksi tutkimalla kuinka tehokkaasti käyttäjät pystyvät suoriutumaan tietyistä järjestelmän käyttötehtävistä ja kuinka tyytyväisiä käyttäjät itse ovat järjestelmän käyttämiseen.

ISO 9241-11 selittää kuinka järjestelmän käytettävyyden laatua voidaan mitata käyttäjäsuorituksen ja tyytyväisyyden perusteella; sillä laajuudella että tarkoitetut tavoitteet on saavutettu, resurssit joita käytetään saavuttavat tarkoitetut tavoitteensa ja käyttäjä pitää järjestelmän käyttämistä hyväksyttävänä. ISO 9241-11 korostaa että käytettävyyden laatu riippuu käytön kontekstista. Laadukas käytettävyyden tason saavuttaminen riippuu niistä olosuhteista missä järjestelmää käytetään. Tämä käytön konteksti rakentuu käyttäjistä, tehtävistä, laitteistosta ja fyysisestä sekä sosiaalisesta ympäristöstä jotka voivat kaikki vaikuttaa järjestelmän käytön laatuun erityisesti työympäristössä. ISO 9241-11:sta voi käyttää kehyksenä kun määritellään käytettävyyden laatuvaatimuksia jotka järjestelmän tulee toteuttaa ollakseen hyväksyttävällä tasolla.

ISO 13407 tarjoaa ohjeistusta jotta saavutetaan laadukas käytettävyys. Se soveltaa käyttäjäkeskeisiä suunnitteluaktiviteetteja koko järjestelmän elinkaaren ajan. Se kuvaa käyttäjäkeskeistä suunnittelua monipuolisena aktiviteettina joka yhdistää ihmis-aspektin ja ergonomian tuntemuksen ja tekniikat, tarkoituksenaan korostaa tehokkuutta, parantaa ihmisen työolosuhteita ja olla vastuksena mahdollisten ihmisen terveyttä, turvallisuutta ja suoritusta uhkaaville asioille. On olemassa neljä käyttäjäkeskeistä suunnitteluaktiviteettiä jotka tulee toteuttaa jokaisessa projektin vaiheessa. Suunnittelijan tulee:

- ymmärtää ja määrittää käytön konteksti
- määrittää käyttäjän ja organisaation vaatimukset
- tuottaa suunnitteluratkaisuja
- arvioida suunnittelua vaatimuksia vasten

Käytön konteksti tarkoittaa käyttäjän ja käyttöympäristön ominaisuuksien ymmärtämistä. Oleellisia käyttäjäpiirteitä voivat olla käyttäjän tieto, taito, kokemus, koulutus, fyysiset attribuutit, tavat ja kykenevyydet [Bevan97]. Käyttöympäristöstä on olennaista tietää mitä ohjelmistoja, verkostoja ja laitteistoja on käytössä, kuten myös missä esimerkiksi työtoimisto sijaitsee, minkälainen on toimiston sisustus, sisälämpötila ja minkälaisia määräyksiä ja lakeja tai kulttuurillisia säädöksiä siellä on käytössä. Ohjelmistoja, verkosivuja ja tietojärjestelmiä laaditaan tarpeeseen. Vain se, että järjestelmä tekee mitä pyydetään, ei vielä riitä tekemään järjestelmästä laadukasta. Jos esimerkiksi suunnitellaan tietokoneohjelmistoa kiireiseen työympäristöön kuten sairaalan päivystysosaston henkilökunnalle on kyseisessä ohjelmistossa otettava huomioon eri asioita kuin toimistokäyttöön tarkoitettussa ohjelmistossa.

2.3 Hyvän käyttöliittymän ominaisuuksia

Käytettävyyttä voi mitata esimerkiksi laskemalla virheiden määrää ohjelmiston yksinkertaisen toiminnon suorittamisessa, tai mittaamalla kuinka paljon aikaa jonkin tehtävän suorittamiseen kuluu [Nielsen93]. Yleisesti ottaen, ohjelman käytettävyttä mitattaessa otetaan huomioon, että ohjelma tekee sen mitä sen oletetaan tekevän. Myös käyttäjän tyytyväisyys on tärkeä aspekti käytettävyttä mitattaessa.

Käytettävyyden standardit ovat viitteitä siitä, minkälaisia vaatimuksia käytettävyydeltään hyvällä ohjelmistolla on. Pelkkä standardien läpikäynti ja implementointi tuotettavassa ohjelmistossa ei kuitenkaan korvaa käytettävyyssmittauksen tärkeyttä. Vaikka ohjelmisto olisi tehty niin sanotusti sääntöjen mukaan ei tämä kuitenkaan takaa, että ohjelmisto olisi käytettävyydeltään hyvä. Standardien pohjalta toteutettu ohjelmisto on todennäköisesti toteutukseltaan parempi kuin ohjelmisto jonka teossa ei ole otettu huomioon käytettävyyden standardeja. Kuitenkin lopullinen käytettävyyden taso pystytään havaitsemaan vasta käytettävyyssmittauksen yhteydessä varsinaisilla aidosti järjestelmän kohderyhmää edustavien testikäyttäjien avulla.

Jakob Nielsen on luonut sääntöjä, jotka ottavat kantaa ohjelmiston ja sen sivutuotteiden laatuun [Nielsen93, luku 5]. Jotta voidaan taata ohjelmiston korkea laatu, on otettava huomioon ne keskeiset toiminnallisuudet joita ohjelmistolla on tarkoitus suorittaa. Ohjelmistoa tehdään nimenomaan asiakkaan ja tulevien käyttäjien ehdoilla. Jos tämä vaatimus ei toteudu, voi ohjelmistosta olla lopulta enemmän haittaa kuin hyötyä.

On tärkeää ottaa huomioon minkälaiseen käyttöön ja ympäristöön ohjelmistoa tehdään. On myös tärkeää käyttää, sekä ohjelmiston koodauksessa että dokumentaatiossa, käyttäjän ymmärtämää yksinkertaista ja luonnollista kieltä. Tekninen sanasto voi näyttää ohjelmiston tekijät viisaammassa valossa, mutta ohjelmistoa tehdään nimenomaan käyttäjän ehdoilla.

Käyttöliittymien tulisi olla mahdollisimman yksinkertaisia. Jokainen uusi toiminnallisuus tai tieto, joka lisätään järjestelmän ruudulle, tekee käyttöliittymästä monimutkaisemman ja antaa näin ollen käyttäjälle yhä enemmän vaihtoehtoja tämän toiminnan suhteen [Nielsen93, sivu 115]. Kun käyttöliittymän eri toimintojen määrä kasvaa, näin ollen kasvaa myös käyttäjän tekemien virheaskeleiden määrä. Käyttäjällä voi kulua aikaa oikean, etsimänsä, toiminnallisuuden löytämiseen ruudulta, tai hän voi klikata näkymästä toiseen etsiessään tarvitsemaansa toiminnallisuutta. Tämä tarkoittaa ylimääräistä työtä. Suoraviivainen ja yksinkertainen käyttöliittymä vähentää turhien askeleiden tekemistä ja näin ollen nopeuttaa tavoitellun toiminnallisuuden löytämistä ja suorittamista.

Käyttöliittymien tulisi olla toteutettu niin, että käyttöliittymän ja toiminnallisuuksien käyttäminen noudattaisi samanlaista suorituspolkua kuin miten käyttäjä ajattelee. Tämä tarkoittaa sitä, että käyttöliittymä seuraa käyttäjän luontaista ajatus- ja toimimispolkua. Tämä näyttäytyy käyttäjälle niin, että jokainen toiminnallisuuden osa on luontaisesti

edelliseen tai seuraavaan toiminnallisuuteen yhteydessä ja käyttäjän mielestä toiminnallisuuden käyttäminen on loogista. Näin ollen käyttöliittymä ja käyttäjän oma ajatusmalli toiminnallisuuden eri vaiheista tulisi olla mahdollisimman yhteneväisiä [Nielsen 93, luku 5.1]. Tähän tavoitteeseen päästään kun käyttöliittymä on minimaalinen ja mahdollisimman yksiselitteinen.

Käyttäjän muistikuormaa pyritään minimoimaan niin, että ohjelmistoa pystyy käyttämään ilman käyttöoppaan jatkuvaa selaamista. Käyttöliittymän tulee olla selkeä ja looginen, eli kauttaaltaan yhdenmukainen, jotta ohjelmistoa on helppo käyttää. Käyttäjän on hyvä myös saada palautetta eri toimintojen suorittamisesta, ja hänelle on myös tärkeää osoittaa selkeä poistumistapa eri tiloista ja toiminnoista. Ohjelmiston kannalta on myös hyvä, jos useaan kertaan toistettavat tehtävät voi tarpeen vaatiessa toteuttaa nopeammin ja yksinkertaisemmin esimerkiksi oikopolkuja käyttäen. Erilaisista virhetilanteista on tärkeää antaa selkeät virheilmoitukset ja turhia virhetilanteita on syytä välttää. Mahdolliset virhetilanteet on hyvä huomata jo ohjelmiston tuotantovaiheessa esimerkiksi hyvän koodauksen ja testauksen avulla [Nielsen93, luku 5.3].

