

Käytettävyyden arviointimenetelmistä

Sami Taipale

12.6.2000

Joensuun yliopisto
Tietojenkäsittelytiede
Kandidaatintutkielma

Tiivistelmä

Käytettävyyden arviointi on ohjelmiston käytettävyyssominaisuuksien tutkimista. Sen tavoitteena parantaa järjestelmän käytettävyyttä. Tässä tutkielmassa tarkastellaan käytettävyyden arviointia ja esitellään kolme siihen käytettävää menetelmää; heuristinen arviointi, kognitiivinen läpikäynti ja muodollinen arviointimenettely. Näitä menetelmiä verrataan toisiinsa niiden resurssivaativuuden, ohjelmistotuotantoprosessiin sijoittumisen sekä arvioijamäärän ja arvioijien koulutuksen mukaan.

Sisältö

| | | |
|----------|---|-----------|
| 1 | Johdanto | 1 |
| 1.1 | Käsitteitä | 1 |
| 1.2 | Käytettävyyden arviointi | 2 |
| 2 | Yleiskatsaus käytettävyyden arviointiin | 4 |
| 2.1 | Käytettävyyden arvioinnin tavoitteet | 4 |
| 2.2 | Arviointimenetelmät | 5 |
| 2.3 | Menetelmän valintaan vaikuttavia tekijöitä | 7 |
| 2.3.1 | Arvioinnin tavoitteet ja resurssit | 7 |
| 2.3.2 | Tehtävänasettelu | 7 |
| 2.3.3 | Arvioijien kokemus | 8 |
| 2.3.4 | Yksilö- ja ryhmäarviointi | 9 |
| 2.3.5 | Arviointimenetelmän käytettävyys | 9 |
| 2.4 | Käytettävyysarviointi käytettävyysuunnittelussa | 9 |
| 2.4.1 | Arviointitulokset | 10 |
| 2.4.2 | Ongelmaraporttien tehokkuus | 11 |
| 2.4.3 | Menetelmien käyttö ohjelmistotuotannon elinkaarella | 11 |
| 2.4.4 | Kustannushyöty | 12 |
| 2.5 | Käytettävyyden arviointimenetelmien tutkimuksesta | 12 |
| 3 | Heuristinen arviointi | 14 |
| 3.1 | Arvioijat | 15 |
| 3.2 | Arvioinnin kulku | 17 |
| 3.2.1 | Arviointi | 17 |
| 3.2.2 | Jälkipalaveri | 19 |
| 3.3 | Käytettävyysongelmien vakavuus | 19 |

| | | |
|----------|--|-----------|
| 3.4 | Heuristisella arvioinnilla löydettäviä käytettävyyssongelmia | 21 |
| 3.5 | Yhteenveto | 22 |
| 4 | Kognitiivinen läpikäynti | 23 |
| 4.1 | Arvioijat | 23 |
| 4.2 | Arvioinnin kulku | 24 |
| 4.2.1 | Läpikäynnin valmistelu | 25 |
| 4.2.2 | Läpikäynti | 26 |
| 4.2.3 | Tiedonkeruu arvioinnin aikana | 28 |
| 4.3 | Yhteenveto | 31 |
| 5 | Muodollinen arviointimenetelmä | 33 |
| 5.1 | Arvioijat | 33 |
| 5.2 | Arvioinnin kulku | 34 |
| 5.2.1 | Tehtävien läpikäynti | 34 |
| 5.2.2 | Ongelmien etsiminen | 36 |
| 5.2.3 | Kuusi logistista askelta | 41 |
| 5.3 | Yhteenveto | 44 |
| 6 | Vertailua | 45 |
| 6.1 | Käytettävyyden arviointi vs. käytettävyytestaus | 45 |
| 6.2 | Arvioijat | 47 |
| 6.3 | Sijoittuminen elinkaarelle | 48 |
| 6.4 | Resurssivaativuus | 48 |
| 7 | Yhteenveto | 49 |
| | Viitteet | 50 |

1 Johdanto

Tässä luvussa selvennetään tutkielmassa käytettäviä käsitteitä ja tarkastellaan käytettävyyden arviointia yleisellä tasolla.

1.1 Käsitteitä

Käyttöliittymä (user interface) on järjestelmän osa, johon käyttäjä on jollain tavalla yhteydessä (Bradford 1994).

Käytettävyys (usability) koostuu viidestä osa-alueesta: opittavuudesta, käytön tehokkuudesta, muistettavuudesta, virheettömyydestä ja käyttäjän tyytyväisyydestä (Nielsen 1993). *Opittavuus* kertoo, kuinka helppoa uuden käyttäjän on oppia järjestelmän yksinkertaiset toiminnot. *Tehokkuus* on kokeneen käyttäjän mahdollisuutta suorittaa tehtävät nopeasti. *Muistettavuus* ilmenee siinä, kuinka helppoa käyttäjän on tauon jälkeen palata käyttämään järjestelmää. *Virheettömyys* tarkoittaa käyttäjän minimaalista mahdollisuutta tehdä virheitä järjestelmän käytössä. Virheistä tulisi olla myös helppoa toipua. *Tyytyväisyys* on käyttäjien tyytyväisyyttä järjestelmän käyttöön. Nämä osa-alueet saattavat olla ristiriidassa keskenään.

Käytettävyysuunnittelu (usability engineering) on joukko toimintoja (taulukko 1), joita tehdään koko järjestelmän elinkaaren aikana (Nielsen 1993). Toiminnot ovat osin rinnakkaisia.

Taulukko 1: Käytettävyyssuunnittelun toiminnot

| |
|--|
| Loppukäyttäjiin perehtyminen |
| Kilpailutilanteen kartoitus |
| Käytettävyystavoitteiden asettaminen |
| Rinnakkainen suunnittelu |
| Osallistuva suunnittelu |
| Käyttöliittymän koordinoitu suunnittelu |
| Suunnitteluperiaatteiden ja heuristiikkojen soveltaminen |
| Protoilu |
| Käyttöliittymän arviointi |
| Iterointi |
| Toimitettujen järjestelmien seuranta |

1.2 Käytettävyyden arviointi

Käytettävyyden arviointia voidaan suorittaa menetelmillä, joissa arvioijat tutkivat järjestelmän käytettävyyssominaisuuksia. Arvioijat vaihtelevat taustoiltaan. He voivat olla joko käytettävyyssiantuntijoita, käyttöliittymätyylin tuntevia järjestelmän kehittäjiä tai sovellusalueen tuntevia loppukäyttäjiä. Arviointimenetelmillä on hieman eri tavoitteet ja ne noudattavat erilaisia ohjenuoria, mutta yhteistä kaikille on luottaminen arvioijan päätöksiin käyttöliittymän kehittämisessä.

Mackin ja Nielsenin (1994) mukaan neljä perustapaa arvioida (evaluate) käytettävyyttä ovat automaattinen, empiirinen, formaali teoreettinen ja epäformaali. *Automaattinen arviointi* tapahtuu antamalla arviointiohjelmistolle syötteeksi käyttöliittymän määrittelyt, joista tämä laskee käytettävyyttä kuvaavia lukuja, esimerkiksi syötejonojen pituutta. Vielä nykytek-

niikalla automaattiset menetelmät eivät kuitenkaan toimi. *Empiirinen arviointi* on järjestelmän käytettävyyden testaamista loppukäyttäjillä, joka on puolestaan hyvä, mutta kallis ja työläs tapa. *Formaalit teoreettiset menetelmät* käyttävät apunaan tarkkoja malleja ja kaavoja käytettävyydsarvojen laskemiseen. Tämä tapa soveltuu huonosti monimutkaisille hyvin interaktiivisille käyttöliittymille, sillä vuorovaikutusta ei pystytä kuvaamaan tarkasti. *Formaalit teoreettiset menetelmät* vaativat myös pitkällisen koulutuksen. *Epäformaalit menetelmät* perustuvat heuristiikoille sekä arvioijien kokemukselle käytettävyyden arvioinnista.

Tässä tutkielmassa tarkastellaan kolmea epäformaalia arviointimenetelmää; heuristista arviointia luvussa 3, kognitiivista läpikäyntiä luvussa 4, sekä muodollista arviointimenetelmää¹ luvussa 5.

¹Nimestään huolimatta muodollinen arviointimenetelmä (formal inspection method) kuuluu edellämainitussa jaottelussa epäformaalien arviointimenetelmien joukkoon

2 Yleiskatsaus käytettävyyden arviointiin

Tässä luvussa käydään läpi käytettävyyden arvioinnin tavoitteet ja esitellään erilaisia käytettävyysoongelmia. Lisäksi tarkastellaan käytettävyyden arvioinnin alan tutkimusta ja kehitettyjä arviointimenetelmiä.

2.1 Käytettävyyden arvioinnin tavoitteet

Käytettävyyden arvioinnin tarkoituksena on löytää arvioitavasta järjestelmästä käytettävyysoongelmia ja antaa niiden pohjalta parannusehdotuksia. *Käytettävyysongelma* on järjestelmän käytettävyyttä heikentävä piirre. Käytettävyys puolestaan muotoutuu järjestelmän oppimisen helppoudesta, muistettavuudesta, opitun järjestelmän käytön tehokkuudesta ja miellyttävyydestä sekä virheiden yleisyydestä ja vakavuudesta. Suuri osa käytettävyyden arviointiin kuuluvasta ajasta menee arvioinnissa löydettyjen käytettävyysongelmien luokitteluun ja laskemiseen. Käytettävyysongelmien analysointi vaatiikin tarkan määrittelyn käytettävyysongelmalle, jottei turhaan käsitellä ominaisuuksia, jotka eivät ole virheitä. Nyrkkisääntönä käytettävyysongelmalle on, että jos käyttöliittymää jostain kohdasta muuttamalla saadaan aikaan parannusta jollain käytettävyyden osaluueella, on kyseessä olevassa kohdassa käytettävyysongelma.

Arvioinnin jälkeen käytössä on lista käytettävyysongelmista, jonka pohjalta voidaan suunnitella järjestelmän käytettävyyden parantamista ongelmia poistaen. Tähän tarvitaan lisätietoa ja -analyyseja, jotka liittyvät pelkkää käytettävyyden arviointia laajempaan käytettävyyssuunnittelun alaan. Ongelmaraportin muodostamisessa tulisi kiinnittää huomiota kolmeen tärkeään seikkaan: korjaus- ja uudelleensuunnitteluehdotuksiin, käy-

tettävyysongelmiin priorisointiin ja ongelmien korjauksen kustannusarvioihin.

Käytettävyyden arviointimenetelmiä voidaan käyttää myös koulutukseen. Osatavoitteena ryhmänä tehtävässä arvioinnissa voi olla tiedon jakaminen ja ryhmän jäsenten välisen kommunikoinnin edistäminen (Wixon et al. 1994). Arviointiryhmän jäsenenä toimiminen ja sitä kautta saatava kokemus arviointimenetelmillä löytyvistä ongelmista on hyvää koulutusta kokemattomille arvioijille. Samalla arvioijalle kehittyy entistä käyttäjäkeskeisempi näkökulma järjestelmän kehittämiseen.

2.2 Arviointimenetelmät

Käytettävyyden arviointimenetelmä on ainakin seuraavat kahdeksan menetelmää kattava yleisnimitys.

- *Heuristinen arviointi* (heuristic evaluation) on epämuodollisin menetelmä, jossa käytettävyysasiantuntijat arvioivat järjestelmän käytettävyyttä vertaamalla sen yksittäisiä elementtejä yleisesti hyväksytyihin käytettävyysperiaatteisiin eli heuristiikkoihin.
- *Tarkastuslista-arviointi* (guideline review) perustuu käyttöliittymän vertaamiseen kattavaan tarkastuslistaan. Tarkastuslistoissa on usein tuhatkin tarkastettavaa kohtaa, joten tämä menetelmä vaatii suurta asiantuntemusta ja on hyvin harvinainen käytännössä.
- *Pluralistinen läpikäynti* (pluralistic walkthrough) on kokous, jossa käyttäjät, järjestelmän kehittäjät ja käytettävyysihmiset käyvät järjestel-

mää läpi skenaarioiden avulla ja keskustelevat samalla elementteihin liittyvistä käytettävyyssasioista.

