

**CASE-välineiden käyttäytyvyyden vaikuttavat
tekijät: empiirinen tutkimus suuressa tietoliikennealan
yrityksessä**

Jarkko Ruuska

Tampereen yliopisto
Tietojenkäsittelytieteiden laitos
Pro gradu -tutkielma
Toukokuu 2001

Tampereen yliopisto

Tietojenkäsittelytieteiden laitos

Jarkko Ruuska: CASE-välineiden käyttäjätyytyväisyyteen vaikuttavat tekijät: empiirinen tutkimus suuressa tietoliikennealan yrityksessä

Pro gradu -tutkielma, 86 sivua, 5 liitesivua

Toukokuu 2001

Tämän työn tarkoituksena on tutkia, mitkä tekijät vaikuttavat CASE-välineiden käyttäjätyytyväisyyden muodostumiseen. Työ koostuu kahdesta eri osiosta, kirjallisuuskatsauksesta ja empiirisestä osiosta. Kirjallisuuskatsauksessa CASE-välineiden käyttäjätyytyväisyyttä tutkitaan välineen ominaisuuksien, ympäristön ja käyttäjien henkilökohtaisten ominaisuuksien kannalta. Empiirisessä osiossa taas käsitellään kyselytutkimuksen avulla saatuja tuloksia MetaEdit+ -ohjelmiston käyttäjätyytyväisyydestä suuressa tietoliikennealan yrityksessä. Saatuja tuloksia verrataan myös kirjallisuuskatsauksessa esiin tulleisiin tekijöihin.

Avainsanat: CASE-välineet, käyttäjätyytyväisyys, meta-CASE

Sisälllys

1	LYHENTEET	5
2	HUOMIOITA	5
3	JOHDANTO	5
4	CASE-VÄLINEET JA KÄYTTÄJÄTYTYVÄISYYS KIRJALLISUUDESSA	7
4.1	CASE-välineiden käyttäjätyytyväisyyteen vaikuttavien tekijöiden jako kategorioihin	8
4.2	CASE-välineiden käyttäjätyytyväisyyden mittaaminen	10
4.3	CASE-välineet ja ympäristön vaikutus käyttäjätyytyväisyyteen	11
4.3.1	Koulutus ja käyttöönotto	12
4.3.2	Vapaaehtoisuus	14
4.3.3	Organisaation rakenteelliset elementit	16
4.3.4	Johdon tuki	16
4.4	CASE-välineiden ominaisuuksien vaikutus käyttäjätyytyväisyyteen	17
4.4.1	Kompleksisuus	18
4.4.2	Käyttöliittymä	18
4.4.3	Menetelmäkeskeinen rakenne	20
4.4.4	Uudelleenkäyttö ja yhteensopivuus	23
4.4.5	Räätälöitävyys ja tuki monelle käyttäjälle	26
4.5	CASE-välineet ja yksilön ominaisuuksien vaikutus käyttäjätyytyväisyyteen	27
5	METAEDIT+ JA KÄYTTÄJÄTYTYVÄISYYS SUURESSA TIETOLIIKENNEALAN YRITYKSESSÄ	30
5.1	MetaEdit+:n esittely	30
5.2	MetaEdit+:n käyttöönottoon johtaneet syyt	32
5.2.1	Tuotetun ohjelmakoodin modulaarisuus	32
5.2.2	Projektin parempi hallinta	32
5.2.3	Suunnitteludokumenttien tuottamisen automatisointi	33
5.3	MetaEdit+:n ennakko-odotusten toteutuminen johdon näkökulmasta	33
5.4	MetaEdit+ ja CASE-välineiden luokat	34
5.5	MetaEdit+:n ominaisuuksien hyödyntäminen	35
5.6	MetaEdit+:n käyttäjätyytyväisyyttä tutkiva empiirinen tutkimus	36
5.6.1	Tutkimusmenetelmät	37
5.6.2	Tutkimuksen osallistujat	38
5.6.3	Tutkimushypoteesit	40
6	KYSELYTUTKIMUKSEN TULOKSET JA NIIDEN ANALYSOINTI	41