Käyttäjän huomio kohdistuu käyttöliittymässä muusta sisällöstä selvästi erottuviin kokonaisuuksiin [Nielsen93, luku 5.3]. Tieto, esimerkiksi teksti, joka sijaitsee länsimaissa aivan sivun vasemmassa ylälaidassa saa yleensä enemmän huomiota kuin sen alla olevat seuraavat osiot. Tekstin korostaminen on tehokas keino saada tärkeät asiat huomatuksi. Tekstin korostaminen on myös eräänlainen sisällön jaottelutapa, jolloin sisältöön saadaan ryhtiä ja sisällön eri osiot tulevat käyttäjälle selvemmiksi. Sisällön jaottelu kokonaisuuksiksi ja näiden kokonaisuuksien välinen sijoittaminen on osa käyttöliittymäsuunnittelua. Tekstiä voi korostaa esimerkiksi erilaisilla väreillä. Kuitenkin, kuten käyttöliittymän toiminnallisuus-suunnittelussa, myös tässä on hyvä harjoittaa minimalistista periaatetta. Värejä ei kannata käyttää liikaa jotta värien käyttämisen tarkoitus ei käänny itseään vastaan. Nielsenin mukaan 5-7 väriä on tarpeeksi. Käyttöliittymän ei kuitenkaan tule olla riippuvainen väreistä. Sivuston on oltava ymmärrettävä myös ilman värierottelua, jotta myös värisokeat käyttäjät otetaan huomioon. Värien käytön ongelmallisuudesta huolimatta värien käyttö on tehokasta esimerkiksi eri kategorioiden ja tekstisisällön korostamiseen. Värejä ei kuitenkaan pidä käyttää normaalin tekstisisällön yhteydessä.

Käyttöliittymän tulisi olla käytettävissä käyttäjän ymmärtämällä luonnollisella kielellä. Suomessa tämä tarkoittaa usein joko suomen- tai ruotsinkielistä käyttöliittymää. Ongelmana on kuitenkin se, että tietojärjestelmiä usein laaditaan englanninkielellä ja näin

ollen käyttöliittymiin ja niiden sisältöihin saattaa joskus päästä myös englanninkielisiä termejä. Täysin englanninkielisillä sivustoilla asiat voivat olla vielä hankalammin, sillä englannin eri sanoilla voi olla eri merkitys riippuen siitä onko käyttäjä, tai sivuston sisällön laatija, Yhdysvalloista, Australiasta vai Iso-Britanniasta. Suomessa tällaista monitulkintaista sanakulttuuria on vähemmän, mutta se tulee ottaa huomioon myös suomenkielisissä käyttöliittymissä. Kielen lisäksi myös lauserakenne on tärkeä. Käyttäjän näkemät ohjeet ja tiedot tulee olla kohdistettu käyttäjälle - käyttäjän näkökulmasta.

Johdonmukaisuus ja yhtenäisyys ovat käyttäjälle tärkeitä. Jos sivuston rakenne ja toiminnallisuudet eivät ole johdonmukaisia käyttäjä voi hämmentyä jonkin normaalista poikkeavan sivustorakenteen tullessa vastaan. Ensinnäkin sivuston perusrakenne tulisi pysyä samana kaikilla sivuston sisältösivuilla. Tämä tarkoittaa esimerkiksi sitä että sivun navigaatiolinkit pysyvät samassa paikassa ja samanlaisina vaikka käyttäjä klikkaisikin eri alisivuille. Jos navigaatiolinkkien paikka sivulla vaihtelee, on käyttäjän huomattavasti hitaampaa kulkea sivustopolkuja pitkin.

Palaute ja erityisesti virhetilanteissa käyttäjälle esitetty palaute on tärkeää. Virheviestien tulisi olla tarkasti muotoiltuja ja kertoa käyttäjälle mistä virhe johtuu.

WWW-sivuilla jotka sisältävät paljon tärkeää tekstiä on hyvä käyttää oikopolkuja tai viitteitä. Nämä oikopoluiksi kutsutut linkit on hyvä esittää heti sivun alussa ikään kuin sivun sisällysluettelona. Jokaista linkkiä klikkaamalla pääsee linkin osoittamaan tiettyyn kohtaan sivulla. Tämä on hyvä esimerkiksi sellaisessa tilanteessa jossa käyttäjän täytyy useita kertoja tarkastella sivulla sijaitsevaa yksityiskohtaista tietoa, esimerkiksi ohjeita jonkin toiminnallisuuden aikaansaamiseksi.

Käyttäjiltä saa tärkeää tietoa ihmisten ja tietokoneiden välisestä vuorovaikutuksesta kuten siitä miten laitetta, tai www-sivustoa, todella käytetään, riippumatta siitä miten se on suunniteltu käytettäväksi.

Käytettävyyden arvioinnin voi jakaa opittavuuteen ja käytön tehokkuuteen [Nielsen 93, luku 5]. Käytön tehokkuus pystytään määrittelemään laskemalla keskimääräinen aika joka kuluu ennalta määriteltujen tehtävien suorittamiseen. Tyypillisiä käytettävyyden mittaamiseen käytettäviä tekniikoita ovat esimerkiksi:

-aika, kuinka kauan käyttäjällä kestää tietyn tehtävän suorittaminen

- tehtävien määrä jotka ehditään suorittaa tietyn aikarajan sisällä
- onnistuneiden ja epäonnistuneiden(virheellisten) suoritusten suhde

3 Käytettävyystudkimus

Seuraavissa aliluvuissa käsitellään käytettävyystudkimukseen käytettäviä tutkimusmenetelmiä, jotka ovat käytettävyydestaus, heuristinen arviointi, Searsin heuristinen läpikäynti ja kognitiivinen läpikäynti. Näiden arviointitekniikoiden vertailun lisäksi tarkastellaan arviointitehtävien laatimiseen, tutkimustilanteen järjestämiseen ja validien tulosten keräämiseen kehitettyjä menetelmiä.

Tarkastelumethodit ovat tarkoitettu käytettäväksi ohjelmistokehityksen alkuvaiheessa ja vaativat usein vähemmän resursseja kuin käyttäjättestaus. Käytettävyyden tarkistukset voidaan suorittaa järjestelmän prototyypin, kuvakaappausten tai suunnitteludiagrammien avulla. Järjestelmän ei siis tarvitse olla valmis ja lopullisessa muodossaan.

3.1 Käytettävyydestaus

Käytettävyydestauksessa todellista kohdekäyttäjää tarkastellaan kun hän suorittaa todellista järjestelmän käyttötapausta kontrolloidussa ympäristössä [Fang08]. Käyttäjälle annetaan testitilanteen alussa testitehtävät joiden perusteella hänen tulee käyttää järjestelmän toimintoja. Kun käyttäjälle on annettu testitehtävät ja hän on lukenut ne kertaalleen läpi, hänelle on annettava aikaa kysyä kysymyksiä testin tarkkailijalta. Näin minimoidaan testin väärinymmärrysten riski. Testin ensimmäinen testitehtävä tulisi olla tarpeeksi helppo jotta käyttäjä pystyy sen yleensä suorittamaan ilman ongelmia. On tärkeää myös että testin viimeinen tehtävä on sellainen että käyttäjä tuntee saaneensa tehtävän päätökseen. Testitehtävät voi antaa käyttäjälle yksitellen, jotta jos he eivät ehdi suorittaa kaikkia tehtäviä annetun aikarajan sisällä, he eivät kuitenkaan tunne itseään kyvyttömiksi. Testitehtävien perussääntö on, että niiden pitäisi edustaa järjestelmän todellista käyttöä mahdollisimman kattavasti. Siis tehtävien tulee olla sellaisia joita niin sanotusti oikeassa elämässä järjestelmän normaaleille käyttäjille tulee vastaan. Tehtävien tulee olla tarpeeksi pieniä jotta ne voidaan suorittaa aikarajoitusten sisällä, mutta ei niin pieniä että ne olisivat yhdentekeviä. Testitehtävän tulisi määritellä tarkasti, minkälaiseen tuloksen käyttäjän tulisi tuottaa.

Riippumatta siitä minkälainen testimetodi valitaan, jonkun on toimittava testin ohjaajana ja järjestäjänä. On hyvä jos testin ohjaajalla on aiempaa kokemusta käytetystä testausmetodista. Nielsenin tutkimusten mukaan kokeneiden testien ohjaajien käyttäminen johtaa parempiin tuloksiin käytettävyytutkimuksessa. Tämä ei kuitenkaan tarkoita sitä että kokemattomia testiohjaajia ei voisi käyttää. On parempi löytää edes muutama käytettävyysoongelma kuin jättää testi esimerkiksi kokonaan tekemättä kokeneen testijohtajan puuttumisen vuoksi. Kokemattoman testijohtajan tausta vaikuttaa siihen minkälainen johtaja hän voi olla. Tietojenkäsittelytieteilijöiden on mahdollista oppia käyttäjätestin metodeja ja soveltaa niitä hyvin tuloksin. Testausmetodin tuntemisen lisäksi testin johtajalla tulee olla hyvä tuntemus testattavasta järjestelmästä ja sen käyttöliittymästä. Testikäyttäjien tekemisten analysointi onnistuu näin helpommin. Testin johtajalla ei kuitenkaan tarvitse olla kattavaa tietoa siitä kuinka järjestelmä on toteutettu., vaikkakin tämä voi olla kätevää jossain testitapauksissa, esimerkiksi jos järjestelmällä on tapana sammua.