- *Yhtenäisyysarviointi* (consistency inspection) keskittyy arvioimaan järjestelmän yhdenmukaisuutta esim. samaan tuoteperheeseen kuuluvien muiden järjestelmien kanssa.
- *Standardinmukaisuuden arviointi* (standard inspection) on jonkin käyttöliittymästandardin tuntijan suorittama käyttöliittymän vertaaminen olemassaolevaan standardiin.
- *Kognitiivinen läpikäynti* (cognitive walkthrough) käyttää hyväkseen yksityiskohtaisesti määriteltyä menetelmää käyttäjän ongelmanratkaisuprosessin simuloinnissa. Kiinnostuksen kohteena on, voidaan-ko simuloitulla käyttäjän tavoitteella ja toimintojen muistamisella päästä oikeaan tulokseen.
- *Muodollinen käytettävyyden arviointimenetelmä* (formal usability inspection) on ajatukseltaan samantyylinen kuin ohjelmistotuotantoprosessissa käytettävät tarkastusmenettelyt. Osallistujilla on tarkoin määritellyt roolit ja menettely käydään läpi kuudessa vaiheessa.
- *Piirrearviointi* (feature inspection) keskittyy järjestelmän toimintoihin ja erityisesti siihen, kattavatko ne aiottujen loppukäyttäjien tarpeet. Piirrearviointiin voi myös sisältyä piirteen suunnittelu.

2.3 Menetelmän valintaan vaikuttavia tekijöitä

Arviointimenetelmää valittaessa joudutaan miettimään arvioinnin tavoitteita ja resursseja, tehtävänasettelua, käytettävissä olevia arvioijia ja heidän hankkimistaan sekä arviointimenetelmän käytettävyyttä.

2.3.1 Arvioinnin tavoitteet ja resurssit

Tavoitteista kerrottiin jo aiemmin yleisellä tasolla, mutta järjestelmän kehityksen yhteydessä tarvitaan kuitenkin tarkempia tavoitteita, jotka ottavat huomioon järjestelmän kehityksen rajoitukset ja kompromissit. Tarkat tavoitteet helpottavat arviointimenetelmän valintaa.

Eri arviointimenetelmillä on erilaiset resurssit. Heuristisessa arvioinnissa käytetään heuristiikkoja arvioinnin apuna, kognitiivisessa läpikäynnissä on apuna kysymyslomake, jolla selvitetään käyttäjien tavoitteiden ja käyttöliittymän toimintojen välistä suhdetta ja standardiarvioinnissa on taas omat ohjeensa. Näiden ohjenuorien valinta ratkaisee koko arviointimenetelmien valinnan. Tämä päätös onkin tärkeä, joten siinä pitää tarkastella laajasti koko menetelmää aina opittavuudesta tulosten hyväksikäyttömahdollisuuksiin.

2.3.2 Tehtävänasettelu

Tehtävänasettelu erottaa myös arviointimenetelmiä toisistaan. Vaihtoehtoina voivat olla esimerkiksi tarkasti määritellyt tehtävät tai yleisohjeet. Tarkat tehtävät esittävät, kuinka loppukäyttäjän oletetaan järjestelmää käyttävän. Tarkoilla tehtävillä voidaan antaa tiettyjä käyttöliittymän kohteita arvioitavaksi ja lisäksi se tarjoaa käyttöliittymään tehtäväsuuntautuneen

näkökulman. Käyttäjien käyttäytyminen ei tosin aina ole näin sidoksissa tietyn tehtävän suorittamiseen, vaan he tutkimalla, päättelemällä ja kokeilemalla luovat itselleen mentaalimallin käyttöliittymästä. Yleisohjeet antavat arvioijalle mahdollisuuden tutustua koko järjestelmään ja tehdä näin löytöjä muistakin osista. Näiden kahden tavan välillä tulisikin tehdä kompromissi, jolla varmistetaan oikeanlaatuinen järjestelmän käyttö ja samalla sallitaan omien kokeilujen tekeminen.

2.3.3 Arvioijien kokemus

Arvioijien kokemuksella käytettävyyden alalta on suuri vaikutus arvioinnin tulokseen, kokeneet arvioijat tuottavat paremmin hyödynnettävissä olevia ongelmaraportteja (Mack & Nielsen 1994). Aiemmin kuitenkin todettiin, että arvioinnin tavoitteet voivat myös olla kokonaan tai osittain koulutuksellisia, jolloin arviointiin otetaan mukaan käytettävyyden alalla kokemattomia järjestelmän kehittäjiä saamaan käyttäjäkeskeistä näkökulmaa. Kustannusten säästön toivossa voidaan arviointi antaa kokonaan järjestelmän kehittäjien tai jopa loppukäyttäjien tehtäväksi löysemällä valvonnalla ja muodollisuudella.

Arvioijien valinta riippuu siis arvioinnin tavoitteista. Kannattaa käyttää hyväksi asiantuntijoiden harkittuja arvioita höystettynä kokemattomien arvioijien tuoreilla näkemyksillä. Suurin panostus arvioijien valinnassa tehdään arvioijien etsimiseen, kouluttamiseen ja värväämiseen kulutetussa ajassa ja vaivassa.

2.3.4 Yksilö- ja ryhmäarviointi

Yksilö- ja ryhmäarvioinneista ei voi yksiselitteisesti sanoa, kumpi on ylipäänsä parempi. Yksilöarvioinnissa arvioija muodostaa oman mielipiteensä käyttöliittymästä ilman muiden vaikutusta. Arvioija voi myös suorittaa arvioinnin hänelle parhaiten sopivaan aikaan, ilman suuria kokousjärjestelyjä. Toisaalta ryhmäarvioinnissa osallistujat voivat auttaa toisiaan huomaamaan ongelmia, joita ei yksin olisi huomattu. Samoin voidaan ongelmasta keskustelemalla huomata, että se ei olekaan ongelma. Yksilö- ja ryhmäarvioinnit voidaan myös suorittaa peräkkäin tai arvioinnin eri vaiheissa.

2.3.5 Arviointimenetelmän käytettävyys

Arviointimenetelmän opittavuus ja käytettävyys on myös otettava huomioon. Toiset menetelmät, kuten heuristinen arviointi, ovat helpompia käyttää ja oppia kuin toiset. Esimerkiksi kognitiivisen läpikäynnin ongelma on sen ajattelutavan vaikeus. Tällä hetkellä yksinkertaisempi on paljastunut vertailuissa myös tehokkaammaksi (Mack & Nielsen 1994). Kognitiivisesta läpikäynnistä saadaan todellinen hyöty irti vasta pidemmän ajan kuluttua, kun sen käytöstä on saatu riittävästi kokemusta.

2.4 Käytettävyysarviointi käytettävyysuunnittelussa

Tässä kappaleessa tarkastellaan käytettävyysarvioinnin sijoittumista laajempaan käytettävyysuunnittelun alaan.

2.4.1 Arviointitulokset

Arviointitulosten oikeellisuus tarkoittaa sitä, kuinka hyvin tulokset ennustavat loppukäyttäjän kokemia käytettävyysoongelmia. Oikeellisuus on tärkeää, sillä tulokset aiheuttavat järjestelmän muuttamista, eikä kukaan halua muuttaa toimivaa ohjelmaa turhaan. Arviointimenetelmien tulokset eivät ole niin hyviä, että niitä voisi suositella käytettäväksi loppukäyttäjille testaamisen sijaan. Tavallisesti ne löytävät noin 30 - 50 % ongelmista (Jeffries et al. 1991; Desurvire 1994; Karat 1994).

Kaikki löydetyt ongelmat eivät ole samanarvoisia. Vakavien käytettävyysongelmien löytäminen on erityisesti tärkeää. Ongelman vakavuus päätellään sen yleisyydellä ja vaikutuksella käyttäjään – häiritsevyydestä tehtävän suorittamisen estymiseen.

Käytettäessä arviointia loppukäyttäjillä testaamisen sijaan, kiinnostaa arvioijien määrä suhteessa löydettyihin ongelmiin. Ryhmäarviointi on yleisesti ottaen yksilöarviointia tehokkaampaa, johtuen ryhmätyöllä saavutettavasta työtehon kasvusta (Hackman & Morris 1975; Karat et al. 1992). Arvioijaryhmät tuottavat myös tarkempia ongelmaraportteja (Karat et al. 1992). Heuristisella arvioinnilla 4-5 arvioijaa löytää suurimmassa osassa tapauksista 80 % ongelmista, jotka löytyisivät paljon suuremmalla arvioijajoukolla tai testattaessa järjestelmää loppukäyttäjillä (Nielsen 1994b). Nielsenin (1994b) mukaan heuristinen arviointi on kustannustehokkain käytettäessä kolmesta viiteen arvioijaa.

Arviointimenetelmissä joudutaan aina luottamaan erilaisiin oletuksiin siitä, kuinka löydetyt ongelmat suhtautuvat todellisiin, loppukäyttäjän ko-

kemiin ongelmiin. Järjestelmän kehityksessä on paljon muutakin huomioon otettavaa kuin käytettävyys. Tämän perusteella pitäisikin arviointimenetelmien tehokkuutta arvioida sen perusteella, kuinka paljon ne lisäävät loppukäyttäjän tyytyväisyyttä koko tuotteeseen.

2.4.2 Ongelmaraporttien tehokkuus

Ongelmaraporttien käyttöä järjestelmän käytettävyyden parantamisessa edesauttaa kirjaukset korjausmahdollisuuksista ja uudelleensuunnitteluehdotuksista. Käytettävyyden arviointimenetelmät ovat parempia löytämään kuin korjaamaan ongelmia. Ongelmien laajat kuvaukset auttavat paljastamaan taustalla olevia syitä ja seurauksia ja vakavuusarviointit auttavat priorisoinnissa. Priorisointi puolestaan kiinnittää huomiota tärkeimpiin ongelmiin, jotka täytyy korjata. Näin ei tuhlaata turhaan aikaa sellaisten ongelmien korjaamiseen, joita käyttäjä ei edes huomaisi.

Arviointimenetelmillä voi olla vaikeaa saavuttaa samanlaista uskottavuutta kuin esimerkiksi videoidulla käyttäjätestauksella. Resursseista päättävät ihmiset uskovat helposti korjaustarpeen nähdessään videolta käyttäjän tuskailavan ongelman parissa. Ongelmaraportti on harvoin yhtä vakuuttava. Järjestelmän kehittäjien ottaminen mukaan arviointiprosessiin voi auttaa vakuuttamaan korjaustarpeesta.

2.4.3 Menetelmien käyttö ohjelmistotuotannon elinkaarella

Käytettävyyden arviointimenetelmät vaativat käyttöliittymän olemassaolon jotta niitä voitaisiin käyttää. Tämä ei kuitenkaan tarkoita sitä, että sen pitäisi olla toteutettuna, sillä käyttöliittymää voidaan arvioida myös sen määrittelyn perusteella. Tällöin arviointia voidaan tehdä suhteellisen ai-

kaisessa vaiheessa, varsinkin verrattuna testaamiseen loppukäyttäjillä. Arviointimenetelmät soveltuvat hyvin iteroivaan suunnitteluprosessiin, jossa niitä voidaan käyttää yhdessä muiden menetelmien, kuten loppukäyttäjillä testaamisen, kanssa.

2.4.4 Kustannushyöty

Arviointimenetelmän valinta on kompromissi käytettävissä olevan ajan ja resurssien välillä. Arvioinnin kustannukset suhteessa hyötyyn verrattuna testaamiseen loppukäyttäjillä tai arvioimatta jättämiseen, on herättänyt keskustelua. Heuristinen arviointi on saanut alkunsa Nielsenin ”halvan käytettävyyden” (discount usability) periaatteesta: monet käytettävyyssuunnittelun toiminnot ovat vaikeita ja aikaavieviä suorittaa, mutta halvalla ja nopeastikin voi saada aikaan tuloksia (Nielsen 1989; Nielsen 1990). Kognitiivinen läpikäynti on puolestaan muodollisempi ja kalliimpi menetelmä käyttää, sillä se vaatii psykologisia tietoja ihmisen ongelmanratkaisusta ja tavoitehakuisesta toiminnasta. Tärkein päätös kuitenkin on, käytetäänkö jotain käytettävyyteen liittyvää tarkastelua vai ei. Useimmilla arviointimenetelmillä, olivat ne sitten jostain näkökulmasta katsottuna kuinka epätyytyttäviä tahansa, saadaan käyttökelpoisia tuloksia käyttöliittymän kehittämiseen.

2.5 Käytettävyyden arviointimenetelmien tutkimuksesta

Tässä kappaleessa tarkastellaan käytettävyyden arviointimenetelmiin liittyvän tutkimuksen suuntautumista.

Yksi tutkimuksen kohde on menetelmien kehittäminen. Tavoitteena on saada ne tuottamaan enemmän ja parempia ongelmaennusteita. Arviointi-

prosessin virtaviivaistaminen on myös tarpeen. Arviointiprosessin parempi ymmärtäminen voisi auttaa menetelmien kehittämistä. Herkkyys mahdollisille ongelmille on riippuvainen arvioijien vaikeuksista tiettyjen käyttöliittymien käytössä, samanlaisten ongelmien todennäköisyydestä muilla arvioijilla sekä kyvystä yleistää nämä henkilökohtaiset ongelmat (Mack & Montaniz 1994). Tähän liittyen Desurvire (1994) on ehdottanut suunnaksi arvioijien itsetuntemuksen ja sen hyödyntämisen kehittämistä.