6.1	MetaEdit+ :n käyttäjättyytyväisyyteen vaikuttavat ominaisuudet	41
6.1.1	Yleistä	41
6.1.2	Kompleksisuus	43
6.1.3	Käyttöliittymä ja käyttäjättyytyväisyys	45
6.1.4	Käyttöliittymän puutteet	46
6.1.5	Käyttöliittymän vahvat puolet	52
6.1.6	Koodigeneraattorin toiminta	53
6.1.7	Uudelleenkäyttö ja yhteensopivuus	54
6.1.8	Räätälöitävyys ja tuki monelle käyttäjälle	56
6.1.9	Menetelmäkeskeinen rakenne	57
6.1.10	Ehdotuksia MetaEdit+ :n ominaisuuksien parantamiseksi	58
6.2	MetaEdit+ ja ympäristön vaikutus	59
6.2.1	Dokumentointi	60
6.2.2	Koulutus ja käyttöönotto	61
6.2.3	Vapaaehtoisuus	63
6.2.4	Organisaation rakenteelliset elementit	64
6.2.5	Johdon tuki	64
6.2.6	Tuki prosessille	65
6.3	MetaEdit+ ja yksilölliset tekijät	66
6.3.1	Oppimiskynnys	66
6.3.2	Opittujen taitojen muuttuminen arvottomiksi	69
6.4	Tutkimushypoteesien toteutuminen	70
6.4.1	Hypoteesin H1 toteutuminen	71
6.4.2	Hypoteesin H2 toteutuminen	74
6.4.3	Hypoteesin H3 toteutuminen	75
6.4.4	Hypoteesin H4 toteutuminen	76
6.4.5	Hypoteesin H5 toteutuminen	76
7	PÄÄTELMÄT JA YHTEENVETO	78
	LÄHTEET	81

1 Lyhenteet

CASE	Computer-Aided / Assisted Systems / Software Engineering
MIS	Management Information Systems
OOC	Object Oriented C

2 Huomioita

Tutkimuksen kohteena oleva yritys on tarkoituksella jätetty mainitsematta salassapitosopimuksen vuoksi. Tämän vuoksi viitataan kyseiseen yritykseen tästä eteenpäin yksinkertaisesti sanalla yritys.

3 Johdanto

Viime vuosina ohjelmistokehityksen kilpailuasetelmat ovat kiristyneet kaikilla rintamilla ja yrityksille on yhä tärkeämpää tuoda uusia ja laadukkaampia tuotteita yhä nopeammin markkinoille. Siksi on myös tärkeää etsiä nopeampia ja tehokkaampia metodeja ja välineitä tuotteiden kehittämiseen. Eräs tapa tehostaa tuotekehitystä on käyttää CASE-välineitä [Norman, 1998; Stobart et al., 1993; Ho, 1992], jotka tukevat uudelleenkäyttöä ja ovat tutkimusten [Dabin, 1991] mukaan nopeammin omaksuttavissa kuin perinteiset kolmannen sukupolven ohjelmointikielet. Tämän vuoksi eräs CASE-välineiden käyttöä puoltava seikka on yksinkertaisesti pula osaavasta työvoimasta [Dabin, 1991; Kemerer, 1992; Sumner, 1995]. On myöskin todettu, että CASE-välineiden käytöllä saavutetaan merkittäviä parannuksia tuotteiden laatuun [Jones, 1992; Low ja Leenanuraksa, 1999; Stobart et al., 1993; Bailer et al., 1993; Kusters ja Wijers, 1993].

CASE-välineiden käyttö ei kuitenkaan ole ongelmaton, sillä niiden monimutkaisuuden ja monipuolisten ominaisuuksien vuoksi niitä ei ole aina helppo omaksua ja käyttää. Tämä tutkimus pyrkii tutkimaan teoreettisin ja empiirisin keinoin sitä, mitkä tekijät vaikuttavat CASE-välineiden käyttäjätyytyväisyyden muodostumiseen. On tärkeää, että käyttäjät kokevat CASE-työkalujen käytön tarpeelliseksi ja hyödylliseksi, koska nämä tekijät vaikuttavat

yrityksissä asetettujen taloudellisten ja laadullisten tulosten saavuttamiseen [Iivari ja Maansaari, 1997]. Käyttäjätyytyväisyydellä tarkoitetaan tässä tutkimuksessa yksinkertaisesti sitä, miten käyttäjät kokevat CASE-välineiden käytön ja kuinka tyytyväisiä he niihin ovat.

Tässä tutkimuksessa käsittelen CASE-välineinä yleisesti ohjelmointiympäristöjä, jotka sisältävät täydelliset välineet ohjelmakoodin tuottamiseen. Kirjallisuuskatsauksessa ei oteta kantaa siihen, millä tavalla ohjelmistojen toiminnallisuus on toteutettu, vaan ohjelmistoja käsitellään hyvin yleisellä abstraktiotasolla.

CASE-välineissä ohjelmointiin käytetään usein yksityiskohtaista graafista kuvauskieltä, joka sisältää itsenäisiä graafisia primitiivejä, joista ohjelmakoodi kootaan visuaalisesti. Graafiset oliot talletetaan erilaisiin tietovarastoihin työkaluista riippuen. Tyypillisesti ympäristöön kuuluu vielä koodigeneraattori, joka generoi tehdyistä visuaalisista ilmaisuista jonkin kolmannen sukupolven ohjelmointikielen koodia. Käännös- ja linkitysvaihe suoritetaan normaalilla kolmannen sukupolven ohjelmointikielen kääntäjällä.