Henkilö, joka on pätevin ohjaamaan käytettävyydestä, on luultavimmin se kenellä on organisaatiossa useita rooleja. Ohjelmistofirmassa tämä henkilö voi olla esimerkiksi käytettävyyssinööri, QA spesialisti tai käyttöliittymäsuunnittelija [Kantner94].

Käytettävyydestä ei tulisi tehdä ilman, että testimetodia olisi ensin testattu muutama testikäyttäjää käyttäen. Usein yksi tai kaksi testikäyttäjää on riittävästi. Koekäyttäjät voidaan valita sen perusteella ketkä henkilöt ovat helposti tavoitettavissa testin tarkkailijan näkökulmasta. Koekäyttäjien ei näin ollen tarvitse edustaa varsinaista kohdekäyttäjryhmää. Kokeellisessa vaiheessa usein huomataan että jonkin tehtävän testaukseen tarkoitetut ohjeet ovat puutteelliset tai käyttäjät tulkitsevat niitä väärin. Samoin esimerkiksi kyselykaavakkeita joudutaan usein muokkaamaan koekierroksen jälkeen. Usein voi myös löytyä väärin laskelmoitu suhde testiin varatun ajan ja testin suorittamiseen käytetyn todellisen ajan välillä. Usein tehtävät ovat vaikeampia kuin on luultu ja näin ollen koetulosten perusteella testejä tulee muokata yksinkertaisempaan muotoon. Varsinaisen käytettävyydestin testikäyttäjien valinnassa tärkeintä on, että he ovat mahdollisimman lähellä testattavan järjestelmän todellista kohdekäyttäjryhmää. Käyttäjryhmä on helppo todeta esimerkiksi silloin kun verkkosivu tai ohjelmisto tuotetaan vain yrityksen sisäiseen käyttöön. Julkista verkkopalvelua tehdessä kohdekäyttäjryhmä on laajempi käsite ja palvelua saattaa eksyä käyttämään myös kohdekäyttäjryhmästä poikkeava yksilö. Testikäyttäjien tulisi edustaa kohderyhmää mahdollisimman laajalti.

Tämä tarkoittaa sitä, että testikäyttäjiksi ei tule valita esimerkiksi vain sellaisia henkilöitä jotka ovat taito- ja tietotasoltaan parhaita.

On tärkeää että käyttöliittymä testataan ainakin aloittelijatason käyttäjillä. Myös kokeneempien käyttäjien käyttäminen testauksessa on toivottavaa. Tyypillisesti nämä kaksi eri käyttäjäryhmää testaavat järjestelmän eri asioita. Joskus käyttäjiä pitää kouluttaa järjestelmän käytön suhteen jotta käytettävyydestä voitaisiin suorittaa ilman suurempia ongelmia. Tämä on tärkeää esimerkiksi silloin kun käyttäjälle jo tutusta järjestelmästä tehdään uudempi ja huomattavasti edellisestä poikkeava versio. Jos käyttäjällä on mielessään ennestään tuttu järjestelmän käyttömalli, hänellä kestää enemmän aikaa tulla tutuksi uuden järjestelmän toimintamallien kanssa. Tämä ei liity nykyisen järjestelmän käytettävyyteen.

Testin pitäjän täytyy miettiä valmiiksi karsintakysymyksiä, joita mahdollisilta testikäyttäjiltä kysytään ennen heidän valitsemistaan mukaan testiin. Näin varmistetaan että käyttäjät ovat sopivia testaamaan järjestelmää.

Käytettävyydestin luotettavuus voi olla kyseenalainen sen takia, että testikäyttäjät voivat olla ominaisuuksiltaan erilaisia. Jotkut testikäyttäjät suoriutuvat tehtävistään nopeammin kuin toiset. Tällöin esimerkiksi testin suorittamisen nopeudesta ei voi suoraan päätellä tuloksia sillä on mahdollista että eri käyttäjät toimivat tietokoneella eri nopeuksilla. Päätöksiä käytettävyyden tuloksista täytyy näin ollen tehdä hieman epäluotettavan datan voimin [Nielsen 93, luku 6].

Validiteetti tarkoittaa sitä mittaako käytettävyydesti todellakin jotain tuotteen käytettävyydelle relevanttia asiaa, kun otetaan huomioon tuotteen oikea käyttö, oikeassa ympäristössä eli poissa laboratorio-oloista. Luotettavuutta voi testata tilastollisilla testeillä, mutta validiteetin testaukseen tarvitaan metodologista ymmärrystä käytetystä testausmenetelmästä sekä hieman perusjärkeä. Tyypillinen validiteettiongelma sisältää väärin käyttäjien käyttämistä tai väärin tehtävien antamista testikäyttäjille tai ajan / sosiaalisten ympäristöiden rajoittamisen puutetta. Esimerkiksi johtotason käyttöön tarkoitettua järjestelmää voitaisiin testata alaa opiskelevilla käyttäjillä jolloin tulokset eivät ole valideja, tai ainakaan samoja kuin jos testikäyttäjinä olisi ollut johtajia. Toisaalta alaa opiskelevilla käyttäjillä on parempi testata järjestelmää kuin esimerkiksi jonkin toisen alan opiskelijoilla [Nielsen93, luku 6].

Ennen testin aloittamista tulee kirjoittaa testisuunnitelma. Testisuunnitelman tulisi ottaa

kantaa muun muassa seuraaviin kohtiin [Nielsen93, luku 6.1]:

- Testin tarkoitus (mitä halutaan saavuttaa)
- Missä ja milloin testi suoritetaan
- Missä tilassa järjestelmän tulisi olla testin aloitusvaiheessa
- Kuka toimii kokeen pitäjänä
- Keitä testikäyttäjien tulisi olla ja kuinka heidät tavoitetaan
- Kuinka monta testikäyttäjää tarvitaan
- Mitä kriteerejä käytetään sen päättämiseen että käyttäjät ovat suorittaneet kunkin tehtävän loppuun oikeaoppisesti
- Mitä käyttäjäapuja (ohjekirja tms.) käyttäjälle annetaan saataville
- Kuinka laajalti kokeen pitäjä saa auttaa käyttäjiä testin aikana
- Mitä dataa kerätään ja kuinka se analysoidaan kun se on kerätty
- Millä kriteereillä käyttöliittymä julistetaan onnistuneeksi/menestyneeksi

Testin tarkoitus voi olla esimerkiksi aiempien havaittujen käytettävyysongelmien jälkeän uudistetun ohjelmiston käytettävyyden tutkiminen. Testikäyttäjien valitseminen ja määrä riippuu siitä, minkälaisesta järjestelmästä on kyse. Esimerkiksi suurelle yritykselle tehtävä ohjelmisto, joka laaditaan monen eri käyttäjäryhmän kuten kehittäjien, konsulttien ja projektipäälliköiden käyttöön, tarvitsee testikäyttäjikseen henkilöitä kustakin käyttäjäryhmästä. Kustakin testitehtävästä on käytävä ilmi mikä sen tarkoitus on, eli milloin se on suoritettu loppuun. Lopputulokseen päätyminen tarkoittaa onnistunutta testitehtävää. On myös mietittävä tarvitseeko testikäyttäjä testitilanteessa järjestelmän käyttämiseen tarkoitettuja ohjekirjoja tai muita apuvälineitä. Jos järjestelmä on yksinkertainen ja ennalta tuttu ei tällaiseen tulisi olla tarvetta. Testitehtäviä laadittaessa on kunkin tehtävän lopputuloksen lisäksi mietittävä myös sitä, mitä oleellista lisätietoa tehtävän onnistunut tai epäonnistunut suorittaminen antaa järjestelmästä. Kerätty data tulee olla sellaista, että sen perusteella voidaan havaita järjestelmän ongelmakohtia ja löytää myös mahdollisia ratkaisuja ongelmiin.

Testitulokset tulisi pitää luottamuksellisena ja raportit kirjoittaa siten että yksittäisiä testikäyttäjää ei voi tunnistaa niistä. Käyttäjiin voi viitata numeerisesti (esimerkiksi käyttäjä1, käyttäjä2), mutta ei nimin tai edes nimikirjaimin. Testin suorittamisen aikana

testin tarkkailijan ei tulisi puuttua testikäyttäjän eteen tuleviin ongelmiin liian hätäisesti vaan tarkkailla miten käyttäjä selviytyy näistä tilanteista - sillä tämä antaa tärkeää tietoa käytettävyydestä. Tarkkailijan ei pidä antaa testikäyttäjän kuitenkaan taistella ongelmien parissa liian kauan niin että käyttäjä turhautuu tai tuntee olonsa epämiellyttäväksi.