Toisena suuntana on tulosten oikeellisuuden ennustaminen ja vakavuusarvioiden suhde todellisen maailman ongelmiin. Selvitystä kaipaisi myös se, mitä tulokset todella ennustavat ja kuinka vakavuusarviot suhtautuvat vaikutukseen markkinoilla. Näiden asioiden selvittäminen auttaa arvioinnilla saavutettavan kustannushyödyn selvittämistä. Ne saattavat myös auttaa valitsemaan parhaan arviointimenetelmän tiettyyn tilanteeseen.

Kolmanneksi tarvittaisiin kehitystä tulosten analysointiin ja tehokkaaseen käyttöön laajasti. Käyttöliittymän suunnittelun ja arvioinnin tulisi olla tiukasti sidottuna toisiinsa. Monissa tapauksissa aikainen käyttöliittymän suunnitteluvaihe sisältää monia läpikäynneistä tuttuja piirteitä. Tämä suhde pitäisi ymmärtää ja sitä pitäisi tukea. Mack (Mack & Nielsen 1994) mainitsee yhtenä mahdollisena tukitoimena käyttöohjeen kirjoittamisen ennen koodia.

Neljäs tutkimuskohde on työkalujen kehittäminen arviointitietojen keräämiseen, järjestämiseen ja uudelleenkäytön avustamiseen. Tarkoituksena on saada arvioinneista kertyvä tieto avuksi myös muiden kuin arvioitavan järjestelmän kehittämässä.

3 Heuristinen arviointi

Useimmat käytettävyyssuunnittelun menetelmät edistäisivät huomattavasti käyttöliittymän käytettävyyttä, jos niitä vain käytettäisiin. Nielsenin (1994a) mukaan järjestelmien kehittäjät pitävät niitä kuitenkin pelottavana, liian kalliina, vaikeina ja aikaavievinä. Tämän vuoksi hän on kehittänyt halvan käytettävyyden ajatuksen, jonka yksi tärkeä menetelmä on heuristinen arviointi. Se on Nielsenin (1994b) mukaan helppo, nopea ja halpa käytettävyyden arviointimenetelmä.

Heuristisen arvioinnin suorittaa pieni joukko arvioijia vertaamalla käyttöliittymää tunnettuihin käytettävyyshauristiikkoihin. Taulukossa 2 on esitellyt alkuperäiset heuristiikat (Molich & Nielsen 1990) ja taulukossa 3 Nielsenin (1994c) 249 käytettävyyssongelman analyysistä johdetut uudet heuristiikat.

Taulukko 2: Alkuperäiset heuristiikat

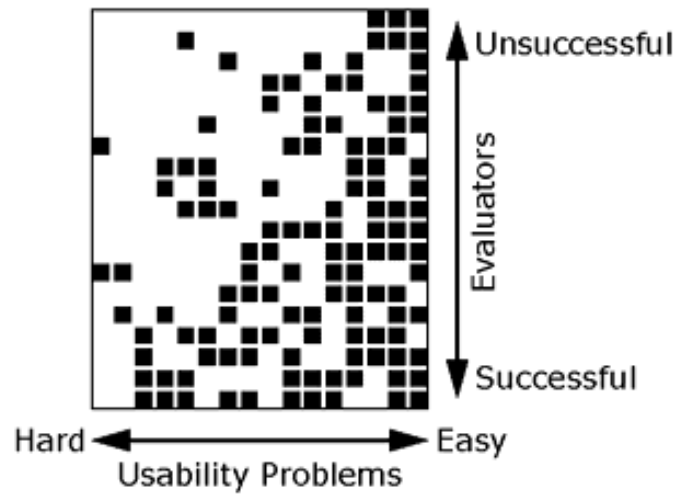
Yksinkertainen ja luonnollinen dialogi
Puhu käyttäjän kieltä
Minimoi käyttäjän muistikuorma
Yhdenmukaisuus
Palaute
Selvästi merkityt poistumistavat
Tarkat ja rakentavat virheilmoitukset
Virheiden estäminen
Avusteet ja dokumentaatio

Taulukko 3: Uudet heuristiikat

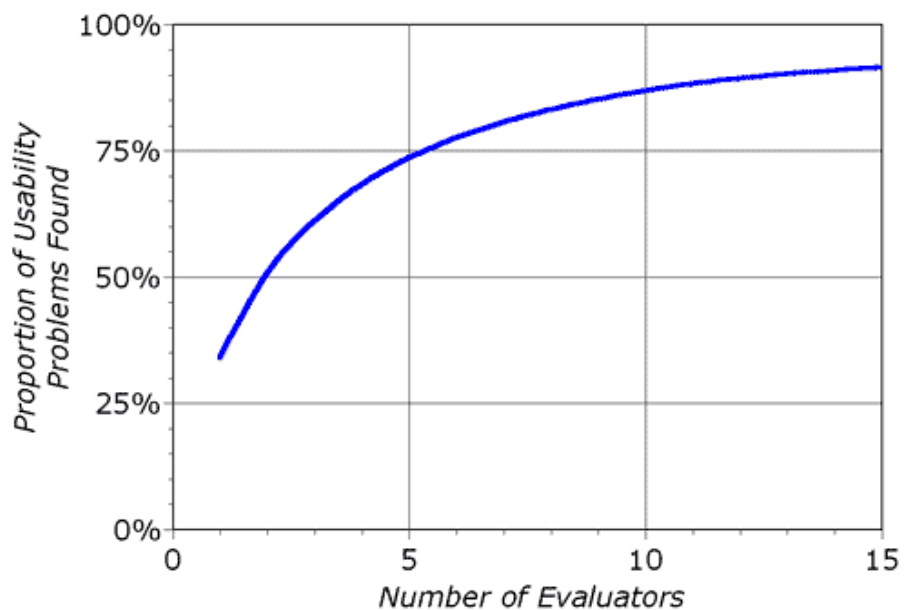
| |
|---|
| Tieto järjestelmän tilasta |
| Yhtenäisyys järjestelmän ja todellisen maailman välillä |
| Käyttäjän kontrolli ja vapaus |
| Yhdenmukaisuus ja standardit |
| Virheiden estäminen |
| Tunnistus, ei muistaminen |
| Käytön joustavuus ja tehokkuus |
| Esteettinen ja minimalistinen ulkonäkö |
| Käyttäjien auttaminen virheiden tunnistamiseen, diagnosointiin ja toipumiseen niistä |
| Avusteet ja dokumentaatio |

3.1 Arvioijat

Periaatteessa yksittäinenkin arvioija voi suorittaa heuristisen arvioinnin, mutta yksi arvioija ei pysty millään löytämään käyttöliittymän kaikkia ongelmakohtia. Kuudesta projektista laskettu keskiarvo (Molich & Nielsen 1990; Nielsen & Molich 1990; Nielsen 1992) yhdellä arvioijalla löytyneistä ongelmista oli 35%. Erilaisilla ihmisillä on kuitenkin tapana löytää erilaisia ongelmia (Nielsen 1992), ja siten menetelmän tehokkuutta on mahdollista parantaa lisäämällä arvioijien määrää. Kuvan 1 jokainen rivi vastaa yhtä arvioijaa ja jokainen sarake puolestaan yhtä käytettävyysongelmaa. Arvioijien määrän tulisi olla ainakin kolme, mahdollisesti viisi (Nielsen 1992). Lopullinen määrä voidaan selvittää kustannushyötyanalyysillä, jossa selvitetään arvioinnista aiheutuvat kustannukset ja vastaavasti siitä saatavan hyödyn rahallinen arvo.



Kuva 1: Eri arvioijien löytämät käytettävyysongelmat (Nielsen 1992)



Kuva 2: Löydetyt virheet suhteessa arvioijamäärään (Nielsen 1992)

Heuristisen arvioinnin menestyksellisyys riippuu suuresti arvioijista. Samanlaisen taustan omaavat henkilöt voivat olla aivan eri tasoisia arvioijia (Nielsen & Molich 1990). Asiantuntijuus käytettävydessä ja sovellusalu-

eella parantaa kuitenkin arvioijan tuloksia suhteessa asiaa tuntemattomiin (Nielsen 1992). Vaikkakin heuristisen arvioinnin voivat suorittaa jopa loppukäyttäjät, on kuitenkin suositeltavaa, että arvioijat ovat aina käytettävyyksiantuntijoita.

3.2 Arvioinnin kulku

Tässä luvussa tarkastellaan heuristisen arvioinnin suoritusta. Ennen varsinaisen arvioinnin alkua arvioijia tarvittaessa perehdytetään sovellusalueeseen ja järjestelmän käyttöön, jotta he osaisivat käyttää järjestelmää itsenäisesti.

3.2.1 Arviointi

Ensiksi kukin arvioija käy käyttöliittymän läpi yksinään useampaan kertaan tarkastaen ja verraten sen osia heuristiikkoihin (taulukot 2 ja 3). Arviointi ei kuitenkaan rajoitu pelkästään tähän, vaan arvioija voi esittää myös omia näkemyksiään käyttöliittymän parantamiseksi. Arvioijat saavat itse päättää, kuinka he käyttöliittymän arvioivat. Suositeltavaa kuitenkin on käydä se läpi ainakin kahdesti. Ensiksi keskittymällä vuorovaikutteisuuteen ja yleisesti järjestelmään. Toisella kerralla keskitytään jokaiseen käyttöliittymäelementtiin erikseen, kun tiedetään sen osa kokonaisuudessa. Arvioijille voidaan myös antaa suoritettavaksi tarkat tehtävät.

Vasta kun kaikki arvioijat ovat saaneet arviointinsa valmiiksi, on aika keskustella ja kerätä havainnot yhteen. Tämä menettely varmistaa itsenäiset ja puolueettomat arviot. Arvioinnin tulokset voidaan joko kirjata paperille arvioijittain tai läpikäymällä käyttöliittymää ja kommentoimalla sitä tarkkailijalle, joka kirjaa kommentit. Kirjallisista raporteista jää muodollinen jälki arvioinnista, mutta ne vaativat ylimääräistä panostusta arvioijil-

ta. Lisäksi arviointitilaisuuden vetäjän pitää lukea raportit ja koota niissä esitetyt ongelmat yhteen. Tarkkailijan käyttö lisää arvioinnin kuormitusta, mutta vähentää arvioijien työmäärää. Arvioinnin tulokset ovat saatavissa nopeasti, sillä tarkkailijan tarvitsee käsitellä vain itse kirjoittamiaan muis-tilappuja. Tarkkailija voi myös toimia avustajana ongelmatilanteissa, kuten jos käytettävissä on epävakaa prototyyppi, arvioijat eivät tunne sovel-lusaluetta tai tarvitsevat lisäselvityksiä jostain käyttöliittymän piirteestä. Tarkkailijan ei välttämättä tarvitse tietää mitään käyttöliittymäarvioinnis-ta.

Arvioijien tulee tarkasti selvittää ongelmaksi merkitsemänsä kohta ver-raten sitä heuristiikkoihin. Heidän tulisi myös merkitä jokainen ongelma erikseen. Tämä auttaa selvittämään ongelmien perimmäiset syyt ja estä-mään niiden toistamista jatkossa.

Käytettäessä sovellusaluetta tuntemattomia arvioijia, pitää heitä tarvittaes-sa auttaa selviytymään eteenpäin sovellusalueriippuvaisissa kohdissa. Tä-mä siksi, ettei menetettäisi kallisarvoista arviointiaikaa. Arvioijat keskitty-vät käytettävyyteen, eikä heidän ole tarkoitus opetella sovellusalueen tun-tijoiksi. Apua ei tulisi kuitenkaan antaa, ennen kuin on selvää, että arvioi-ja on vaikeuksissa ja hän on kirjannut kyseessä olevan käytettävyysongel-man.

Arviointi saa kestää yhden arvioijan kohdalla maksimissaan noin kaksi tuntia. Pitemmät arvioinnit voivat olla tarpeen monimutkaisissa järjestel-missä, mutta tällöin on suositeltavaa pilkkoa arvioinnit osiksi käyttöliitty-män osien mukaan ja pitää taukoja osien välillä.