Käyttäjätyytyväisyyden mittaaminen on vaikeaa, koska kyse on subjektiivisista käyttäjien mielipiteistä. Yleispäteviä mittareita käyttäjätyytyväisyyden mittaamiseen on hankala määrittellä, joten ilmeiseksi vaihtoehdoksi jää tietojen kerääminen käyttäjiltä [Moore ja Benbasat, 1991; Hartrum et al., 1989]. Tämän tutkimuksen empiirisessä osiossa käyttäjätyytyväisyyttä tutkittiin eräässä suuressa tietoliikennealan yrityksessä, joka käytti MetaEdit+ -ohjelmistoa. Tätä tarkoitusta varten muodostettiin testikysymykset, joiden katsottiin mittaavan erilaisia käyttäjätyytyväisyyteen liittyviä tekijöitä (liite 1). Tuloksia arvioitiin viiden asetetun tutkimushypoteesin avulla, jotka on esitelty kohdassa 5.6.3.

CASE-välineiden synnyttämää käyttäjätyytyväisyyttä tai sen puutetta on tutkittu varsin vähän. Yleinen mielenkiinto on keskittynyt pikemminkin CASE-välineiden tehokkuuden ja ominaisuuksien arviointiin. Jarzabekin ja Huangin mielestä nykyään markkinoilla olevissa välineissä ei ole tarpeeksi huomioitu käyttäjiä ja heidän tarpeitansa [Jarzabek ja Huang, 1998]. Iivari ja Maansaari ovat tutkineet CASE-välineiden vaikutuksia ammattilaisten työnkuvaan ja motivaatioon [Iivari ja Maansaari, 1997]. Iivari on myös tehnyt tutkimusta siitä, miksi CASE-työkaluja ei haluta käyttää [Iivari, 1996]. Ivan Aaen on suorittanut tutkimusta CASE -välineiden organisaationaalista vaikutuksesta kyselytutkimuksen avulla [Aaen, 1993]. Diane Lending ja Norman L. Chervany ovat tehneet ehkä eniten tätä tutkimusta

muistuttavan kyselyn käyttäjien suhtautumisesta CASE-välineisiin [Lending ja Chervany, 1998]. Finnigan et al. ovat kysyneet käyttäjiltä, minkälainen olisi heidän ihanteensa CASE-välineeksi [Finnigan et al., 2000].

Saatujen tulosten mukaan on selvää, että useat markkinoilla olevista työkaluista eivät tyydytä käyttäjiä. Syytä tähän on kuitenkin useita, eikä syy aina ole tarjolla olevissa välineissäkään. Monien tutkimusten mukaan suurimpia vaikuttavia tekijöitä ovat itse asiassa ympäristölliset tekijät [Iivari, 1996; Iivari ja Maansaari, 1997]. Merkittävä tekijä käyttäjätyytyväisyyden kannalta on myös se, miten uusi väline organisaatiossa esitellään käyttäjille [Orlikowski, 1992]. Jos välineen käyttö on vapaaehtoista ja koulutus sen käyttöön on käyttäjien toiveita vastaavaa, näyttävät työkalun käyttäjät olevan tyytyväisiä myös itse työkalun käyttöön. Järjestetyn kyselytutkimuksen perusteella tutkitussa yrityksessä käytetyssä MetaEdit+ -työkalussa pidettiin eniten sen tarjoamasta ohjelmakoodin nopeammasta tuottamisesta ja graafisen suunnittelun havainnollisuudesta. Kuitenkin useat käyttöliittymään ja koodigeneraattoriin liittyvät puutteet vaikuttivat käyttäjätyytyväisyyteen negatiivisesti.

Tutkimukseni jakaantuu kahteen osaan: teoreettisen kirjallisuuskatsaukseen sekä teorian soveltamiseen käytäntöön. Ensimmäisessä osassa käsitelen yleisesti käyttäjätyytyväisyyden tutkimusta ja esittelen eräitä jaotteluita ja teorioita. Viidennestä luvusta alkava toinen osa keskittyy kyselytutkimuksen tuottamaan aineistoon MetaEdit+:n käyttäjätyytyväisyydestä, jota pohditaan ensimmäisen osan tarjoaman viitekehyksen puitteissa.

4 CASE-välineet ja käyttäjätyytyväisyys kirjallisuudessa

Jotta voisimme paremmin käsitellä tutkimuksen toisen osan empiirisen tutkimuksen tuottamia tuloksia, on ensin syytä tutustua muihin aiheita käsitteleviin tutkimuksiin ja niissä saavutettuihin tuloksiin. Näin saamme ensin yleiskuvan käsitelystä aihepiiristä ja välineet kyselytutkimuksen tulosten analysointiin.