3.2 Käytettävyydestestauksen suorittaminen

Käytettävyydestestauksessa on tyypillisesti neljä eri vaihetta: valmistautuminen, esittely, testi, selvitys. Valmistautumisvaiheessa testin ohjaajan tulisi varmistua siitä että testaukseen käytettävä huone on valmis ja tietokone on siinä valmiustilassa kuin sen tulee olla ensimmäisen testitapauksen aloittamisen yhteydessä. Esimerkiksi ruudunsäästäjät ja muut päivitysilmoitukset tulisi olla poissa näkyvistä, jotta ne eivät keskeyttäisi testin kulkua. Esittelyvaiheessa testin johtaja toivottaa testikäyttäjän tervetulleeksi ja antaa lyhyen selityksen testin tarkoituksesta. Tämän jälkeen johtaja selittää testin menetelmätavan. Tärkeimpiä ovat:

- Testin tarkoituksena on arvioida järjestelmää, ei käyttäjää
- Testin johtajan tulisi ilmoittaa (jos näin on) että hänellä ei ole henkilökohtaista yhteyttä järjestelmää kohtaan joten testikäyttäjä voi arvioida sitä vapaasti
- Testituloksia käytetään järjestelmän parantamiseen, joten todellinen tuleva versio on mahdollisesti erilainen kuin testikäyttöön tarkoitettu versio
- Toteamus että testiin osallistuminen on vapaaehtoista ja käyttäjä voi keskeyttää milloin vain
- Vakuutus siitä että testin tulokset pysyvät luottamuksellisina eikä tuloksia näytetä kellekään niin että tietyn testikäyttäjän voisi niistä tunnistaa
- Selitys siitä että käyttäjä saa mielellään kysyä kysymyksiä sillä on tärkeää tietää mitkä osat järjestelmästä ovat epäselviä
- Kutsu testikäyttäjälle kysyä lisäkysymyksiä nyt, ennen testin alkua

Testitilanne on testikäyttäjälle jännittävä tilanne. Tämän takia yllä olevat kohdat on syytä esittää käyttäjälle ennen testin aloittamista. Tällöin tilanne tulee käyttäjälle tutum-

maksi ja mahdollinen suorituspaine voi vähentyä. Testin tarkoituksena on tutkia järjestelmän käytettävyyttä. Järjestelmän käytettävyys ei koskaan ole vain yhden henkilön mielipide. Jos esimerkiksi sama järjestelmä käydään läpi kolmen eri testihenkilön avulla ja näistä yhdellä on vaikeuksia käyttää järjestelmää ja hän törmää järjestelmässä koko ajan tilanteisiin joissa joutuu pyytämään testin johtajan apua, ei tämä vielä tarkoita että järjestelmä olisi huono. Tämä ei kuitenkaan myöskään tarkoita että testikäyttäjä tekisi jotain väärin. Jos esimerkiksi kaikki kolme testikäyttäjää tuntuvat jäävän jumiin samaan vaiheeseen testitehtävien suorittamisessa voi olla että kyseessä on tyypillinen käytettävyysongelma. Tämä on ongelma joka pitää huomioida tuloksissa. Jos käyttäjä jatkuvasti pysähtyy erilaisiin järjestelmän käytön ongelmatilanteisiin voi testitilanne aiheuttaa turhautumista. Tässä tapauksessa käyttäjä on altis tekemään yhä enemmän virheitä. Tässä tilanteessa ohjaajan tulee rauhoittaa tilanne ja auttaa käyttäjää pääsemään eteenpäin ja saamaan onnistumisen tunteita. Tarkoituksena on ylläpitää mahdollisimman neutraali tilanne jossa järjestelmän suoriutuminen käyttäjän sille asettamista vaatimuksista on pääosassa – ei käyttäjän henkilökohtainen suoriutuminen testitehtävistä.

Testin suorittamisen aikana testin johtajan tulisi pidättäytyä vuorovaikuttamisesta käyttäjän kanssa ja tulisi ehdottomasti olla ilmaisematta omia henkilökohtaisia mielipiteitään tai osoittaa meneekö käyttäjällä hyvin vai huonosti. Vain jos käyttäjä on selkeästi jumissa ja tulee onnettomaksi, voi johtaja auttaa häntä.

Testin jälkeen testikäyttäjältä selvitetään (engl. debriefing) yleisiä mielipiteitä järjestelmään liittyen ja kysytään onko heillä parannusehdotuksia järjestelmää koskien. Kun testikäyttäjä on lähtenyt, testin johtajan tulisi mahdollisimman lyhyen ajan kuluttua tarkistaa että kaikki käyttäjän testitulokset on merkattu käyttäjän testinumerolla (esimerkiksi käyttäjä3). Johtajan tulisi myös kirjoittaa lyhyt raportti testikokemuksesta mahdollisimman pian kun tapahtumat ovat vielä kirkkaana johtajan mielessä. Täydellinen raportti kirjoitetaan myöhemmin näiden mukaan.

Suorituksen mittausta käytetään käytettävyyden elinkaareissa arvioimaan onko käytettävyysoikeudet saavutettu. Tämä suoritetaan testikäyttäjien voimin, jotka suorittavat ennalta määritellyn joukon testitehtäviä, samalla kun tästä kootaan aika ja virhedataa. Suuri sudenkuoppa tällaisessa mittauksessa on se että mahdollisesti mitataan jotakin joka liittyy heikosti itse asiaan jota yritetään arvioida.

Ääneen ajattelu on osa käytettävyydestä. Sanomalla ääneen ajatuksiaan testikäyttäjä

antaa mahdollisuuden ymmärtää kuinka he näkevät tietokonejärjestelmän ja tämä taas tekee helpoksi tunnistaa käyttäjän suurimmat väärinymmärryskohdat. Suurin haitta metodin käytössä on se että se ei itsessään sovi kovin hyvin yhteen suoritusmittaustekniikoiden kanssa [Nielsen93, luku 6.8]. Toisaalta sen vahvuus on siinä, että sen avulla voidaan kerätä suuresti laadullista eli kvalitatiivista dataa hyvin pieneltä käyttäjämäärältä. Myös monet käyttäjäkommentit ovat niin tarkkoja ja kuvailevia että niiden käyttäminen raporteissa tekevät testiraportin luettavammaksi ja muistettavammaksi. Toisaalta ääneen ajattelu voi myös antaa vääriä luuloja käytettävyysongelmien syistä jos käyttäjän omille teorioille annetaan liikaa painoarvoa, kuten mielipiteille siitä mikä aiheutti ongelman ja mikä auttaisi sen ratkaisemiseen [Nielsen93, luku 6.8]. Ääneen ajattelu on myös hyvin vaikeaa joillekin käyttäjille ja jatkuva ääneen ajattelu testitilanteessa voi olla hankalaa. Tämä luonnottomuus voi testitilanteessa hankaluuden lisäksi vaikuttaa myös tuloksiin. Puhuminen voi hidastaa käyttäjän järjestelmän käyttämistä ja voi näin vaikuttaa tehokkuuteen. Toisaalta tämä voi myös nopeuttaa käyttäjän toimintoja ja rentouttaa häntä sillä tilanteesta syntyy ns. vuorovaikutus testin tarkkailijan kanssa Jos testikäyttäjä on hiljaa liian pitkän aikaa, testin tarkkailijan tulee kysyä/muistuttaa häntä ajattelemaan ääneen. Tarkkailijan tulee kuitenkin pidättäytyä vastaamasta käyttäjän ääneen lausumiin ihme-tyksiin/kysymyksiin ja antaa käyttäjän ensin yrittää ratkaista ongelmat itse.

Käytettävyysslaboratoriot ovat käytettävyyستهsteissä käytettyjä huoneita/tiloja jotka on erityisesti varustettu tätä tarkoitusta varten. Ne ovat hyödyllisiä, mutta eivät missään nimessä pakollisia, käytettävyyستهstejä tehdessä. On mahdollista muokata normaali toimisto hetkellisesti käytettävyysslaboratorioksi, mutta toisaalta on mahdollista suorittaa käytettävyyستهstausta vähemmällä varustuksellakin kuten pelkällä muistiolla. Käytettävyysslaboratorioissa on usein äänieristetty yksisuuntainen peili joka erottaa tarkkailuhuoneen testihuoneesta jotta mahdolliset testin tarkkailijat ja asiantuntijat voivat keskustella testin suorituksesta häiritsemättä testin kulkua. Testikäyttäjät toki tietävät mitä varten peili on, joten heille on hyvä näyttää peilin takaa löytyvä tarkkailuhuone ja siellä istuvat tarkkailijat ennen testin alkua jotta tilanteesta saadaan tutumpi ja vähemmän stressaava. Käytettävyysslaboratorio on usein varustettu myös usealla videokameralla joita säädetään kaukosäätimen avulla tarkkailuhuoneesta.

Videointi on kätevää ja melkein pakollista tilanteissa joissa käytettävyyttä halutaan mitata minuutin tarkkuudella. Kuitenkin käytännön käytettävyyستهtutkimuksessa videoinnille ei ole tarvetta sillä tarkoitus on löytää suurimmat käytettävyyستهngelmat, ei suinkaan

joka ikistä.