3.2.2 Jälkipalaveri

Heuristisessa arvioinnissa ei ole järjestelmällistä tapaa tarjota korjauksia käytettävyysoongelmiin tai arvioida uudelleensuunnitteluehdotusten kel-
poisuutta. Mutta koska tavoitteena on selittää ongelmat suhteessa heu-
ristiikkoihin, niin jo tietyn heuristiikan rikkominen aiheuttaa jonkinlais-
ta tajua siitä, kuinka käyttöliittymää voitaisiin tältä osin parantaa. Useat
käytettävyysongelmat ovat sitä paitsi hyvin yksinkertaisia korjata, kun-
han ne on ensin löydetty. Suunnitteluohjeiden tuottamiseen voidaan kyl-
lä käyttää jälkipalaveria viimeisen arvioinnin jälkeen. Palaverin osanotta-
jat koostuisivat tällöin arvioijista, mahdollisesti käytetystä tarkkailijasta ja
käyttöliittymän suunnittelijoista. Palaveri olisi luonteeltaan aivoriihityyp-
pinen ja keskittyisi suurimpien ongelmien korjausehdotuksiin ja yleisesti
käyttöliittymän ongelmakohtiin. Samoin voidaan tuoda esille käyttöliit-
tymän hyviä puolia, heuristisessa arvioinnissa kun ei tätä puolta muuten
käsitellä.

3.3 Käytettävyysongelmien vakavuus

Vakavuusasteiden määrittämisä käytetään päätettäessä resurssien jakami-
sesta ongelmien korjaamiseksi. Vakavimpien ongelmien korjaamiseen kan-
nattaa panostaa enemmän kuin vain kosmeettista haittaa aiheuttaviin. Va-
kavuusarvioiden avulla saadaan myös eräänlainen yleiskatsaus siitä, tar-
vitaanko käytettävyyden parantamistoimenpiteitä lainkaan.

Vakavuusarviot pohjautuvat kolmelle tekijälle:

- Ongelman *taajuus*: Kuinka usein ongelma esiintyy?
- Ongelman *vaikutus*: Kuinka vaikea ongelmasta on päästä ohi?

- Ongelman *pysyvyys*: Onko ongelma kertaluonteinen vai jatkuva?

Ongelman markkinavaikutukset tulisi myös ottaa huomioon. Joillakin ongelmilla voi olla suuri negatiivinen vaikutus tuotteeseen markkinoilla, vaikka ongelma olisi objektiivisesti tarkasteltuna kuinka vähäpätöinen tahansa.

Arvioinnin aikana arvioijilta ei saa kunnollisia arvioita ongelmien vakavuudesta, sillä he ovat keskittyneitä etsimään uusia ongelmia. Parempi tapa onkin kerätä vakavuusarviot arvioinnin jälkeen, esimerkiksi kyselykaavakkeella. Kaavaketta laadittaessa on muistettava, että yksittäinen arvioija on löytänyt vain pienen osan ongelmista. Tämän vuoksi kustakin ongelmasta on oltava riittävän tarkka kuvaus ja tarvittaessa kuvaruutukaappaukset. Esimerkki arviointiasteikosta on taulukossa 4.

Taulukko 4: Esimerkki vakavuusarvion asteikosta

0. Ei käytettävyysongelma
 1. Kosmeettinen ongelma; korjataan, jos aikaa jää
 2. Pieni käytettävyysongelma; alhainen prioriteetti
 3. Suuri käytettävyysongelma; korkea prioriteetti
 4. Käytettävyysskatastrofi; korjattava heti, ennen tuotteen julkistamista
-

Eri arvioijien arviot ongelmien vakavuuksista vaihtelevat. Yhdeltä arvioijalta saadut vakavuusarviot voivat olla epäluotettavia, joten niiden pohjalta ei kannata ruveta suuria panostuksia tekemään. Monen arvioijan arvioista laskettu keskiarvo tarjoaa huomattavasti luotettavamman perustan. Tässäkin pätee sääntö: mitä enemmän arvioijia, sitä luotettavammat

vakavuusarviot saadaan.

3.4 Heuristisella arvioinnilla löydettäviä käytettävyysongelmia

Suuret ongelmat ovat helpompia löytää kuin pienet. Heuristisella arvioinnilla arvioija löysi kuudessa arviointitapauksessa annetut suuret käytettävyysongelmat 42 % tapauksista, kun vastaava luku pienille ongelmille oli 32 % (Nielsen 1992). Samassa tutkimuksessa heuristisella arvioinnilla löydettiin 59 suurta ja 152 pientä käytettävyysongelmaa. Pienet ongelmat muodostivat siis pääosan ongelmaraportissa. Vaikkakin suuret ongelmat ovat tärkeimpiä löytää ja poistaa, ei pieniäkään pidä aliarvioida.

Käytettävyysongelmat voidaan paikallistaa neljällä eri tavalla (taulukko 5). Nielsenin (1992) mukaan arvioijat löytävät kaikkiin neljään kategoriaan sijoittuvat ongelmat yhtä hyvin. Käyttöliittymän toteutuksen aste vaikuttaa suuresti ongelmien löydettävyyteen. Puuttuvien kategoriaan kuuluvat ongelmat oli hieman muita helpompi löytää valmiista järjestelmästä, mutta paljon vaikeampi löytää paperilla. Tämän vuoksi paperiprototyyppiä arvioitaessa tulisikin kiinnittää erityistä huomiota järjestelmästä mahdollisesti puuttuviin piirteisiin.

Taulukko 5: Käytettävyysongelmiä paikallistaminen

1. Sijainti yhdessä paikassa
2. Sijainti kahdessa tai useammassa paikassa
3. Ongelma koko rakenteessa
4. Puuttuu käyttöliittymästä

Vaikka heuristisella arvioinnilla löydetäänkin ongelmia, joita ei loppukäyttäjillä testaamisessa löydetä, se ei löydä kaikkia ongelmia, joita testaamisella löydettäisiin. Varsinkin erittäin sovellusaluekohtaisissa järjestelmissä arvioijat voivat ylenkatsoa joitain ongelmia oman tietämättömyytensä vuoksi. Parhaan tuloksen ongelmien löytämiseksi saa käyttämällä sekä heuristista arviointia että testaamista loppukäyttäjillä. Yleensä tämä tehdään iteroiden; ensin heuristinen arviointi ja sillä löydettyjen ongelmien korjaus ja tämän jälkeen testaus. Näin varmistetaan se, ettei testaa- jien kiusaksi tule jo tiedossa olevia ongelmia. Kummassakaan vaiheessa ei tehdä turhaa työtä, sillä heuristisen arvioinnin ja käytettävyydestestauksen on todettu löytävän eri tyyppisiä ongelmia (Desurvire et al. 1992; Jeffries et al. 1991; Karat et al. 1992).

3.5 Yhteenveto

Heuristinen arviointi on helppokäyttöinen käytettävyyden arviointimenetelmä. Arvioinnissa käyttöliittymä tulisi käydä läpi kahdesti; ensin tuntu- ma ja sen jälkeen yksittäiset elementit. Sopiva määrä arvioijia on 3 - 5, ja he arvioivat käyttöliittymän yksin, verraten sitä heuristiikkoihin (taulukot 2 ja 3). Arvioinnin jälkeen on mahdollista pitää jälkipalaveri, jossa arvioi- jat ja käyttöliittymän suunnittelijat tapaavat ja keskustelevat ongelmista ja parannusehdotuksista. Arvioinnin jälkeen arvioijilta voi myös kerätä va- kavuusarvioita ongelmista.

4 Kognitiivinen läpikäynti

Tutkimalla oppiminen (learning by exploration) on oppimisstrategia, jossa käyttäjä opettelee järjestelmän käyttöä normaalien työtehtäviensä lomassa. Tällöin järjestelmän uusia piirteitä opetellaan siinä vaiheessa, kun niiden opetteleminen tulee työnteon kannalta aiheelliseksi.

Kognitiivinen läpikäynti (Lewis et al. 1990; Polson et al. 1992) on käytettävyyden arviointimenetelmä, jossa järjestelmän käytettävyyttä arvioidaan tutkimalla oppimisen näkökulmasta. Menetelmän perustana on huomio siitä, että useat käyttäjät opettelevat järjestelmien käyttöä tutkimalla sen sijaan, että harjoittelisivat sen käyttöä järjestelmällisesti oppaiden mukaan (Carrol & Rosson 1987; Fischer 1991).

Keskittyminen ainoastaan opittavuuteen perustuu tutkimuksiin, joiden mukaan tutkimalla oppimisen helpottaminen tukee *taidonhankintaa* (skill acquisition) (esim. Anderson 1987). Muut käytettävyyden osa-alueet riippuvat taas opittavuudesta. Jos esimerkiksi järjestelmän toiminnallisuus sopii huonosti käyttäjän tarpeisiin ja käyttäjä joutuu tekemään ihmeellisiä temppuja saadakseen työnsä tehtyä, niin järjestelmän käyttö on hankala oppia.

4.1 Arvioijat

Kognitiivinen läpikäynti voidaan suorittaa sekä yksilö- että ryhmäarviointina. Ryhmäarvioinnissa käyttöliittymän suunnittelija esittää suunnitelmansa työtovereilleen, ja käyttää heiltä saamaansa palautetta suunnitelman muuttamiseen tai vahvistamiseen. Ryhmäarviointi suoritetaan tavallisesti

jonkin etapin, esimerkiksi prototyypin valmistumisen, saavuttamisen jälkeen. Ryhmä voi koostua muista käyttöliittymäsuunnittelijoista, ohjelmistosuunnittelijoista ja vaikkapa organisaation muiden osien edustajista, kuten markkinointi-, koulutus- ja dokumentointihenkilöstöstä. Yksi osallistujista toimii sihteerinä, joka kirjaa läpikäynnin aikana esiin nousevat asiat ja kommentit. Toinen osallistuja toimii keskustelun johtajana huolehtien tilaisuuden etenemisestä.

Yksilöarviointia voidaan käyttää oman työn arvioimiseen. Koska läpikäynti perustuu selkeään malliin tutkimalla oppimisesta, on suunnittelijalla tilaisuus sisäistää tietoa malliin liittyvästä prosessista. Tämä tieto vaikuttaa tuleviin päätöksiin hänen kehittäessään järjestelmää eteenpäin.

4.2 Arvioinnin kulku

Kognitiivisessa läpikäynnissä on kaksi vaihetta; valmistelu ja analysointi. Valmisteluvaiheessa *arvioijat* (analysts) sopivat tehtävät, niiden suorittamiseen tarvittavat *toiminnot* (actions), käyttäjäkunnan ja arvioitavan käyttöliittymän. Analysointivaiheessa arvioijat käyvät läpi jokaisen tehtävän kaikki toiminnot. Molemmat vaiheet riippuvat suuresti siitä, missä vaiheessa ohjelmistotuotantoprosessia menetelmää käytetään. Läpikäynti voidaan tehdä paperisimulaatiolla, minimaalisella prototyypillä tai täysin toimivalla prototyypillä.

4.2.1 Läpikäynnin valmistelu

Ennen läpikäynnin aloittamista on arvioijien sovittava neljästä asiasta.

1. Käyttäjäkunta
2. Arvioitavat tehtävät
3. Tehtävien oikea suoritustapa
4. Käyttöliittymän määrittely

Ensimmäinen asia näyttää helpolta. Ei kannata kuitenkaan sortua liian ylimalkaiseen kuvaukseen. Läpikäynti tuottaa parempia tuloksia, jos käyttäjäkunta kuvataan tarkasti niiltä kokemuksilta ja tiedoilta, jotka saattavat vaikuttaa työskentelyyn käyttöliittymän parissa.

Arvioitavat tehtävät vaihtelevat. Toiminnoiltaan vähäisen järjestelmän voi arvioida kaikilta osiltaan ennen markkinoille päästöä, kun taas joskus voi olla tarpeen arvioida jo käytössä olevasta järjestelmästä jokin tietty tehtävä. Yleensä käytetään pientä edustavaa joukkoa tehtäviä. Tehtävien valinta tulisi tehdä markkinatutkimuksien, tarveanalyysien, käsitetestauksien ja vaatimusmäärittelyn pohjalta. Osan tehtävistä tulisi keskittyä järjestelmän perustoimintoihin ja osan niiden yhdistelyyn. Toimintaympäristön pitäisi olla mahdollisimman realistinen apuohjelmiseen ja tietokantoineen.

Jokaisesta tehtävästä pitää kuvata, kuinka käyttäjän odotetaan tarkastelevan sitä ennen käyttöliittymän oppimista. Samoin pitää kuvata toiminnot, joilla tehtävä saadaan suoritettua. Nämä toiminnot voidaan kuvata yksityiskohtaisesti ('paina ENTER') tai luontevasti yhteenkuuluvissa sarjoissa

(‘kirjautu järjestelmään’). Kuvauksen muoto riippuu odotettujen käyttäjien kokemuksesta.