Tutkimukseni alkaa CASE-välineisiin vaikuttavien seikkojen luokkajaon arvioinnilla ja jatkuu pohdinnalla siitä, miten käyttäjätyytyväisyyttä tulisi mitata. Tämän jälkeen käsitellään niitä seikkoja, jotka vaikuttavat käyttäjien käsitysten syntymiseen välineestä. Ensiksi arvioidaan

ympäristötekijöiden vaikutusta, sitten itse välineiden ominaisuuksien vaikutusta ja kolmantena käyttäjien yksilöllisten erojen vaikutusta käyttäjätyytyväisyyteen. Tämän jälkeen keskustellaan muista tekijöistä, joita ei voida sovittaa mihinkään kolmesta esittelemästani pääkategoriasta.

4.1 CASE-välineiden käyttäjätyytyväisyyteen vaikuttavien tekijöiden jako kategorioihin

Tämän kappaleen tarkoituksena on kartoittaa niitä tekijöitä, jotka vaikuttavat siihen, miten käyttäjät CASE-välineiden käytön kokevat. Jaotteluita on useita, mutta tässä tutkimuksessa jaottelen käyttäjätyytyväisyyteen vaikuttavat tekijät neljään kategoriaan seuraavasti:

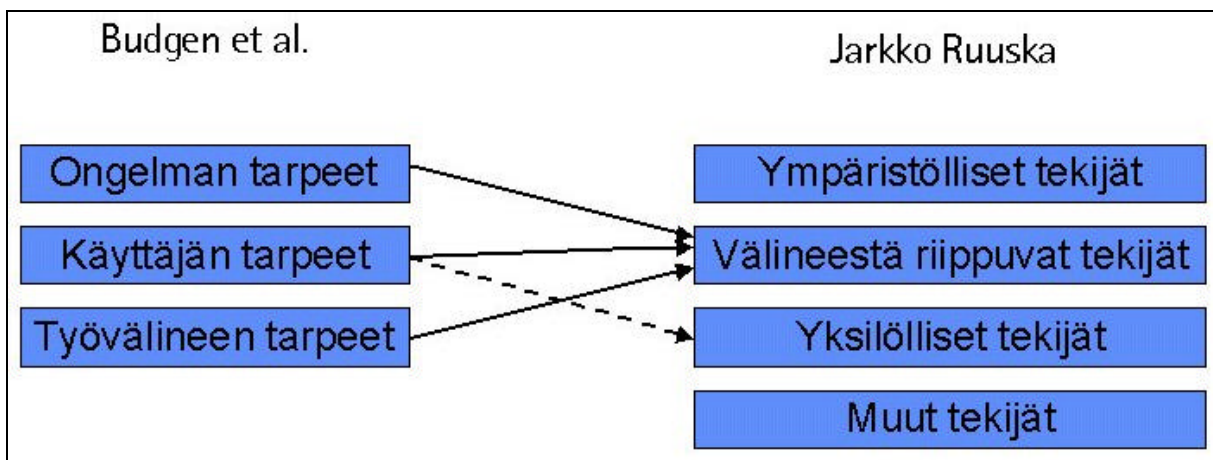
1. Ympäristölliset tekijät
2. Välineestä riippuvat tekijät
3. Yksilölliset tekijät
4. Muut tekijät

Budgen et al. puolestaan [Budgen et al., 1993] jaottelevat CASE-välineen suunnitteluympäristön tarpeet seuraavasti:

1. Ongelman tarpeet
2. Käyttäjän tarpeet
3. Työvälineen tarpeet

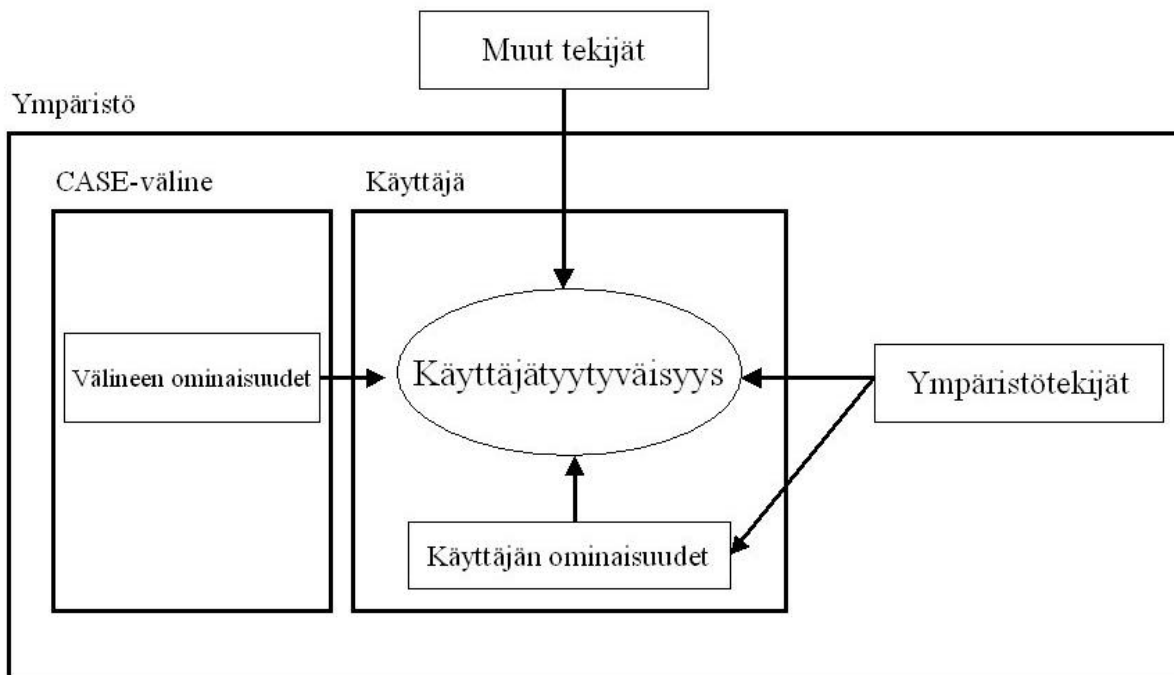
Budgenin et al. jaottelu on varsin looginen suunnittelijan näkökulmasta, mutta se ei kuitenkaan mielestäni ota tarpeeksi kantaa erilaisiin CASE-työvälineiden käytön ympäristötekijöihin. Suurin osa Budgenin et al. jaottelun kohdista sopii ehkä oman jaotteluni kohtaan kaksi (välineestä riippuvat tekijät). Osa käyttäjän tarpeista voidaan lukea myös kohtaan kolme (yksilölliset tekijät) kuuluviksi. Vaikka Budgenin et al. jaottelusta puuttuvat

kokonaan ympäristölliset tekijät, ei se mielestäni ole varsinaisesti puute suunnittelijan kannalta ajateltuna, sillä suunnitteluvaiheessa ei ole etukäteen mahdollista ottaa huomioon kaikkia mahdollisia ympäristötekijöitä. Käyttäjätyytyväisyyden muodostumisessa nämä seikat kuitenkin näyttävät esittävän varsin suurta osaa. Tässä onkin yksi CASE-välineiden suurimmista ongelmista, sillä kehitettävän välineen ominaisuudet eivät ole riittävä tae menestykselle, vaan on luotava myös ihanteelliset olosuhteet sen käyttöönotolle ja käyttämiselle. Näiden eri jaotteluiden vertailu kuviossa 1.



Kuvio 1. Budgenin et al. ja oman jaotteluni vertailua.

Kohdistan huomion myöhemmin tutkimuksessani ensin kolmeen ensimmäiseen kategoriaan, ympäristöllisiin, välineestä riippuviin ja yksilöllisiin seikkoihin. Neljäs kategoria on varattu niille seikoille, jotka eivät kolmeen edelliseen kategoriaan mahtuneet. Tässä tutkimuksessa ei ole identifioitu seikkoja, jotka tällaisia voisivat olla, mutta kategoria on otettu mukaan, jotta jaottelu olisi aukoton. Oma näkemykseni CASE-välineiden käyttäjätyytyväisyyteen vaikuttavista tekijöistä on kuvattu kuviossa 2.



Kuvio 2. CASE-välineen käyttäjätyytyväisyyteen vaikuttavat tekijät.

Tässä kappaleessa tarkoitukseni oli tuoda esiin se seikka, että CASE-välineiden vaikutuksia ei voida aukottomasti tutkia ja ennustaa ainoastaan välineen ominaisuuksia tarkastelemalla. Kuten Aaenkin [Aaen, 1993] toteaa: "...CASE-välineiden vaikutusta ei pitäisi selittää viittauksilla yksittäisiin ominaisuuksiin, vaan pikemminkin välineen kaikkien ominaisuuksien, näistä käytettyjen ominaisuuksien ja välineellä suoritettujen tehtävien muodostaman monimutkaisen vuorovaikutuksen tuloksena."