Testikäyttäjiä käyttäessä täytyy ottaa huomioon se, että käyttäjät ovat ihmisiä. Testit tulee toteuttaa kunnioittaen testikäyttäjien tunteita ja hyvinvointia. Testiin osallistuminen voi olla stressaava tilanne käyttäjille. Käyttäjillä on suuret suorituspainet vaikka heille kerrotaan, että tarkoituksena on testata järjestelmää eikä käyttäjän suoriutumista. Käyttäjä voi myös tuntea itsensä hitaaksi tai tyhmäksi jos he hän ei suoriudu testitehtävistä tietyn ajan kuluessa tai kysymättä lisätietoa ja apuja testin ohjaajalta. Myös tieto siitä, että käyttäjää seurataan ja hänen tekemisensä tallennetaan voi tuntua käyttäjästä epämukavalta. Testin johtajalla on velvollisuus saada käyttäjä tuntemaan itsensä rennoksi ja mukavaksi testin aikana ja sen jälkeen. Testin johtajan tulee ennen testin alkua vakuuttaa käyttäjälle että testataan järjestelmää, ei käyttäjää. Käyttäjän tulee saada tietää, että testin suorittamisesta tai käyttäjän omakohtaisesta suorituksesta ei paljasteta julkisesti mitään tietoja. Testin tulee tapahtua rennossa ilmapiirissä ja testin johtajan tulisi varata aikaa pienelle juttelulle ennen testin alkua jotta testikäyttäjä tuntisi olonsa rentoutuneeksi. Ensimmäisen testitehtävän tulisi olla "alkulämmittelyä" eli niin helppo että sen on melkein taattu onnistuvan. Testin johtajan tulisi ennen testiä varmistua siitä että testaukseen käytettävät tilat, tietokone ja tietokoneen ohjelmisto ovat valmiita ennen testikäyttäjän saapumista. Lisäksi tulee olla kopiot testitehtävistä ja mahdollisista kyselylomakkeista. Testisessio tulisi suorittaa ilman häiriöitä: huoneen oveen voi laittaa kyltin tätä varten ja puhelimet tulisi laittaa äänettömälle.

Käytettävyyystutkimuksessa tärkeintä on, että kerätty tieto on validia. Testikäyttäjien tulee olla oikeanlaisia. Testikäyttäjien tulee todella olla osa järjestelmän kohdekäyttäjär ryhmää. Jos osallistujat eivät tätä ole, tulokset eivät heijasta järjestelmän todellisia käytettävyyso ongelmia. Esimerkiksi jos tutkitaan ohjelmistoa joka on tehty sairaalohenkilökuntaa varten ja järjestelmää tutkitaan testikäyttäjillä tai ympäristössä joka ei vastaa todellista sairaalaan kiireistä ilmapiiriä voi tärkeitä käytettävyyso ongelmia jäädä huomaamatta. Tämän voi huomata esimerkiksi silloin jos potilaan tiedot tulee kiireellisesti tarkistaa ohjelmiston tietokannasta. Ohjelmisto ja potilaan tiedot on epäilemättä suojattu käyttäjätunnusten ja salasanojen avulla. Rauhallisessa ympäristössä tehty käytettävyyso testaus ei tuo esille sitä käytettävyyso ngelmaa joka piilee siinä, että kiireisessä tilanteessa moneen eri järjestelmän osa-alueeseen kirjautuminen vie huomattavasti ylimääräistä aikaa. Tällöin ohjelmiston todellisen käyttäjät saattavat käyttää hyvin yksikertaista, samaa, salasanaa kaikkiin järjestelmän osiin tai jopa jättää kirjautumatta ulos järjestelmäs-

tä kun eivät sitä tarvitse. Tämä taas aiheuttaa riskejä potilaiden tietoturvas-
sa.

Myöhään havaittujen ongelmien välttämiseksi testausta varten tulee suunnitella tehtäviä jotka ovat järjestelmälle ominaisia ja luonnollisia. Tämä tarkoittaa sitä, että testitehtävien tulisi keskittyä järjestelmän niihin toiminnallisiin joita käytetään usein. Nämä ovat prioriteetiltaan tärkeämpiä tutkittavia ominaisuuksia kuin toiminnallisuudet joita harvoin käytetään. Vain todenmukaiset käyttäjätehtävät auttavat tunnistamaan mahdolliset järjestelmän käytössä ilmentyvät ongelmat. Itse testitilanteet tulee järjestää niin yhäpitäviksi kullekin testikäyttäjälle kuin mahdollista.

Testin suunnittelu vie suurimman osan testiajasta. Itse käytettävyydestä voi kestää esimerkiksi tunnin, jonka aikana testaa-
ja käy läpi monta eri testitehtävää. Näiden testitehtävien laatimiseen ja käytettävyydestä käytännön jär-
jestelyihin on kuitenkin tätä ennen käytetty huomattava määrä aikaa. On syytä valmistua testitilanteeseen hyvin, jotta itse testaus sujuu näin ollen mutkattomasti ja tuloksena saadaan oikeanlaista tietoa järjestelmän käytettävyydestä. Myös tulosten analysointi jälkeenpäin vie paljon aikaa. Testin suunnittelussa pitäisi ottaa huomioon ainakin suurimmat kysymykset mihin testillä pyritään vastaamaan, testikäyttäjien lukumäärä ja heidän vaaditut ominaisuutensa, resurssit joita tarvitaan testin suorittamiseen sekä testauksen aikataulu.

3.3 *Variaatioita käytettävyydestä testausmenetelmistä*

Ääneen ajattelun variaatio on nimeltään rakentava vuorovaikutus (engl. constructive interaction) [Nielsen93, luku 6.8]. Siinä kaksi testikäyttäjää käyttää järjestelmää yhdessä. Metodien paras puoli on se, että testitilanne on paljon luonnollisempi kuin normaali yhden henkilön ääneen ajattelutesti, sillä ihmiset ovat tottuneet puhumaan kun he yrittävät ratkaista ongelmaa yhdessä. Tällöin käyttäjät myös tekevät enemmän kommentteja. Metodien huono puoli on se, että käyttäjillä voi olla eri oppimisen ja tietokoneen käyttämisen strategioita. Tällöin testitilanne voi vaihdella laidasta laitaan eri järjestelmän käyttötapojen välillä. Joskus myös huomataan että kaksi testikäyttäjää eivät pysty yhteistyöhön. Tämä metodi on kuitenkin hyödyllinen esimerkiksi lapsille tarkoitettujen järjestelmien testaamiseen sillä testitilanne on lapsille helpompi kuin normaali ääneen ajattelumetodi.

Retrospektiivinen testaus (engl. retrospective testing) tarkoittaa sitä, että käyttäjän suorittamasta testitilanteesta on olemassa videonauhoite ja suorituksen jälkeen käyttäjää

pyydetään katsomaan video ja suoritustilanne läpi [Nielsen93, luku 6.8]. Tällöin voidaan kerätä lisätietoa itse testitilanteen ulkopuolella. Jälkeenpäin, videota katsoessa, tehdyt kommentit voivat olla avarampia ja raottavat testikäyttäjän mielipiteitä laajemmin kuin testitilanteen kiireessä tehdyt kommentit. Huono puoli tässä on se että jokainen testi kestää näin ollen tuplasti kauemmin sillä ensin pidetään itse testitilanne ja jälkeenpäin testitilanteen videolta katselutesti.

Valmennusmetodi (engl. coaching method) on jokseenkin erilainen kuin muut käytettävyydestaustametodit. Tässä testikäyttäjän ja testin tarkkailijan välillä on erityinen vuorovaikutussuhde [Nielsen93, luku 6.8]. Muissa metodeissa tarkkailija yrittää välttää testiin puuttumista, mutta valmennusmetodissa tarkkailija itse asiassa ohjaa käyttäjää oikeaa suuntaan järjestelmän käytössä. Valmennustilanteessa käyttäjä saa kysyä mitä vain järjestelmään liittyviä kysymyksiä asiantuntija-valmentajalta (tarkkailijalta) joka vastaa kysymyksiin hänen parhaan tietämyksensä mukaisesti. Valmennusmetodi on luonnollisempi kuin ääneen ajattelumetodi. Metodin avulla esimerkiksi noviisikäyttäjistä saadaan helposti asiantuntijakäyttäjiä, ja tämä on hyödyllistä siinä tapauksessa kun järjestelmän asiantuntija testikäyttäjiä ei ole saatavilla testitilannetta varten.

3.4 Heuristinen arviointi

Heuristisessa arvioinnissa pieni ryhmä arvioijia tutkivat käyttöliittymää ja arvioivat miten se noudattaa käytettävyyden periaatteita eli käytettävyyden heuristiikkoja [Nielsen92]. Jotta saataisiin mahdollisimman monipuolinen kuva käyttöliittymän käytettävyysongelmista, on hyvä että käytettävyyttä arvioi pieni ryhmä eikä vain yksi ihminen. Yksikin arvioija löytää jo monia käytettävyysogelmia, mutta ihmiset ovat erilaisia ja useampi arvioija mahdollistaa mahdollisimman kattavan heuristisen arvioinnin eli mahdollisimman monen eri käytettävyysongelman havaitsemisen. Suositeltava tarkastajien määrä on 3-5 henkilöä [Nielsen93].

Heuristisessa arvioinnissa jokainen arvioija tarkastaa käyttöliittymän yksin. Tämä on tärkeää sillä arvioijat voisivat vaikuttaa toistensa löytöihin ja mielipiteisiin. Arvioinnin tulokset voidaan saada esimerkiksi kirjallisena raporttina jolloin arvioija kirjoittaa mielipiteensä ylös. Toinen vaihtoehto on että paikalla on tarkkailija joka kirjaa ylös tarkastajan ääneen lausumat kommentit ja mielipiteet testin aikana [Nielsen93, luku 5.11]. Tällaisessa tilanteessa tarkkailijalla on vastuu myös tulkita ja kirjata ylös arvioijan käytöstä testin aikana havaitakseen miten käyttäytyminen ja reagointi liittyvät käyttöliitty-

mässä esiin tuleviin ongelmiin.