Käyttöliittymästä pitää kuvata kaikkia toimintoja edeltävät kehoitteet ja käyttöliittymän reagointi näihin toimintoihin. Jos käyttöliittymä on jo toteutettu, on siinä kaikki tarvittava tieto. Pelkää määrittelyä arvioitaessa kuvausten tarkkuus riippuu odotettujen käyttäjien kokemuksesta.

4.2.2 Läpikäynti

Läpikäynti koostuu ratkaisuun tarvittavien toimintojen tutkimisesta ja yrityksestä keksiä selitys sille, miksi odotettu käyttäjä valitsisi juuri tämän toiminnon. Selitykset perustuvat oletuksille käyttäjän taustatiedoista ja tavoitteista, sekä oikeaan tavoitteeseen johtavan ongelmanratkaisuprosessin ymmärtämisestä. Ongelmanratkaisuprosessi on kuvattu tarkasti Polsonin ja Lewisin CE+ -teoriassa tutkivasta oppimisesta (Polson & Lewis 1990). Teorian mukaan käyttäjät aloittavat ratkaistavan tehtävän karkealla kuvauksella tutkien käyttöliittymää ja valiten toimintoja, jotka heidän mielestään ratkaisevat tehtävän. Tämän jälkeen he tarkkailevat käyttöliittymän reaktioita nähdäkseen, saivatko toiminnot aikaan haluttuja vaikutuksia ja siirtyvät seuraavaan toimintoon. Teoria huomioi myös muutamia heuristiikkoja, joita käyttäjät käyttävät. Toimintojen nimien perusteella (label-following) tapahtuva järjestelmän kokeileva käyttö on hyvin suosittu. Siinä käyttäjä valitsee toiminnon, jonka nimi sopii tehtäväkuvaukseen (Engelbeck 1986). Esimerkiksi toiminto ‘tulosta asiakirja’ johtaa käyttäjän valitsemaan toiminnon, jonka nimenä on joko ‘tulosta’ tai ‘asiakirja’, tai niihin liittyvä kuvake.

Käyttöliittymän kriittiset piirteet ovat käyttäjien mieltämät yhteydet tehtäväkuvauksen ja oikean toiminnon välillä sekä edistymisestä kertova palaute. Läpikäynnin edetessä arvioijat käyttävät teoriaa perustellessaan, miksi käyttäjä valitsisi juuri oikean vaihtoehdon jokaisella askeleella. Seuraavaksi tarkastellaan neljää arvioinnissa avustavaa kysymystä, sekä mahdollisia ratkaisuja niiden avulla paljastuneisiin ongelmiin.

1. *Yrittääkö käyttäjä saada oikeaa asiaa aikaan?*

Jos esimerkiksi asiakirjan tulostaminen edellyttää tulostimen valintaa, niin tietääkö käyttäjä, että hänen pitäisi saada tulostin valituksi. Ongelman ratkaisemiseksi on ainakin kolme lähestymistapaa. Ensimmäinen vaihtoehto on poistaa toiminto; antaa se järjestelmän tehtäväksi tai yhdistää se johonkin toiseen toimintoon. Toinen vaihtoehto on tarjota kehote, joka kertoo käyttäjälle, mitä tämän pitäisi tehdä. Kolmas vaihtoehto on muuttaa tehtävän suorittamista siten, että käyttäjä ymmärtää toiminnon tarpeen paremmin.

2. *Huomaako käyttäjä, että oikea vaihtoehto on saatavilla?*

Vaihtoehdon ollessa näkyvillä valikossa, tämä ei ole ongelma. Eri asia on, jos käyttäjän pitää kolmoisklikata tulostimen kuvaketta saadaakseen toiminnon aikaan. Yleinen ratkaisu näkyvyyden parantamiseen on kehotteen tai valikon lisääminen.

3. *Yhdistääkö käyttäjä oikean toiminnon oikean tuloksen aikaansaamiseen?*

Nimillä on suuri vaikutus. Suunnittelijan pitää tuntea käyttäjä ja sovellusalueen terminologia. Näin nimet ja kuvaukset saadaan samoik-

si, mitä käyttäjä itse käyttää tehtäviään kuvatessaan. Voi olla myös tarpeen vaihtaa muiden kontrollien nimet oikean vaihtoehdon esilletuomiseksi.

4. Antaako järjestelmä palautetta oikeasta toiminnosta?

Useimmissa tilanteissa mikä tahansa palaute on tyhjää parempi, ja palaute joka kertoo *mitä* on tapahtunut, on puolestaan parempi kuin palaute joka kertoo, että *jotain* on tapahtunut.

Muita mahdollisia ongelmia ovat aikarajat (time-outs), fyysisesti vaikeat tehtävät (physically difficult actions) ja lopetusmerkkien unohtaminen (dropped terminator actions). Aikarajoja on esimerkiksi puhelinpalveluissa. Niiden tulisi olla riittäviä hitaillekin käyttäjille. Fyysisesti vaikeita tehtäviä ovat esimerkiksi monen näppäimen painaminen yhtäaikaan tai pienen kohteen valitseminen hiirellä. Lopetusmerkkejä on esimerkiksi ohjelmoinnissa rivin tai suuremman kokonaisuuden lopetuksessa.

4.2.3 Tiedonkeruu arvioinnin aikana

Tiedonkeruuseen on monia menetelmiä: videointi (Rowley & Rhoades 1992), elektroninen nauhoitus (Rieman et al. 1991), fläppitaulu tai kalvot (Wharton et al. 1991) sekä paperilomakkeet (Wharton 1992; Wharton et al. 1992; Lewis et al. 1991). Ryhmäarvioinneissa on suositeltavaa käyttää keruumenetelmää, jonka kaikki osallistujat näkevät, kuten fläppitaulua tai kalvoja. Tarvittaessa voidaan koko läpikäyntitilaisuus videoida, jolloin päästään jälkeinpäin varmistamaan ja jäljittämään kommentteja ja päätöksiä.

Läpikäynnin aikana pitää kerätä monenlaista tietoa. Tärkeimpiä näistä ovat käyttäjältä vaadittavat tiedot, oletukset käyttäjäkunnasta, huomautukset sivuseikoista, muutosehdotukset ja kertomus siitä, miksi käyttäjä tekisi päätöksensä. Tämän vuoksi on suositeltavaa käyttää kolmea menetelmää; yhtä kertomuksen pääkohtien kirjaamiseen, toista käyttäjätietojen keräämiseen ja kolmatta muille tiedoille ja muutosehdotuksille.

Käyttäjätietojen keräämisessä tulisi keskittyä kahteen seikkaan; vaadittaviin pohjatietoihin ja suorituksessa oppimiseen. Sivuseikat ja muutosehdotukset kannattaa kirjata niin laajasti, että niiden tarkempi käsittely jälkeenpäin on mahdollista. Kertomuksista kirjataan avainkohdat seuraavien esimerkkien mukaisesti.

Kertomus 1: Edistynyt Macintosh-käyttäjä aloittaa tehtävän kaksoikkamalla ohjelman kuvaketta käynnistääkseen sen.

Puoltavia seikkoja:

- Käyttäjä yrittää käynnistää ohjelman, sillä hän tietää, että ohjelma on käynnistettävä jotta sitä voisi käyttää.
- Käyttäjä tietää kokemuksesta, että kaksoisklikkaus on mahdollista.
- Käyttäjä tietää kokemuksesta, että kaksoisklikkaus on oikea tapa käynnistää ohjelma.
- Muutokset näytössä ja valikkorivillä kertovat, että ohjelma käynnistyy.

Ensimmäiset kolme seikkaa eivät päde käyttäjille, joilla ei ole kokemusta tietokoneista. Toinen ja kolmas seikka puolestaan pätevät vain Mac-

käyttäjille.

Kertomus 2: Pankin asiakas käyttää puhelimitse pankin järjestelmää siirtääkseen rahaa tililtä toiselle. Järjestelmä kehoittaa käyttäjää antamaan tunnusluvun ja käyttäjä näppäilee sen.

Puoltavia seikkoja:

- Käyttäjä yrittää näppäillä numeron, sillä järjestelmä pyysi sitä.
- Käyttäjä käyttää puhelimen näppäimiä, sillä ne ovat ainoat näkyvillä olevat tarkoitukseen sopivat näppäimet.
- Käyttäjä tietää tunnuslukunsa, sillä hän on opetellut sen ulkoa saatuaan sen pankilta.
- Käyttäjä uskoo asioiden menneen oikein, kun järjestelmä alkaa toistaa toimintovalikkoa.

Arvioijat käyvät toiminnot läpi seuraavista neljästä näkökulmasta:

1. Käyttäjä tietää, mitä pitää saada aikaan:

- koska se on osa alkuperäistä tehtävää
- koska hänellä on kokemusta järjestelmän käytöstä
- koska järjestelmä kehottaa häntä tekemään niin

2. Käyttäjä tietää, että toiminto on saatavilla:

- kokemuksesta
- nähdessään jonkin nappulan tms.

- nähdessään tietyn valikon kohdan
3. Käyttäjä tietää, että toiminto soveltuu tavoitteen saavuttamiseen:
- kokemuksesta
 - koska käyttöliittymä tarjoaa kehotteen tai nimiön, joka yhdistää toiminnon suoritettavaan tehtävään
 - koska kaikki muut vaihtoehdot vaikuttavat vääriltä
4. Käyttäjä tietää, että asiat sujuvat oikein toiminnon jälkeen:
- kokemuksesta
 - tunnistamalla yhteyden järjestelmän reaktion ja suoritettavan tehtävän välillä

Näiden seikkojen perusteella voidaan korostaa tietämystä siitä, kuinka käyttäjät kuvailevat tehtävät. Järjestelmän käyttäessä samaa terminologiaa ja grafiikkaa kuin käyttäjä, käyttäjä todennäköisesti valitsee oikeat toiminnot. Palaute tulisi myös antaa käyttäjän normaalia sanavarastoa käyttäen, sillä oudot termit vaikeuttavat järjestelmän käyttöä ilman lisätietoja.

4.3 Yhteenveto

Kognitiivinen läpikäynti korostaa käyttöliittymän opittavuutta. Menetelmä jakautuu kahteen vaiheeseen: valmisteluun ja varsinaiseen läpikäyntiin. Arvioijat yrittävät jokaisen yksittäisen toiminnon kohdalla keksiä selityksen sille, miksi käyttäjä valitsi juuri oikean vaihtoehdon käyttöliittymästä.

Kognitiivinen läpikäynti on arvokas työkalu käyttöliittymien parantamiseen. Se on joustava ja sopii mihin tahansa ohjelmistotuotantoprosessiin. Menetelmää voi käyttää aikaisessa vaiheessa prosessia ja se tarjoaa ongelmiin ratkaisut kuvailemalla niiden syitä. Käyttäjillä käyttöliittymää voi testata vasta kun se on toteutettu ja testaus ei paljasta niin hyvin ongelmien syitä. Parhaan tuloksen saavuttamiseksi arvioijilla tulisi olla taustatietoja kognitiivisesta psykologiasta (Jeffries et al. 1991; Wharton et al. 1992). Kognitiivista läpikäyntiä ei ole suunniteltu löytämään kaikkia ongelmia käyttöliittymästä. Menetelmän kehittäjät suosittelevat sitä käytettäväksi muiden käyttöliittymän kehittämiseen tarkoitettujen menetelmien kanssa.

5 Muodollinen arviointimenetelmä

Muodollinen käytettävyyden arviointimenetelmä (formal usability inspection method) (Kahn & Prail 1994) on kehitetty auttamaan järjestelmän kehittäjiä löytämään käytettävyysongelmia. Menetelmässä järjestelmän kehittäjät toimivat arvioijina. Menetelmässä määritellään muodollinen prosessi ongelmien löytämiseksi ja kuvaamiseksi sekä selkeät roolit osallistujille. Lisäksi määritellään kuuden askeleen logistinen kehys, jonka avulla menetelmä nivoutuu aikataulutettavaksi tapahtumaksi käytettävyyssuunnittelun elinkaarelle. Menetelmässä on myös kiinnitetty huomiota ajankäytön tehokkuuteen.

5.1 Arvioijat

Arviointi tapahtuu ryhmässä, joka on kooltaan 4-8 henkeä. Ihanteellista olisi saada ryhmään eri alojen asiantuntijoita, kuten järjestelmän kehittäjiä, heidän työtovereitaan, dokumentoijia, käytettävyyssuunnittelijoita ja tukihenkilöitä. Laaja asiantuntemus lisää löydettyjen ongelmien määrää ja lajityyppejä. Kokoa kannattaa kuitenkin rajoittaa, sillä ryhmään lisätty uusi arvioija lisää löydettyjen ongelmien määrää suhteessa koko ajan vähemmän (Nielsen & Landauer 1993). Käytettävyyden alalta kokemusta hankkineet arvioijat löytävät eniten ongelmia, mutta kokemattomiakin kannattaa ottaa mukaan, jotta he saisivat kokemusta.