4.2 CASE-välineiden käyttäjätyytyväisyyden mittaaminen

Käyttäjätyytyväisyys CASE-välineiden käytössä on varsin subjektiivinen kriteeri, koska eri ihmiset voivat olla hyvinkin eri mieltä samoistakin asioista. Tämän vuoksi yleispäteviä mittareita käyttäjätyytyväisyyden mittaamiseksi on vaikea laatia. Onkin usein tyydyttävä tutkimaan erilaisista välineistä saatuja käyttäjien kommentteja ja kokemuksia. Mooren ja Benbasatin [Moore ja Benbasat, 1991] mukaan haastatteluista ja kyselyistä on kuitenkin mahdollista saada luotettavampaa ja yleistettävämpää tietoa kuin muilla objektiivisemmilla metodeilla. Saaduista tiedoista tehtyjen tilastollisten päätelmien oikeellisuutta ja luotettavuutta voidaan mitata esimerkiksi Cronbachin alpha-arvoilla, kuten esimerkiksi Iivari tutkimuksessaan tekee [Iivari, 1996]. Hartrum et al. ovat artikkelissaan esitelleet

metodologian CASE-välineiden arviointia varten [Hartrum et al., 1989]. Myös heidän mukaansa paras tapa arvioida käyttäjätyytyväisyyttä on kuunnella käyttäjien mielipiteitä. He kuitenkin korostavat käyttäjien subjektiivisten kommenttien analysointia kvantitatiivisesti.

Haastattelututkimuksissakin on rajoituksensa, kuten Stobart et al. esittävät artikkelissaan [Stobart et al., 1993]. Heidän kyselytutkimuksensa mukaan ohjelmistotaloissa vain 20 prosenttia henkilöstöstä halusi täysin uusia CASE-työvälineitä ja menetelmiä. Muissa organisaatioissa tilanne oli vielä kummallisempi, sillä tutkimuksen mukaan kukaan ei halunnut täysin uusia CASE-välineitä tai menetelmiä. Kuitenkin 100 prosenttia ohjelmistotalojen käyttäjistä ja 20 prosenttia muiden organisaatioiden CASE-käyttäjistä halusi parantaa jo olemassa olevia välineitä. Näiden tulosten tarkoituksena on osoittaa, että ihmisten henkilökohtaiset asenteet ja ominaisuudet saattavat vaikuttaa haastattelututkimusten paikkansa pitävyyteen. Näistä yksilöllisistä vaikutuksista puhutaan vielä tarkemmin kohdissa 4.5 ja 6.3.

Edellisten kohtien tarkoituksena oli selvittää, millä perusteilla käyttäjätyytyväisyyttä voidaan ylipäätään arvioida. On myöskin muistettava, että tämän tutkimuksen kirjallisuuskatsaus pohjautuu suurelta osin jo tehtyihin empiirisiin tutkimuksiin, joten mittaamiseen liittyviä tekijöitä on syytä arvioida kunkin lähteen kohdalla erikseen.

4.3 CASE-välineet ja ympäristön vaikutus käyttäjätyytyväisyyteen

Tutkittaessa CASE-välineitä laajemmassa mittakaavassa keskitytään usein välineen ominaisuuksiin ulkoisten riippuvuussuhteiden kustannuksella. Mielestäni tutkittaessa ihmisille suunnattuja välineitä on aina huomioitava se organisationaalinen konteksti, joka kulloinkin ohjaa käyttäjien reaktioita. Myös käyttäjätyytyväisyyttä tutkittaessa on syytä muistaa, että välineen ominaisuudet ovat vain osatekijä käyttäjän käsityksen muodostumisessa. Empiirisissä tutkimuksissa on todettu, että myös ympäristöllä on oma vaikutuksensa käyttäjien mielipiteisiin CASE-välineistä [Iivari, 1996].

Seuraavissa kohdissa selvitetään ensin välineistä annettavan koulutuksen vaikutusta käyttäjätyytyväisyyteen ja sen jälkeen todetaan välineiden käytön vapaaehtoisuuden suuri vaikutus.

4.3.1 Koulutus ja käyttöönotto

Koulutus on henkilökohtaisten kokemustenikin mukaan suurimpia CASE-välineiden käyttäjätyytyväisyyteen vaikuttavia ulkoisia tekijöitä. Erityisen tärkeää on, kuinka paljon työntekijöillä on mahdollista saada laadukasta ja juuri heille sopivaa koulutusta välineen käytöstä ja sen tarjoamista ominaisuuksista. Maalaisjärjelläkin ajateltuna on selvää, että käyttäjätyytyväisyyden voidaan olettaa kasvavan työntekijöiden oppiessa käyttämään uutta työkalua yhä tehokkaammin. Käyttäjätyytyväisyys näyttää kasvavan, kun käyttäjät saavat aikaan nopeammin ja helpommin haluamiaan tuloksia. Jones [Jones, 1992] toteaa artikkelissaan, että Hewlett-Packardin laboratorioissa suoritetuissa tutkimuksissa suurin eri työkalujen menestystä ennustava tekijä oli riittävän koulutuksen järjestäminen. Jones myös varoittaa artikkelissaan, että CASE-projekteista, joissa koulutukseen ei ole suunnattu merkittävää taloudellista panosta, yli 50 prosenttia näyttää epäonnistuvan. Epäonnistumisella tarkoitettiin tässä yhteydessä sitä, että pian tutkimuksen jälkeen työkalun käyttö oli lopetettu, käyttäjät olivat tyytymättömiä sen käyttöön tai työkalun avulla ei saavuttu haluttuja tuloksia.