Tyypillisesti yksi heuristinen arviointisessio yhdelle arvioijalle kestää 1-2 tuntia. Arviointi-session aikana arvioija käy käyttöliittymän läpi monta kertaa ja tarkastaa useat dialogielementit ja vertaa niitä listaan tunnettuja käytettävyyssperiaatteita. Nämä periaatteet eli heuristiikat ovat yleisiä sääntöjä jotka näyttävät kuvaavan yleisiä käyttöliittymän ominaisuuksia. Näiden heuristiikkojen lisäksi käyttäjällä on toki lupa arvioida sivuston käytettävyyttä myös omasta mielestään käytettävyyteen liittyvien sääntöjen kautta jotka hänelle tulevat mieleen. Periaatteessa arvioija päättää itse miten hän haluaa suorittaa arvioinnin. On kuitenkin suositeltavaa että arvioija ensin käy läpi koko arvioitavan käyttöliittymän yleisellä tasolla, ja toisella läpikäyntikerralla kiinnittää enemmän huomiota yksityiskohtiin, tietäen näin miten eri yksityiskohdat liittyvät käyttöliittymän kokonaiskuvaan [Nielsen93, luku 5.11].

Heuristisen arvioinnin tuloksena syntyy lista käyttöliittymän käytettävyysoongelmista, jotka perustuvat niihin käytettävyyssperiaatteisiin joita käyttöliittymän suunnittelu rikkoi kussakin käyttötapauksessa. Heuristinen arviointi ei kuitenkaan tarjoa systemaattista tapaa luoda korjauksia näihin ongelmiin tai mahdollisuutta arvioida ongelmien perusteella päivitetyn käyttöliittymän käytettävyyttä. Aiempien havaintojen perusteella parannetun käyttöliittymän käytettävyyttä ei voida varmistaa peilaamalla aiemmin havaittuja ongelmia nyt tehtyihin korjauksiin. Käytettävyyden varmistamiseksi on suoritettava uusi heuristinen arviointi. Tällöin voidaan todeta että aiemmin havaitut käytettävyysongelmat ovat poistuneet, mutta myös että tilalle on tullut mahdollisia uusia ongelmia. Heuristisen arvioinnin tulokset tarjoavat hyvän lähtökohdan tai suuntaviivan joilla päivitetystä käyttöliittymästä saadaan kitkettä nyt löydetty ongelmat pois.

Lista Nielsenin heuristiikoista:

1. Järjestelmän tilan näkyvyys.
2. Järjestelmän ja todellisen maailman yhteneväisyys.
3. Käyttäjän kontrollointi ja vapaus
4. Yhdenpitävyys ja standardit
5. Joustavuus ja käytön tehokkuus
6. Esteettinen ja minimalistinen suunnittelu
7. Virheiden esto
8. Tunnistaminen ennen mieleen palauttamista

9. Apu ja dokumentaatio

10. Käyttäjän auttaminen tunnistamaan, diagnosoimaan ja selviytymään virheistä

Nielsenin heuristiikat antavat hyvän suuntaviivat sille, mitä asioita järjestelmässä on hyvä tarkastella. Käyttäjän tulee tietää missä tilassa tai kohdassa he ovat järjestelmässä (1. Järjestelmän tilan näkyvyys). Esimerkiksi verkkosivun navigaation tulee olla selkeä ja käyttäjällä tulisi olla selkeä kuvan verkkosivun rakenteesta ja siitä missä kohtaa verkkosivua hän kullakin hetkellä on. Alisivuille mentäessä käyttäjän voi olla vaikea hahmottaa minne hänen tulisi seuraavaksi mennä. Verkkosivun tuleekin noudattaa niin sanotusti todellisen maailman rakennetta. Jos käyttäjä klikkaa linkkiä jonka tulee selityksen mukaan viedä käyttäjä sivulle A. Tulee käyttäjän todella linkkiä klikatessa päätyä sivulle A.

Dringusin ja Cohenin kategorialista Internetin kurssiverkkosivuja varten kehitetyistä heuristiikoista perustuu soveltuvien osin Jakob Nielsenin heuristiikkoihin [Dringus05]:

1. Näkyvyys
2. Toiminnallisuus
3. Estetiikka
4. Palaute ja apu
5. Virheiden esto
6. Muistettavuus
7. Kurssien hallittavuus
8. Interaktiivisuus
9. Joustavuus
10. Yhdenmukaisuus
11. Tehokkuus
12. Toiston vähentäminen
13. Saavutettavuus

Muun muassa näkyvyys, toiminnallisuus, estetiikka, apu, muistettavuus ja yhdenmukaisuus perustuvat Nielsenin heuristiikkalistaan. Lista on kuitenkin jaoteltu hieman eri tavalla soveltumaan paremmin nimenomaan toistoa vaativan verkkosivuston, esimerkiksi tapauksessa verkkokurssien hallinnoinnin, arvioimiseen. Tämän takia mukana on erikseen muun muassa toiston vähentäminen. Verkkosivu tai -palvelu jota käytetään toistu-

vasti samaan tehtävään kannattaa olla toteutettu niin, että vältetään liialta toistolta. Esimerkiksi sivusto, jossa ylläpidetään tietoa ja jonne voi lisätä ja josta voi poistaa tietoa toimii parhaiten silloin kun jatkuvasti poiston tai lisäyksen yhteydessä ei kysytä ”oletko varma että haluat lisätä/poistaa tiedoston?”. Tämän sijaan käyttäjä voisi esimerkiksi itse määrittellä haluaako nähdä varmistusvaiheen jatkossa vai ei.

Yksi mahdollisuus laajentaa heuristisen arvioinnin metodia, edistääkseen jatkosuunnittelua ehdotusten avulla, on suorittaa aivoriihi johon osallistuisivat arvioijat, tarkkailijat sekä sivuston suunnittelijat [Nielsen93]. Keskustelu kohdistuisi pääasiassa suurimpien käytettävyysohjelmien ratkaisemiseen uudessa käyttöliittymässä. Aivoriihi on myös hyvä tapa antaa positiivista palautetta käyttöliittymän suunnittelusta.

Nielsen vertaili heuristiseen arviointiin käytettäviä arvioijia ja näiden aiempaa kokemusta [Nielsen92]. Hän jakoi arvioijat kolmeen ryhmään: aloittelijoihin, joilla on aiempaa kokemusta tietokoneista, mutta ei kokemusta käytettävyydestä, yhden osa-alueen asiantuntijoihin (engl. single experts), jotka ovat käytettävyyden asiantuntijoita, mutta ei tarkempaa kokemusta arvioitavan käyttöliittymän alueelta, sekä kahden osa-alueen asiantuntijoihin (engl. double experts), jotka ovat käytettävyyden asiantuntijoita ja joilla on asiantuntemusta myös arvioitavasta käyttöliittymästä. Tutkimusten mukaan aloittelija-arvioijien tulokset olivat suhteellisen heikot, löytäen keskimäärin 22% käyttöliittymän kaikista käytettävyysohjelmista. Yhden osa-alueen asiantuntijat löysivät 41% ja kahden alueen asiantuntijat 60% kaikista käytettävyysohjelmista. Tulosten perusteella eri käyttäjäryhmät saavat eri tuloksia perustuen siihen minkälainen heidän ennako-asetelmänsä on arvioitavaa käyttöliittymää kohtaan. Käyttäjä jolla ei ole aiempaa käytettävyysohjelmista voi suorittaa heuristisen arvioinnin, mutta on suositeltavampaa että käytetään arvioijaa kenellä on ennestään jo jonkinmoinen käytettävyyden ja arvioitavan järjestelmän tuntemus.

3.5 Kognitiivinen läpikäynti

Kognitiivinen läpikäynti on metodi jonka avulla selvitetään kuinka helposti järjestelmien käyttämisen voi oppia. Metodi soveltuu erityisesti sellaisten järjestelmien arviointiin, joiden tulisi olla käyttäjän ymmärrettävissä jo ensimmäisellä käyttökerralla, kuten julkinen verkkosivu, informaatiopiste kauppakeskuksessa, tai esimerkiksi julkisen kirjaston teoshakupalvelu.

Kognitiivisessa läpikäynnissä arvioija käy sivuston läpi ennalta määriteltyjen tehtävien avulla, jolloin hän voi löytää ongelmia prosessissa ja epäjohtonmukaisuuksia työnkullussa [Warton94]. Ideana on tunnistaa järjestelmän vuorovaikutusta vaativissa osissa ne kohdat missä käyttäjät todennäköisesti kohtaavat ongelmia.

Kognitiivinen läpikäynti on metodi jota on yritetty myös kehittää henkilöille jotka eivät ole ihmisen ja tietokoneen välisen vuorovaikutuksen asiantuntijoita [Aluker05].