Ryhmän jäsenillä on tarkasti määritellyt roolit.

Valvoja (moderator) johtaa arviointiprosessia. Hänen tehtävänä on arviointiin liittyvän materiaalin kerääminen ja jakelu, arvioinnin aikataulu-

tus ja tilojen varaaminen sekä seurannan koordinointi. Valvoja ei ole normaalissa työssään tekemisissä arvioitavan kohteen tai sen projektin kanssa.

Tekijä (owner) on arvioitavan kohteen tekijä tai siitä vastaava. Hän ymmärtää kaikki löydetyt ongelmat ja auttaa korjausehdotuksien keksimisessä.

Arvioija (inspector) etsii ongelmat ja raportoi niistä sekä osallistuu korjausehdotuksien keksimiseen. Myös valvoja ja tekijä voivat toimia arvioijina.

Kirjuri (scribe) kirjaa julkisesti kaikki ongelmat ja arvioinnin aikana esiin nousseet asiat. Kirjurin rooli voidaan antaa yhdelle arvioijalle.

5.2 Arvioinnin kulku

Arvioijat käyttävät arvioinnissaan oletetuista käyttäjistä luotuja *profiileja* (profile) sekä heille keksittyjä *tehtäviä* (task scenario). Käyttäjäprofiilissa kuvataan käyttäjän koulutus ja kokemus siten kuin se järjestelmän käytön kannalta on olennaista. Arvioijille annetaan aluksi *arviointipaketti* (inspection packet), jossa on kuvaus arvioitavasta järjestelmästä, käyttäjäprofiilit sekä tehtävät. Mukana saattaa myös olla ohjeita tehtävien suorittamiseen ja vihjeitä virheiden etsintään.

5.2.1 Tehtävien läpikäynti

Järjestelmän kuvaus sisältää yleensä piirroksia näytöstä ja selitystekstejä, mutta se voi olla myös prototyyppi tai kuvakäsikirjoitus (storyboard). Kukin käyttäjäprofiilin kuvaus sisältää vähintään profiilin nimen, koulutuk-

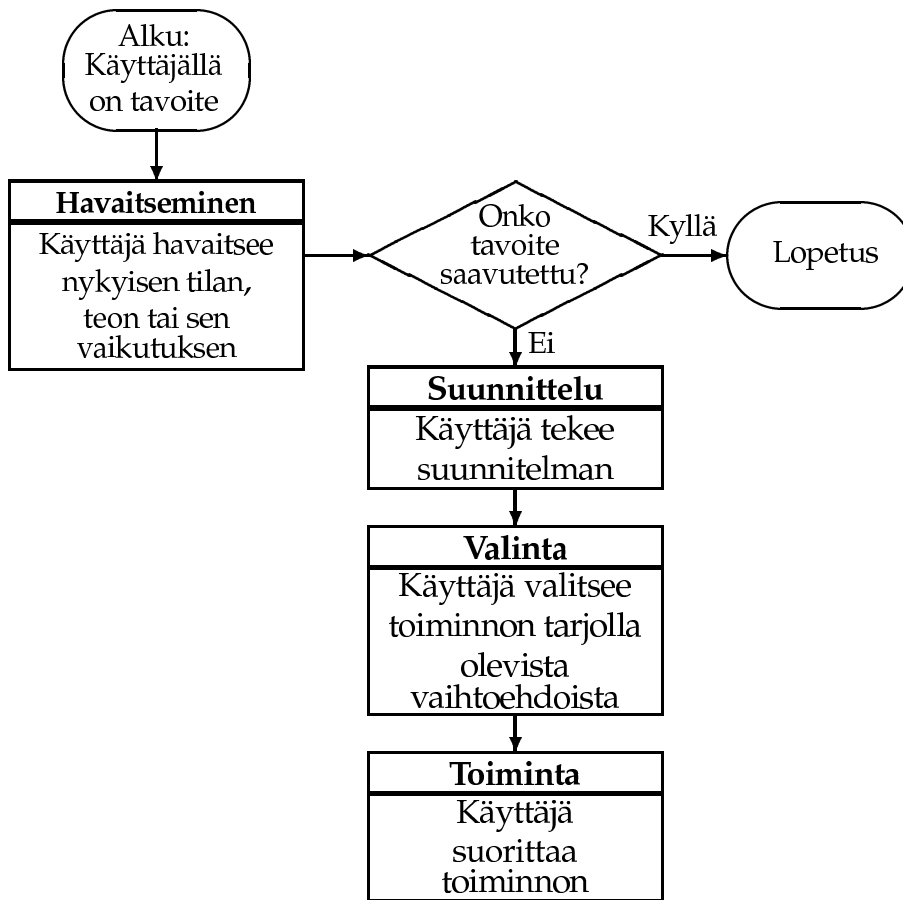
sen ja käyttökokemuksen. Käyttökokemus on kokemusta nimenomaan arvioitavan kaltaisesta järjestelmästä.

Jokainen tehtäväkuvaus sisältää vähintään käyttäjän kielellä kuvatun tavoitteen, järjestelmän ja tehtävän tilan aloitusvaiheessa sekä muut välilliset tilanteet, jotka saattavat vaikuttaa tehtävän suorittamiseen. Kuvaukset eivät sisällä yksityiskohtaisia ohjeita tehtävän suorittamiseen, sillä menetelmän idea on se, että arvioijat joutuvat itse miettimään, kuinka tehtävä suoritetaan.

Tehtävät ja käyttäjäprofiilit auttavat arvioijia ymmärtämään käyttäjän näkökulmaa, sitä minkälaista tietoa heillä on käytettävissä ja minkälaista ei. Samoin ne auttavat huomaamaan, että järjestelmän tulevat käyttäjät eivät ole samanlaisia kuin he itse. Tehtävien tekeminen on juuri sitä, mihin käyttäjät järjestelmää käyttävät. Arvioitaessa järjestelmää abstraktimmalla tasolla, arvioijat katsovat ongelmia helpommin läpi sormien.

Haluttaessa minimoida ajankäyttöä ja siitä aiheutuvia kuluja, voidaan arvioijille antaa vain yksi käyttäjäprofiili ja pieni määrä tehtäviä. Käyttäjäprofiiliksi valitaan se, jonka oletetaan kohtaavan eniten ongelmia. Tehtäviksi valitaan yleisimpiä ja oletettavasti eniten ongelmia aiheuttavia.

Varsinainen arviointi tapahtuu siten, että arvioija valitsee tietyn käyttäjäprofiilin ja käy sillä tehtävät läpi. Jos profiileja ja/tai tehtäviä on useita, prosessia toistetaan, kunnes kaikki yhdistelmät on käyty läpi. Jos arvioija ei saa tehtävää suoritettua tai hämmentyy jossain sen vaiheessa, on hän kohdannut käytettävyysongelman ja kirjaa sen.



Kuva 3: Tehtävnsuoritusmalli (Kahn & Prail 1994)

5.2.2 Ongelmien etsiminen

Mietittäessä mistä kohdasta käyttäjän *tehtäväävirtaa* (task flow) ongelmia kannattaa etsiä, on apuna *tehtävnsuoritusmalli* (kuva 3). Malli muodostuu vuokaaviosta, joka kuvaa abstraktisti ihmisen suoriutumista tehtävästä ja siinä on neljä pääkohtaa; *havaitseminen* (perceiving), *suunnittelu* (planning), *valinta* (selecting) ja *toiminta* (acting) (Kahn & Prail 1994). Mallin on tarkoitus avustaa arvioijia havaitsemaan, että suuri osa tehtävän suorittamisesta tapahtuu käyttäjän päässä edellämainituissa vaiheissa ja etsimään virhemahdollisuuksia näistä vaiheista.

Ensiksi käyttäjä aloittaa tavoitteen toteuttamisen. Havaitsemisvaiheessa hän tarkastelee järjestelmän antamia merkkejä ja toteaa, että tavoitetta ei ole saavutettu. Tämän jälkeen hän suunnittelee yksityiskohtaiset vaiheet tavoitteeseen pääsemiseksi. Suunnitteluvaiheessa usein luodaan tai palautetaan mieleen järjestelmän *mentaalimalli* (mental model), jonka avulla suunnitelma sitten luodaan. Mentaalimalli on käyttäjän itselleen muodostama kuva järjestelmästä. Valintavaiheessa käyttäjä päättää, minkä käytettävissä olevan vaihtoehdon hän valitsee. Valinnan tekemisen jälkeen palataan havaitsemisvaiheeseen, jossa katsotaan saavutettiinkö tavoite. Jokaiseen neljään vaiheeseen liittyy kysymyksiä, jotka auttavat arvioijaa ongelmien löytämisessä.

Havaitseminen

- Näkeekö käyttäjä tarvitsemansa tiedon?
- Voiko käyttäjä päätellä, onko tavoite saavutettu?
- Mitä ongelmia käyttäjällä voi olla päätellessään:
 - järjestelmän tilaa?
 - suoritettiinkö valittu toiminto?
 - oliko valittu toiminto oikea?
 - tapahtuiko virheitä?
 - tarvittavia toimia virheestä toipumiseen?

Suunnittelu

- Mitä suunnitelmaa tai mallia käyttäjä aiempien kokemustensa pohjalta todennäköisimmin käyttää?

- Kuinka käyttäjä kuvaa tavoitteen?
- Minkälaista tukea käyttäjälle on tarjolla oikean suunnitelman tekemiseen?
- Pitääkö käyttäjän tarkastella tavoitetta erikseen omalta ja järjestelmän kannalta?

Valinta

- Mitä ongelmia käyttäjällä voi olla tarjolla olevien toimintojen tunnistamisessa?
- Onko tarjolla virhemahdollisuuksia aiheuttavia samankaltaisia toimintoja?
- Mitä vinkkejä käyttäjälle on tarjolla oikean valinnan tekemiseen?

Toiminta

- Onko käyttäjällä fyysisiä ongelmia suorittaa toimintoa?
- Mitä odottamattomia toimintoja saattaa tapahtua?
- Onko toiminto käyttäjän vai järjestelmän vastuulla?
- Kuinka kauan toiminnon suorittaminen kestää? Onko aika kohtuullinen?

Kaikkia arvioijia tulisi kouluttaa tehtävänsuoritusmallin käyttöön. Yleisimmät tavat käyttää mallia ovat *tarkastuslistatapa* (checklist approach), jossa arvioija tarkastelee mallia ja kysymyksiä jokaisella askeleella, sekä *ennakoiva tapa* (priming approach), jossa arvioija tarkastelee mallia ennen

tehtävien suorittamista ja soveltaa sen osia suorituksen aikana. Tarkastuslistatavalla löytyy enemmän ongelmia (Kahn & Prail 1994), mutta useimmissa tapauksissa se on liian aikaavievä ja puuduttava. Sitä voidaan tosin käyttää rajoitetusti joidenkin tehtävien suorittamisessa.

Kahn ja Prail (1994) ovat sitä mieltä, että edellä mainitut kysymykset menevät liian syvälle ihmisen kognitiiviseen prosessiin ja niitä pitäisi opettaa vain käytettävyyssiantuntijoille sekä muille kiinnostuneille. Tarkastuspaketista ne pitäisi myös jättää pois.

Taulukko 6: Tehtäväperustaiset heuristiikat (Kahn & Prail 1994)

- Minimoi tehtävän suorittamiseen tarvittavat toiminnot
 - Tue luonnollista tehtäväävirtaa
 - Älä pyydä käyttäjää tekemään asioita, jotka kone tekee paremmin
 - Tarjoa oletusarvoja
 - Tarjoa kumoa-, uudelleenteko-, pysähdys- ja jatka-toiminnot
 - Tarjoa oikoteitä
 - Tee työn tuhoaminen vaikeaksi
 - Tee virheiden korjaaminen helpoksi
 - Älä hukkaa käyttäjän aikaa
 - Tarjoa selvästi merkityt poistumistiet
-

Muodollisessa arviointimenettelyssä käytetään heuristisen arvioinnin (luku 2) tapaan apuna heuristiikkoja (taulukot 2 ja 3). Kahn ja Prail ovat jakaneet heuristiikat kahteen luokkaan; *tehtäväperustaisiin* (task-based) (taulukko 6) ja *käyttäjäperustaisiin* (user-based) (taulukko 7) heuristiikkoihin, korostaakseen käyttäjäprofiilien merkitystä arvioinnissa. Tehtäväperustai-

set heuristiikat on tarkoitettu näytöllä navigoimisen ja tehtävävirran arviointiin, eivätkä ne vaadi käyttäjäprofiilien tuntemista. Käyttäjäperustaisia heuristiikkoja käytettäessä tarvitaan tietoa oletetuista käyttäjistä. Heuristiikkalistoja voidaan käyttää samoilla tavoin kuin tehtävänsuoritusmalleja: ennen arviointia tai sen aikana.