Ho kertoo vuonna 1992 suorittamassaan tutkimuksessaan, että CASE-välineet lisäävät tuottavuutta erityisesti silloin, kun niiden käyttöön on annettu koulutusta [Ho, 1992]. Hon tutkimus tosin keskittyi ainoastaan tietovirtakaavioiden piirtämiseen kuluneen ajan mittaamiseen, mutta koska CASE-välineillä työskentely suurelta osin on juuri piirtämistä, voidaan tätä tulosta pitää relevanttina. Ho toteaa myös, että hyvin suunniteltua koulutusta tarvitaan, jos halutaan CASE-välineiden käytön nostavan tuotteiden laatua.

Koulutuksen laatuun on kuitenkin kiinnitettävä erityistä huomioita, sillä on myöskin todettu, että liian yksityiskohtainen koulutus saattaa entisestään laskea käyttäjätyytyväisyyttä [Iivari, 1996]. Jos käyttäjät eivät osaa käyttää välineen perusominaisuuksiakaan, heitä ei ole syytä hämmentää vielä enemmän kertomalla välineiden edistyksellisimmistä ja monimutkaisimmista mahdollisuuksista. Liian suuri määrä tietoa välineiden ominaisuuksista saattaa vaikeuttaa välineen monimutkaisuuden ymmärtämistä, josta puhutaan kohdassa 4.4.1. Norman et al. toteavat koulutuksen usein vain lisäävän odotuksia välineen ominaisuuksien suhteen, vaikka sen pitäisi enemmänkin auttaa suunnittelijoita välttämään vääriä työskentelytapoja [Lending ja Chervany, 1998].

Bailer et al. toteavat tutkimuksessaan [Bailer et al., 1993], että menestyneiden ohjelmointitiimien muodostamiseen eivät riitä pelkät luennot ja workshop-tyyppiset istunnot. Opetuksen riittävään onnistumiseen tarvitaan varsinaista aitoa omakätistä kokemusta oikeasta projektityöstä. Bailer et al. myös esittelevät CASE-välineiden koulutukseen liittyvän vaikean ongelman. Heidän tutkimuksensa osoittaa, että liian monet yritykset eivät huomioi koulutuksessaan välineen perustana olevaa suunnittelumetodia tai -menetelmää. Tämän vuoksi käyttäjät eivät voi ymmärtää täydellisesti välineen tarjoamia mahdollisuuksia. CASE-välineiden menetelmäkeskeisestä rakenteesta ja sen hyvistä ja huonoista puolista keskustellaan enemmän kohdassa 4.4.3. Bailer et al. pitävät myös välttämättömänä koulutuksen suuntaamista niille henkilöille, jotka potentiaalisesti voivat täyttää uuden järjestelmän käyttämiseen tarvittavat vaatimukset. Mielestäni tämä kuitenkin voi olla hyvinkin vaikeaa, koska uuden CASE-työkalun vaatimuksia juuri kyseisessä organisaatiossa on vaikea ennustaa.

Nosek et al. puolestaan painottavat CASE-välineen ja sen perustana olevan metodologian eriyttämistä toisistaan koulutuksen suhteen. Heidän mielestään käyttäjien tyytyväisyys CASE-välineiden käyttöön kärsii, ellei välineen perustana olevasta metodologiasta anneta koulutusta ennen varsinaista välineen käyttöön liittyvän koulutuksen aloittamista [Nosek et al., 1992].

Finnigan et al. [Finnigan et al., 2000] korostavat tutkimuksessaan sitä, että käyttöönotossa ei saisi samanaikaisesti esitellä sekä uutta välinettä että uutta metodologiaa. Tästä seuraa heidän mukaansa suuria ongelmia käyttäjien kannalta, koska heidän täytyy opetella liian paljon uusia asioita kerralla. Orlikowskin [Orlikowski, 1992] mukaan myös välineen käyttöönottoon varatulla ajalla ja käyttöönottoaikataululla on merkitystä käyttäjien tyytyväisyyteen. Mitä enemmän aikaa käyttäjillä on sopeutua uuteen järjestelmään, sitä tyytyväisempiä he todennäköisesti ovat.