Lewis ja Wharton, jotka ovat kehittäneet kognitiivisen läpikäynnin, suosittelevat että valitut testitehtävät ovat tärkeitä järjestelmälle ja realistisia suhteessa siihen minkälaisia tehtäviä todelliset järjestelmän käyttäjät tulevat järjestelmällä suorittamaan [Wharton94]. Kognitiivisessa läpikäynnissä testitehtävän suorittamiseen vaadittu oikea toiminnallisuuden sarja ei välttämättä ole kaikista optimaalisin kun otetaan huomioon hiiren klikkauksen määrät tai tehtävän suoritusnopeus. Oikea toimintojen järjestys on se mikä edustaa suunnittelijan näkemystä siitä miten jokin tietty toiminto suoritetaan. Kognitiivisen läpikäyntisession aikana arvioija suorittaa toiminnallisuuden ja päätelee kuinka menestyneesti tietty käyttäjä voisi suorittaa jokaisen toiminnallisuuden. Arvioija luo "onnistuneen reitin" jokaisessa testitehtävän suoritusvaiheessa. Epäonnistunut reitti edustaa eroavaisuutta suunnittelija aikomuksissa käyttöliittymän käytön suhteen ja käyttäjän aikomuksissa.

Kognitiivisessa läpikäynnissä [Wang08]

-arvioija asettaa tavoitteen joka järjestelmän pitää toteuttaa

-arvioija etsii käyttöjärjestelmästä mahdollisia toiminnallisuuden (esimerkiksi. menu, linkkinappulat)

-arvioija valitsee toiminnallisuuden joka vaikuttaa todennäköisimmältä vaihtoehdolta jotta hän voisi edetä kohti toiminnallisuuden onnistunutta suoritusta

-arvioija suorittaa valitun toiminnallisuuden ja arvioi järjestelmän palautetta saadakseen todistuksen siitä että edistystä on tapahtunut kohti kyseistä toiminnallisuuden suorittamista

Kognitiivisessa läpikäynnissä on lukuisia esivaatimuksia joita tulee miettiä:

- yleinen kuvaus siitä keitä testikäyttäjät ovat ja mitä relevanttia tietoa heillä on
- kuvaus yhdestä tai useammasta tehtävästä joita järjestelmällä tulee suorittaa
- lista oikeista toiminnoista joita tarvitaan jotta tehtävät voi järjestelmällä suorittaa

Kun kognitiivista läpikäyntiä käytetään, täytyy miettiä jokaisen testitehtävän yksittäisen toimenpiteen kohdalla:

- yrittääkö käyttäjä saada aikaan oikeanlaisen lopputuloksen?
- huomaako käyttäjä että oikeanlainen toiminto on saavutettavissa?
- yhdistääkö käyttäjä oikeanlaisen toiminnon ja halutun lopputuloksen?
- jos oikeanlainen toiminto suoritetaan, huomaako käyttäjä että edistystä on tapahtunut kohti tehtävän lopullista ratkaisua?

Jos yllä oleviin neljään kysymykseen saadaan kaikkiin positiivinen vastaus, tätä kutsutaan "onnistuneeksi suoritukseksi" kyseiselle toiminnallisuudelle. Jos yhteenkin kysymykseen saadaan negatiivinen vastaus, tätä kutsutaan "epäonnistuneeksi suoritukseksi".

3.6 Heuristinen läpikäynti

Heuristinen läpikäynti (engl. heuristic walktrough) on tekniikka joka yhdistää heuristiset arvioinnin ja kognitiivisen läpikäynnin [Sears94]. Tekniikka tarjoaa enemmän rakennetta kuin heuristinen evaluointi, mutta vähemmän kuin kognitiivinen läpikäynti.

Heuristiseen läpikäyntiin on lainattu kognitiivisesta läpikäynnistä kaksi ideaa: heuristisessa läpikäynnissä yhdistetään lista käyttäjätehtävistä ja lista kysymyksistä joiden tarkoituksena on saada käyttäjä keskittymään järjestelmän tärkeisiin ominaisuuksiin. Käytettävyyden arviointi on tällöin kaksivaiheinen prosessi. Arvioijia ohjataan tärkeysjärjestyksessä olevalla käyttäjätehtävien listalla, listalla käytettävyysheuristiikoista ja listalla "ajatuksen keskittäviä" kysymyksiä.

Ensimmäisessä vaiheessa arvioijat tutkivat tehtäviä järjestetystä listasta. Toisessa vaiheessa arvioijat ohjeistetaan tehtäväkeskeisesti järjestelmän käyttöön ja heille selitetään myös käyttäjäheuristiikat. Toisessa vaiheessa arvioija voi tutkia mitä vain järjestelmän näkökulmaa etsien samalla käytettävyysongelmia. Molemmassa vaiheissa löydetty on-

gelmat dokumentoidaan ja jokaiseen ongelmaan liitetään arvio sen vakavuudesta, jotta pystytään päättämään mitkä ongelmat tulisi korjata ensin. Arvioijat vertaavat myöhemmin heidän itsenäisesti merkattuja arviointejaan johonkin yleisesti tunnettuun arviointiskaalaan.

Vaihe 1: Tehtäväorientoitunut arviointi. Tärkeysjärjestykseen järjestetty tehtävälista opastaa tätä arviointiprosessin vaihetta. Listan tulisi sisältää usein toistuvia, yleisiä tai tärkeitä tehtäviä ja voi myös sisältää ylimääräisiä tehtäviä jotka auttavat arvioijaa ymmärtämään järjestelmän kokonaiskuvaa. Jokaiseen tehtävään liitetään lisäksi prioriteettiluku. Tämä luku liittyy yleensä tehtävän tärkeyteen tai siihen kuinka usein käyttäjä suorittaa kyseisen tehtävän. Arvioijat voivat suorittaa mitä tehtäviä haluavat, missä järjestyksessä vain, mutta tehtävien järjestysnumeron tulisi opastaa arvioijaa kun he etsivät sopivaa tehtävää tutkittavaksi. Kun arvioijat tutkivat tehtäviä heitä opastaa neljä ajatusta keskittävää kysymystä jotka on saatu kognitiivisesta läpikäynnistä:

- Tietävätkö käyttäjät mitä heidän tulee tehdä seuraavaksi? Onko mahdollista että he eivät yksinkertaisesti pysty tajuamaan mitä tehdä seuraavaksi?
- Huomaavatko käyttäjät että ruudulla on näkyvillä toiminto (esimerkiksi menu, linkkipainike) joka auttaa heitä suorittamaan tehtävän seuraavan osan? Onko mahdollista että kyseinen toiminto on piilotettu tai terminologia ei ole yhteneväinen sen kanssa mitä käyttäjä etsii. Molemmissa tapauksissa tarvittu toiminto löytyy mutta käyttäjä ei löydä sitä.
- Kun käyttäjä on löytänyt toiminnon, tietävätkö he kuinka käyttää sitä (esimerkiksi klikata sitä, tuplaklikata sitä)
- Jos käyttäjät suorittavat oikean toiminnon, näkevätkö he että edistystä on tapahtunut tehtävän suorittamisen hyväksi? Tarjoaako järjestelmä sopivaa palautetta? Jos ei, käyttäjät eivät ehkä ole varmoja että äsken suoritettu toiminto oli oikein.

Vaihe 2: Vapaamuotoinen arviointi. Toisessa vaiheessa arvioijat voivat tutkia mitä vain järjestelmän aspektia/näkökulmaa. Kuitenkin, heitä opastaa se tieto minkä he kasvattivat ensimmäisessä vaiheessa: lista käyttäjätehtävistä, käytettävyyshauristiikat ja ajatuksen keskittävät kysymykset. Mitä vain käytettävyyshauristiikkoja voidaan käyttää.

3.7 Tekniikoiden vertailua

Tarkastamiseen perustuvat arviointitekniikat ovat suosittuja koska ne vaativat vähemmän formaalia koulutusta, ovat nopeita ja niitä voidaan käyttää koko tuotteen kehityskaaren läpi. Bastien ja Scapin määrittivät kolme tunnusmerkkiä jotka pitäisi pitää mielessä kun verrataan käyttöliittymän arviointitekniikoita [Bastien95]:

-validisuus: arvioijan kyky keskittyä tiettyyn, edeltä määriteltyyn käyttöliittymän ominaisuuteen

-kattavuus: arviointi on kattava jos keskitytään tutkimaan käyttöliittymää mahdollisimman laajalti

-luotettavuus: arviointi on luotettavaa jos ne johdonmukaisesti tarjoavat samoja tuloksia samojen edellytysten vallitessa

Kaikki arvot ovat numeroiden 0 ja 1 välillä. Optimaalinen arvo on 1. HE=heuristinen evaluointi, KL=kognitiivinen läpikäynti, HL=heuristinen läpikäynti. [Sears94, s230, taulukko 6]

Arvioijien lukumäärä	Validiteetti			Kattavuus			Luotettavuus		
	HE	KL	HL	HE	KL	HL	HE	KL	HL
2	0.75	0.94	0.96	0.51	0.41	0.57	0.83	0.77	0.81
3	0.73	0.93	0.95	0.66	0.51	0.68	0.88	0.84	0.87
4	0.71	0.92	0.94	0.76	0.56	0.75	0.91	0.88	0.91
5	0.69	0.91	0.93	0.85	0.61	0.80	0.94	0.92	0.94

Taulukko 1. Heuristisen evaluoinnin, kognitiivisen läpikäynnin ja heuristisen läpikäynnin vertailu

Searsin tutkimusten mukaan (taulukko 1) heuristinen evaluointi näyttäisi olevan vähemmän validi kuin heuristinen läpikäynti tai kognitiivinen läpikäynti. Tämä johtuu

siitä huomattavan suuresta määrästä väärin tunnistettuja ongelmia joita arvioijat löysivät käyttäessään heuristisen arvioinnin menetelmiä. Kognitiivinen läpikäynti vaikuttaisi olevan vähemmän kokonaisvaltainen kuin heuristinen läpikäynti ja heuristinen arviointi. Tämän näkee siitä vähäisestä keskinkertaisten ja pienien ongelmien määrästä joita arvioijat löysivät soveltaessaan kognitiivisen läpikäynnin tekniikoita. Heuristinen läpikäynti ja heuristinen arviointi vaikuttavat tulosten mukaan olevan luotettavampia kuin kognitiivinen läpikäynti, pienellä arvioijamäärällä testatessa. Erot pienenevät kun käytetään suurempaa arvioijamäärää.