Taulukko 7: Käyttäjäperustaiset heuristiikat (Kahn & Prail 1994)

Tarjota mentaalimalli, joka mukautuu aiemmin luotuihin malleihin
Käytä yksinkertaista ja luonnollista dialogia
Puhu käyttäjän kieltä
Ole yhdenmukainen
Käytä intuitiivista asettelua
Erota osat toisistaan
Tee toiminnallisuus ilmeiseksi ja helposti saatavaksi
Tarjota hyvät avusteet
Tarjota eri taitotasoja
Salli käyttäjän mukauttaa
Anna käyttäjän päättää tehtävästä ja käyttöliittymästä
Tarjota palautetta ja minimoi epävarmuus
Minimoi muistikuorma ja mietintätarve
Minimoi tilat
Ota käyttäjän virheet huomioon suunnittelussa
Suojele käyttäjää toteutuksen yksityiskohdilta

5.2.3 Kuusi logistista askelta

Muodollinen arviointimenetelmä muodostuu kuudesta logistisesta askeleesta:

1. Suunnittelu (planning)
2. Aloituskokous (kickoff meeting)
3. Valmistelu (preparation)
4. Arviointi (logging meeting)
5. Ratkaiseminen (rework)
6. Seuranta (follow-up)

Askeleiden tarkoitus on varmistaa, että itse arviointi suoritetaan tehokkaasti ja että se on yhteydessä käytettävyyssuunnittelun kokonaisuuteen.

Valmisteluvaiheessa valvoja ja tekijä varmistavat yhdessä, että osallistujat ovat ajan tasalla ja että he pääsevät käsiksi arvioinnissa tarvittavaan tietoon. He määrittelevät arvioinnin tavoitteet, valitsevat arvioijat ja koostavat tarkastuspaketin (ks. taulukko 8). Yleensä tehtävät ja käyttäjäprofiilit ovat valmiina aiemmin tehdyistä käyttäjäkartoituksista.

Kokemukset (Kahn & Prail 1994) osoittavat, että arvioijilla on taipumus kirjata ongelmat hyvin ympäröivästä. Heitä pitäisikin opastaa ongelmien kuvaamiseen käyttäjäkeskeisesti. Näin he ymmärtävät paremmin käyttäjän kokemat ongelmat ja osaavat jakaa ne muiden arvioijien kanssa. Jaettu ymmärrys on tarpeen ratkaisujen mietintävaiheessa.

Taulukko 8: Tarkastuspaketin sisältö (Kahn & Prail 1994)

Tarkastusohjeet: paketin käyttöohjeet

Järjestelmän kuvaus: prototyyppi, näyttökaappaukset...

Tukevat dokumentit: standardit, vastaavat tuotteet..

Käyttäjäprofiilit: kuvaukset jokaisesta käyttäjätypistä

Tehtävät: riittävä kuvaus kustakin tehtävästä

Tehtävänsuoritusmalli: ks. kuva 3

Heuristiikat: ks. taulukot 2, 3, 6 ja 7

Lomake ongelmien kirjaamiseen

Logistiikka: ajat ja paikat

Aloituskokouksessa arvioijat tapaavat toisensa ensimmäistä kertaa. Valvoja ja tekijä jakavat arvioijille tarkastuspaketin ja kertovat sen sisällöstä. Valvoja kertoo kuinka arvioijien tulisi käyttää käyttäjäprofiileja, tehtäviä, tehtävänsuoritusmallia ja heuristiikkoja ongelmien etsimisessä. Hän myös korostaa, että arvioijille on varattu aikaa arvioinnin suorittamiseen ja heidän tulisi keskittyä arviointiin.

Valmisteluvaiheessa arvioijat työskentelevät yksin. Ensin he tutustuvat tarkastuspaketin sisältöön ja sen jälkeen käyvät jokaisen käyttäjäprofiilin tehtävät läpi. Apuna he käyttävät heuristiikkoja ja tehtävänsuoritusmalleja. Kohdatessaan käytettävyysongelman arvioija kirjaa sen käyttäjäkeskeisestä näkökulmasta. Ongelmista kirjataan tehtävän numero, ongelman sijainti ja kuvaus.

Arviointivaiheessa kokoonnutaan taas yhteen esittelemään ongelmia ja löytämään uusia. Valvoja käy tehtävät läpi käyttäjäprofiileittain kysellen

arvioijilta, mitä käyttäjä mahtaisi seuraavaksi tehdä ja olisiko tällä ongelmia. Arvioijat kuvaavat valmisteluvaiheessa löytämänsä ongelmat sekä kertovat, jos eivät ole saaneet tehtävää suoritettua oikein. Nämä epäonnistumiset lasketaan myös käytettävyysongelmiksi. Arviointivaiheessa löytyy myös ongelmia, joita kukaan arvioijista ei ole valmisteluvaiheessa löytänyt. Valvoja käy tehtävät läpi käyttäjakeskeisesti ja pyytää tarvittaessa arvioijia kuvaamaan ongelmia tarkemmin. Tekijän osana on toimia arvioijana tai vain istua ja ymmärtää esiin nousevat ongelmat. Kirjuri kirjaa ongelmat siten, että kaikki osallistujat näkevät ne. Piirtoheitinkalvo sopii hyvin tähän tarkoitukseen. Tässä vaiheessa saattaa tulla korjausehdotuksia, mutta niiden laajempi käsittely tulee siirtää myöhemmäksi. Tämä parantaa arvioinnin tehoa (Kahn & Prail 1994). Tilaisuuden loppuksi arvioijille jaetaan lista ongelmista.

Arvioinnin aikana arvioijat havaitsevat, kuinka toiset etsivät ja löytävät ongelmia. Tämä on hyväksi kokemattomille arvioijille, sillä he huomaavat kuinka kokeneemmat käyttävät välineitään hyväksi ja oppivat itsekin tehostamaan arviointiaan.

Ratkaisuvaiheessa etsitään ja toteutetaan ongelmien ratkaisut. Pienille järjestelmille tämä voi olla hyvin suoraviivainen prosessi, suuremmissa voi tulla ongelmia. Saattaa käydä niin, että ongelman korjaaminen vaatii jonkun arviointiryhmään kuulumattoman henkilön ohjelmakoodin muokkaamista, eikä kellään ole oikeutta mennä sitä muuttamaan. Tällöin kirjataan koodin omistaja myös ongelman omistajaksi. Hänen vastuulleen jää tunustaako ja korjaako hän ongelman vai ei. Monimutkaisissa tapauksissa saatetaan myös kerätä ongelmia ryhmiin tai jaotella niitä vakavuusastei-

den mukaan.

Arvioijat tuottavat ratkaisuja aivorihiperiaatteella, joista tekijä ja ongelmien omistajat valitsevat ja toteuttavat ne. Valvoja antaa palautetta arvioijille. Valvoja voi myös tarkastaa järjestelmän jälkeinpäin tai ottaa yhteyttä tekijään varmistaakseen, että ratkaisut on toteutettu.

Kun ratkaisut on toteutettu, valvoja kerää, arvioi ja jakelee tietoa arviointiprosessista. Kiinnostuksen kohteena on usein löydetyt ongelmat työtuntia kohden ja ongelmien korjausprosentti. Tieto jaetaan arviointiryhmälle sekä tarvittaessa yrityksen johdolle. Tämän jälkeen arviointiprosessi katsotaan päättyneeksi.

5.3 Yhteenveto

Muodollinen käytettävyyden arviointimenetelmä on kehitetty järjestelmän kehittäjien käyttöön. Muodoltaan se muistuttaa ohjelmakoodin tarkastusmenettelyä. Se muodostuu kuudesta askeleesta; suunnittelusta, aloituskokouksesta, valmistelusta, arvioinnista, ratkaisujen kehittämisestä sekä seurannasta. Arvioinnissa käytetään apuna ihmisen tehtävänsuoritusmallia (kuva 3), heuristiikkoja (taulukot 2, 3, 6 ja 7), käyttäjäprofiileja sekä tehtäviä. Arvioijat käyvät kohteen ensin läpi yksin, jonka jälkeen varsinainen arvointi suoritetaan ryhmässä. Ryhmäarvioinnin jälkeen kehitetään ratkaisuja ongelmiin.

Ryhmän jäsenillä on määritellyt roolit: valvoja, arvioijat, tekijä ja kirjuri. Kuudella vaiheellaan menetelmä asettuu käytettävyyssuunnittelun elinkaarelle pelkkää arviointia laajemmin.

6 Vertailua

Tässä luvussa verrataan edellä esiteltyjä menetelmiä toisiinsa. Tarkastelun kohteena ovat vaatimukset arvioijille, sijoittuminen ohjelmistotuotantoprosessin elinkaarelle sekä resurssivaativuus. Ensiksi kuitenkin tarkastellaan käytettävyyden testaamisen ja arvioinnin eroja.

6.1 Käytettävyyden arviointi vs. käytettävyydestaus

Käytettävyyden arviointimenetelmät ovat menetelmiä, joissa arvioijat (yleensä käytettävyydsasiantuntijoita) tutkivat järjestelmän käytettävyyttä. Käytettävyydestauksessa tulevien käyttäjien edustajat suorittavat tehtäviä järjestelmällä joko laboratorio-oloissa tai todellisessa työympäristössään. Heidän tekemisiään tarkkaillaan ja kohdatut ongelmat kirjataan. Menetelmät eivät ole toisiaan poissulkevia, vaan täydentäviä, sillä niillä löydetyt käytettävyysongelmat ovat eri tyyppisiä (Nielsen 1994b).

Taulukko 9: Käytettävyydestauksen ja -arvioinnin ongelmamäärät

| | Arvioijat | Löydetyt ongelmat | Parannusehdotukset |
|---------------------------------|----------------------------|-------------------|--------------------|
| <i>Testaus</i> | - | 25 | 31 |
| <i>Heuristinen arviointi</i> | Käytettävyydsasiantuntijat | 44% | 77% |
| | Järjestelmän kehittäjät | 16% | 3% |
| | Noviisit | 8% | 6% |
| <i>Kognitiivinen läpikäynti</i> | Käytettävyydsasiantuntijat | 28% | 16% |
| | Järjestelmän kehittäjät | 16% | 3% |
| | Noviisit | 8% | 6% |

Käytettävyystestauksella löydetään arviointia enemmän käytettävyysongelmia (Desurvire 1994). Taulukoissa 9 ja 10 on listattu eräässä tutkimuksessa (Desurvire et al. 1992) havaitut erot ongelmien määrässä ja niiden jakautuminen eri vakavuusasteille. Tutkimuksen testausosuus tehtiin laboratorio-oloissa. Prosenttiluvut kertovat, kuinka suuri osuus testauksella löydettyistä ongelmista löydettiin arviointimenetelmillä.

Taulukko 10: Löydettyjen käytettävyysongelmien jakautuminen eri vakavuusasteille

| | Arvioijat | Ärsyttävyys | Virheen aiheutuminen | Tehtävän estyminen |
|---------------------------------|--------------------------|-------------|----------------------|--------------------|
| <i>Testaus</i> | - | 5 | 3 | 17 |
| <i>Heuristinen arviointi</i> | Käytettävyyssiantuntijat | 80% | 67% | 29% |
| | Järjestelmän kehittäjät | 40% | 0% | 12% |
| | Noviisit | 20% | 0% | 6% |
| <i>Kognitiivinen läpikäynti</i> | Käytettävyyssiantuntijat | 40% | 67% | 18% |
| | Järjestelmän kehittäjät | 0% | 0% | 12% |
| | Noviisit | 20% | 0% | 6% |

Taulukoista 9 ja 10 nähdään, että arviointimenetelmiä käytettäessä paras tulos saavutetaan käyttämällä arvioijina käytettävyyssiantuntijoita. Esimerkiksi kognitiivisessa läpikäynnissä järjestelmän kehittäjät löysivät käytettävyysongelmia vasta kun tehtävän suorittaminen estyi. Heuristinen arviointi näyttäisi olevan parempi menetelmä pienten, ärsyttävien ongelmien löytämiseen. Taulukosta huomataan myös, että järjestelmän kehittäjät ovat vain hieman parempia arvioijia kuin noviisit.