Käytettävyyden tarkistustekniikat ovat suosittuja sillä ne antavat käyttöliittymän suunnittelijoille mahdollisuuden arvioida heidän suunnitteluaan nopeasti ilman testikäyttäjien apua. Tällaiset arvioinnit eivät löydä kaikkia käytettävyyso ongelmia, mutta voivat nopeasti tunnistaa monia ongelmia jotka olisi voitu löytää perinteisessä käytettävyydestauksessa. Tietenkin ne voivat myös tunnistaa joitakin asioita ongelmiksi jotka eivät todellisuudessa haittaa käyttäjiä. Tällaiset väärin tunnistetut ongelmat voivat hidastaa itse käytettävyyden arviointia ja tulosten analysointia sillä ne tulee poistaa käytettävyyso ongelmien listasta. On siis tärkeää ymmärtää kuinka monta todellista ongelmaa löydetään sekä kuinka monta väärin tunnistettua ongelmaa löydetään.

Tekniikka on validi jos arvioijat pystyvät keskittymään ongelmiin jotka ovat relevantteja. Tämä tarkoittaa sitä, että arvioijien ei tulisi tunnistaa joitakin ongelmia käytettävyyso ongelmiksi siinä tapauksessa jos ongelmat eivät itse asiassa vaikuta käyttäjiään. Validisuutta voi mitata laskemalla suhde todellisten käytettävyyso ongelmien ja kaikkien löydettyjen käytettävyyso ongelmiksi väitettyjen ongelmien välillä.

Kattavuus tarkoittaa sitä että arvioijat ovat kykeneviä arvioimaan kaikkia käyttöliittymän näkökulmia. Kattavuutta voi mitata vertaamalla todellisten löydettyjen ongelmien ja todellisten olemassa olevien ongelmien suhdetta.

Luotettavuus tarkoittaa sitä että samoja ongelmia löytyy samojen olosuhteiden valitessa jos testauksen uusii. Luotettavuutta voi testata esimerkiksi selvittämällä löytävätkö eri arvioijat tai arvioijaryhmät samoja määriä ongelmia heidän käyttäessä samaa tekniikkaa. Tätä voi mitata verrattaessa löydettyjen ongelmien standardia poikkeamaa (engl. standard deviation) keskimääräiseen löydettyjen ongelmien lukuun.

Yhtenä, huonona, tuloksena tästä vapaamuotoisesta arviointitekniikasta voi olla liian pelkistetty arviointi. Vähemmän kokeneet arvioijat arvioivat jokaisen ruudun tarkasta-

malla samalla heuristiikkojen listaa. Jos heuristiikkaa rikotaan, ongelma huomataan ja kirjataan ylös. Jos heuristiikkaa ei rikota, ruutua pidetään ongelmapaana. Tämä johtaa siihen että tunnistetaan jokainen heuristisen säännön rikkominen käytettävyysongelmana, vaikka näin ei aina olekaan [Sears94].

3.8 *Validin datan kerääminen*

Kun käytettävyyttä tutkitaan, kannattaa miettiä mitä vastauksia halutaan kysymyksiin: Miksi järjestelmän käytettävyyttä halutaan testata? Mitä haluat oppia tästä testistä? Mistä tiedetään että järjestelmä on hyvä?

Ilman testistä halutun tiedon keräämissuunnitelmaa kerätyn tiedon läpikäyminen ja merkityksellisten tulosten analysointi on hankalaa. Tiedon keräämiseen voi käyttää esimerkiksi tiedon keräämistaulukkoa (engl. data crosswalk form) [Rojek94]. Tätä taulukkoa käytetään pääosin testin suunnitteluvaiheessa mutta siitä on apua myös myöhemmin tulosten analysointivaiheessa jolloin siihen voi palata. Taulukko koostuu neljästä sarakkeesta:

-Testikysymykset: Testikysymykset kannattaa luokitella niiden ominaisuuksien mukaisesti. Kysymys voi olla mielipide-, oppimis-, strategia-, ymmärtämis- tai ongelmanratkaisukysymys. Se voi liittyä myös tuotteen ominaisuuksiin tai käyttäjäryhmään.

-Datatyyppi: Minkälainen data vastaa kysymykseen. Onko se jotain mitä käyttäjä tulee tekemään? Onko se mielipidekysymys? Pitääkö minun tietää kauanko aikaa käyttäjällä kuluu tehtävän suorittamiseen? Tarvitsenko kvalitatiivista vai kvantitatiivista tietoa? Pitääkö minun pitää kirjaa ongelmista ja virheistä joita käyttäjä tekee? Tyypillinen kerätty data on: Suoritusdataa (virheiden tyyppi, virheiden lukumäärä, polku, suorituksen kesto, onnistumis-/epäonnistumis-aste), Verbaalinen data (ääneen ajattelu), Järjestelmästä kaapattu data (nauhoitetaan hiiren klikkaukset), Kenttädata (käyttäjätiedot kerätään järjestelmän todellisessa käyttöympäristössä eikä käytettävyysslaboratoriossa), haastatteludata, kyselylomakedata.

-Todistusaineisto: Täytyy tunnistaa ja määritellä tulkinnalle avoimet termit. Eri termit kuten (helppo, hyvä, luonnollinen, vaikea, hankala..) tulkitaan eri tavoin henkilöstä riippuen. Tällöin tarvitaan yhtenäinen skaala jolla eri käyttäjien mielipiteitä voidaan tulkita samalla tavoin.

-Tehtäväideat: -

Taulukko on systemaattinen tapa nähdä, minkälaista tietoa käytettävyytutkimuksesta tulee kerätä.

4 Tietojenkäsittelytieteen laitoksen verkkosivujen käytettävyytutkimus

4.1 Määritelmä ja käytetyt tekniikat

4.2 Testitapaukset

4.3 Heuristinen läpikäynti

4.4 Käytettävyystestaus

5 Käytettävyytutkimuksen tulokset ja johtopäätökset

6 Yhteenveto

Lähteet

- Aluker05 Aluker S., Allendoerfer K., Adapting the Cognitive Walkthrough Method to Assess the Usability of a Knowledge Domain Visualization, IEEE Symposium on Information Visualization 2005, sivut 195-202.
- Bastien95 Bastien, J. M. C., & Scapin, D. L. (1995). Evaluating a user interface with ergonomic criteria. *International Journal of Human-Computer Interaction*, 7, sivut 105-121.
- Bevan97 Bevan, N., Azuma, M., *Quality in Use: Incorporating Human Factors into the Software Engineering Lifecycle*, 1997, sivut 169-179.
- Dringus05 Dringus L. P., Cohen M. S., An Adaptable Usability Heuristic Checklist for Online Courses, *Frontiers in Education*, 2005. FIE '05. Proceedings 35th Annual Conference
- ISO11 ISO 9241-11:1998, Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability.
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnnumber=16883.
- ISO07 ISO 9241-171:2008, Ergonomics of human-system interaction -- Part 171: Guidance on software accessibility,
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnnumber=39080.
- Rojek94 Jill Rojek, Amt Kanerva, A Data-Collection Strategy for Usability Tests, *IEEE Transactions on Professional Communication*, Vol. 37, No. 3, Syyskuu 1994, sivut 149-156.
- Kantner94 Laurie Kantner, Techniques for Managing a Usability Test, *IEEE Transactions of Professional Communication*, vol. 37, no.3, Syyskuu 1994, sivut 143-148.
- Nielsen92 Nielsen J., Finding usability problems through heuristic evaluation, *ACM CHI'92 Conf*, sivut 378-380.

- Nielsen93 Nielsen J., Usability Engineering. Academic Press, New York, USA, 1993.
- Sears97 A. Sears, Heuristic Walkthroughs: Finding the Problems Without the Noise, 1997, *International Journal of Human-Computer Interaction*, sivut 213-234.
- Wang08 Wang X., Using Cognitive Walkthrough Procedure to Prototype and Evaluate Dynamic Menu Interfaces: a Design Improvement, 2008 IEEE, sivut 76-80.
- Wharton94 Wharton, C., Rieman, J., Lewis, C., Polson, P., The Cognitive Walkthrough Method: A Practitioner's Guide, 1994. In *Usability Inspection Methods*, J. Nielsen and R.L. Mack (Eds.), New York: John Wiley & Sons, sivut.105-141.