Taulukosta 9 nähdään, että käytettävyyden arviointimenetelmiä ei kannata käyttää testaamisen sijasta, vaikka arvioijina olisikin asiantuntijoita. Heuristinen arviointi kahdesta menetelmästä parempana löysi 44% testauksella löydetyistä virheistä, ja kognitiivinen läpikäynti 28%. Arviointimenetelmät löysivät myös vain vähän tehtävän estymisen aiheuttavia virheitä. Paras hyöty arviointimenetelmistä saadaan käytettävyyssuunnittelun alkuvaiheessa sekä varojen puutteessa silloin, kun vaihtoehtona olisi olla tekemättä mitään.

6.2 Arvioijat

Kaikki edellä esitetyt menetelmät ovat pääosin ryhmäarviointiin perustuvia. Tosin heuristinen arviointi ja kognitiivinen läpikäynti voidaan tarpeen vaatiessa suorittaa yksinkin. Käytettävyydsiantuntemuksen osalta kognitiivinen läpikäynti on vaativin, hyvät tulokset sen käytössä ovat saavutettavissa vain kokemuksen kautta. Heuristisessa ja muodollisessa arviointimenetelmässä voi olla mukana myös loppukäyttäjiä. Taulukossa 11 on koottu yhteen eri menetelmien vaatimukset arvioijille.

Taulukko 11: Menetelmien vaatimukset arvioijille

| | Heuristinen arviointi | Kognitiivinen läpikäynti | Muodollinen arviointimenetelmä |
|--------------------|-----------------------|--------------------------|--------------------------------------|
| Määrä | 3 - 5 | ryhmä | 4 - 8 |
| Käytettävyykokemus | suositeltavaa | osalla | osalla |
| Menetelmäkokemus | suositeltavaa | edellytys | suositeltavaa |
| Loppukäyttäjiä | mukana | ei mukana | ei mukana |
| Muuta | - | eri alojen tuntijoita | eri alojen tuntijoita, tarkat roolit |

6.3 Sijoittuminen elinkaarelle

Kappaleessa 1.6.3 todettiin, että käytettävyyden arviointimenetelmät soveltuvat käytettäväksi suhteellisen aikaisessa ohjelmistotuotantoprosessin vaiheessa. Samoin todettiin, että niitä ei kannata käyttää kovin myöhäisessä vaiheessa. Suuria eroja yksittäisten menetelmien välillä ei ole, vaan kaikkia kolmea voidaan käyttää sekä paperilla olevien määrittelyjen että toimivien prototyyppien arviointiin.

6.4 Resurssivaativuus

Suurin kulutettava resurssi käytettävyyttä arvioitaessa on arvioijien aika. Tässä suhteessa taloudellisin menetelmä on heuristinen arviointi; sen oppii puolessa päivässä ja suurimmassa osassa tapauksista arviointi sujuu yhden päivän aikana (Nielsen 1994b). Kognitiivisen läpikäynnin tehokas hyödyntäminen vaatii kognitiivisen psykologian tietoja (Wharton et al. 1994), joten sen oppiminen on huomattavasti työläämpää. Läpikäynnin suorittamiseen ei kuitenkaan lopuksi kulu juurikaan enempää aikaa kuin heuristiseen arviointiin. Muodollisessa arviointimenetelmässä erään (Kahn & Prail 1994) laskelman mukaan viideltä ihmiseltä (valvoja, tekijä + 3 arvioijaa) kuluu koko arviointimenetelmän läpivientiin 94 tuntia; noin 19 tuntia henkeä kohti. Todelliset ajat riippuvat monista eri tekijöistä, mutta kaikilla menetelmillä ne ovat samaa suuruusluokkaa.

Muita resursseja ovat mm. tarvittavat tiedonkeruuvälineet (piirtoheitin, videokamera, lomakkeet...), tilat, tietokoneet jne. Nämä kuitenkin riippuvat varsin vähän itse arviointimenetelmästä, sillä minkään menetelmän käyttö ei vaadi tiettyjä välineitä.

7 Yhteenveto

Käytettävyyden arviointimenetelmien käyttöalueet löytyvät ohjelmistotuotantoprosessin alkuvaiheilta jossa niitä käytetään käyttöliittymän määrittämissä arvioimiseen, sekä käytettävyyssuunnittelun iteratiiviseen vaiheeseen, jossa niillä 'puhdistetaan' käyttöliittymä ilmeisimmistä virheistä ennen käytettävyydestä. Arviointimenetelmät eivät yksinään ole riittävä vaihtoehto kunnolliseen käytettävyyden kehittämiseen. Arviointimenetelmillä ei juurikaan löydetä tehtävän suorittamisen estäviä sovellusaluekohtaisia virheitä. Yhdessä käytettävyydestäuksen kanssa arviointimenetelmät tarjoavat kuitenkin vankan perustan käytettävyyden kehittämiseksi. Toki pelkkiä arviointimenetelmiä kannattaa käyttää silloin, jos varat, aika tai jokin muu seikka estää kattavamman tarkastelun. Aina on parempi tehdä jotain käytettävyyden hyväksi, kuin ei mitään.

Sopivan menetelmän valinta on riippuu käytettävistä arvioijista, ajasta ja monista muista seikoista. Yleispätevää ohjetta valintaan ei ole, mutta jonkinlaista ohjenuoraa voi kuitenkin antaa. Kognitiivinen läpikäynti on sijoitus tulevaisuuteen; sen opettelu vie aikaa, mutta kokeneen arvioijan käytössä se on tehokas. Kognitiivisen läpikäynnin käyttöä kannattaa harkita silloin, jos tarvitaan tehtävän syvällistä ymmärrystä. Heuristinen arviointi on helppo oppia ja nopea käyttää. Tämän lisäksi se on myös tehokas ja halpa menetelmä. Tämän vuoksi sitä voi suositella varsinkin aloitteleville arvioijille. Muodollinen menetelmä voisi olla luonnollinen valinta yritykselle, jossa on jo käytössä ohjelmakoodin tarkastusmenettelyt. Prosessi olisi jo tuttu ja siten nopeuttaisi menetelmän käyttöönottoa ja vähentäisi mahdollisia pelkoja.

Viitteet

Anderson, J.R. (1987). Skill acquisition: Compilation of weak-method solutions. *Psychological Review* 94: 192-211.

Bradford, J.S. (1994). Evaluating High-Level Design. *Usability Inspection Methods*, ed. Nielsen, J., & Mack, R.L.: 235.

Carroll, J.M., & Rosson, M.B. (1987). The paradox of the active user. *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*, ed. Carroll, J.M., Cambridge: Bradford Books/MIT Press: 80-111.

Desurvire, H.W. (1994). Faster, Cheaper!! Are Usability Inspection Methods as Effective as Empire Testing? *Usability Inspection Methods*, ed. Nielsen, J., & Mack, R.L.: 173-202.

Desurvire, H.W., Kondziela, J.M., & Atwoodm M. (1991). What is gained and lost when using evaluation methods other than empirical testing. *People and Computers VII*, ed. Monk, A., Diaper., D, & Harrison, M.D. Cambridge: Cambridge University Press: 89-102.

Engelbeck, G.E. (1986). Exceptions to generalizations: Implications of formal models of human-computer interaction. *Unpublished Masters Thesis*, Department of Psychology, University of Colorado, Boulder.

Fisher, G. (1991). Supporting learning on demand with design environments. *Proceedings of the International Conference on the Learning Sciences, 1991* (Envaston, IL, August). Charlottesville, VA: Association for the Ad-

vancement of Computing in Education: 165-172.

Hackman, J.R., & Morris, C.G. (1975). Group tasks, group interaction process, and group performance effectiveness: A review and proposed integration. *Advances in Experimental Social Psychology 8*, ed. Berfkowitz, L. New York: Academic Press.

Jeffries, R., Miller, J.R., Wharton, C., & Uyeda, K.M. (1991). User interface evaluation in the real world: A comparison of four techniques. *Proceedings ACM CHI'91 Conference* (New Orleans, LA, April 28-May 2): 119-124.

Kahn, M.J., & Prail, A. (1994). Formal Usability Inspections. *Usability Inspection Methods*, ed. Nielsen, J. & Mack, R.L.: 141-171.

Karat, C. (1994). A Comparison of User Interface Evaluation Methods. *Usability Inspection Methods*, ed. Nielsen, J., & Mack, R.L.: 203-233.

Karat, C., Campbell, R.L., & Fiegel, T. (1992). Comparison of empirical testing and walkthrough methods in user interface evaluation. *Proceedings ACM CHI'92 Conference* (Monterey, CA, May 3-7): 397-404.

Lewis, C., Polson, P., Wharton, C., & Rieman, J. (1990). Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. *Proceedings ACM CHI'90 Conference* (Seattle, WA, April 1-5): 235-242.

Lewis, C., Polson, P.G., & Rieman, J. (1991). Cognitive walkthrough forms and instructions. Institute of Cognitive Science *Technical Report #ICS 91-14*.

University of Colorado, Boulder, CO, 80309.

Mack, R., & Montaniz, F. (1994). Observing, Predicting, and Analyzing Usability Problems. *Usability Inspection Methods*, ed. Nielsen, J., & Mack, R.L.: 295-339.

Mack, R., & Nielsen, J. (1994). Executive Summary. *Usability Inspection Methods*, ed. Nielsen, J. & Mack, R.L.: 1-23

Molich, R., & Nielsen, J. (1990). Improving a human-computer dialogue. *Communications of the ACM* 33, 3 (March): 338-348.

Nielsen, J., & Landauer, T.K. (1993). A mathematical model of the finding of usability problems. *Proceedings ACM/IFIP INTERCHI'93 Conference* (Amsterdam, The Netherlands, April 24-29): 414-417.

Nielsen, J. (1989). Usability engineering at a discount. *Designing and Using Human-Computer Interfaces and Knowledge Based Systems*, ed. Salvendy, G., & Smith, M.J., Amsterdam: Elsevier Science Publishers: 394-401.

Nielsen, J. (1990). Big paybacks from 'discount' usability engineering. *IEEE Software* 7, 3 (May): 107-108.

Nielsen, J. (1992). Finding usability problems through heuristic evaluation. *Proceedings ACM CHI'92 Conference* (Monterey, CA, May 3-7): 373-380.

Nielsen, J. (1993). *Usability Engineering*. Boston: Academic Press.

Nielsen, J. (1994a). Guerrilla HCI: Using discount usability engineering to penetrate the indimitation barrier. *Cost-Justifying Usability*, ed. Bias, R.G., & Mayhew, D.J. Boston: Academic Press: 245-272.

Nielsen, J. (1994b). Heuristic Evaluation. *Usability Inspection Methods*, ed. Nielsen, J., & Mack, R.L.: 25-62.

Nielsen, J. (1994c). Enhancing the explanatory power of usability heuristics. *Proceedings ACM CHI'94 Conference* (Boston, MA, April 24-25).

Nielsen, J., & Mack, R.L. (ed.) (1994). *Usability Inspection Methods*. John Wiley & Sons, Inc.

Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. *Proceedings ACM CHI'90 Conference* (Seattle, WA, April 1-5): 249-256.

Polson, P., & Lewis, C. (1990). Theory-based design for easily learned interfaces. *Human-Computer Interaction* 5, 2&3: 191-220.

Polson, P., Lewis, C., Rieman, J., & Wharton, C. (1992). Cognitive walkthroughs: A method for theory-based evaluation of user interfaces. *International Journal of Man-Machine Studies* 36: 741-773.

Rieman, J., Davies, S., Hair, D.C., Esemplare, M., Polson, P.G., & Lewis, C. (1991). An automated cognitive walkthrough. *Proceedings ACM CHI'91 Conference* (New Orleans, LA, April 28-May 2): 427-428.

Rowley, D.E., & Rhoades, D.G. (1992). The cognitive jogthrough: a fast paced user interface evaluation procedure. *Proceedings ACM CHI'92 Conference* (Monterey, CA, May 3-7): 389-395

Wharton, C. (1992). Cognitive Walkthroughs: Instructions, Forms and Examples. Institute of Cognitive Science *Technical Report* CU-ICS-92-17 University of Colorado, Boulder, CO 80309.

Wharton, C., Bradford, J., Jeffries, R., & Franzke, M. (1992). Applying cognitive walkthroughs to more complex user interfaces: Experiences, issues, and recommendations. *Proceedings ACM CHI'92 Conference* (Monterey, CA, May 3-7): 381-388.

Wharton, C., Rieman, J., Lewis, C., & Polson, P. (1994). The Cognitive Walkthrough Method: A Practitioner's Guide. *Usability Inspection Methods*, ed. Nielsen, J., & Mack, R.L.: 105-140.