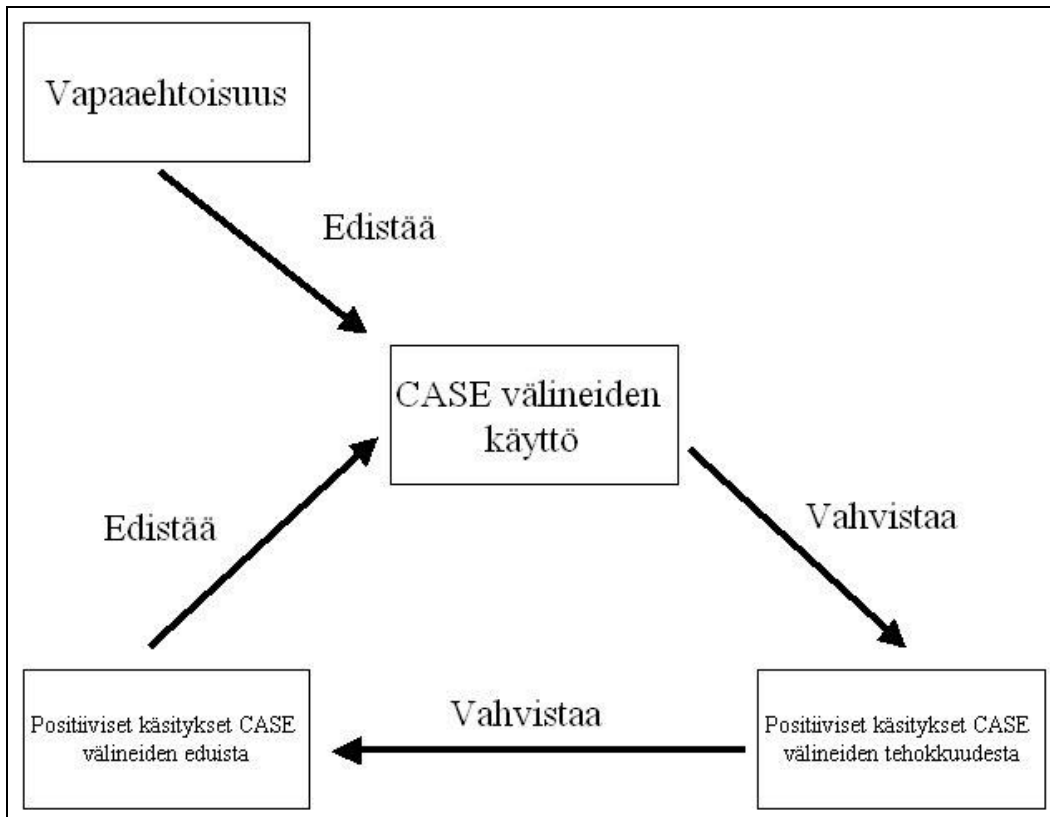
Edellisissä kohdissa tarkoitukseni oli korostaa, kuinka suuri merkitys koulutuksella ja hyvin suunnitellulla käyttöönotolla on CASE-välineiden käyttäjätyytyväisyyden muodostumisessa. Koulutuksen tulee olla käyttäjien kannalta oleellista ja sen pitää huomioida käyttäjien lähtötaso välineen käytön suhteen.

4.3.2 Vapaaehtoisuus

Tutkimusten mukaan vapaaehtoisuus on merkittävä CASE-välineiden käyttöön ja käyttäjätyytyväisyyteen vaikuttava tekijä [Iivari, 1996; Iivari ja Maansaari, 1997]. Vapaaehtoisuudella ymmärretään tässä yhteydessä sitä, tuntevatko käyttäjät olevansa vapaita valitsemaan työskentelyvälineensä vai pakotetaanko heidät korkeamman johdon taholta käyttämään jotain tiettyä työkalua. Vapaaehtoisuus on kuitenkin mielestäni käsitteenä vaarallinen ja epämääräinen. Onhan luonnollista, että käyttäjät ovat tyytyväisempiä juuri niihin välineisiin, jotka he ovat itse valinneet ja kokeneet itselleen parhaiten sopiviksi. Toisaalta taas organisaatiot tuskin esittelevät kustannussyistä johtuen samanaikaisesti monia CASE-välineitä, joten kovin suurta valinnanvaraa näillä vapaasti valitsemaan päässeillä käyttäjillä tuskin on ollut. Voimme siis perustellusti olettaa, että vapaaehtoisuudella todella on merkitystä käyttäjätyytyväisyyden muodostumisessa.

Iivari [Iivari, 1996] esittelee tutkimuksessaan käsitteen CASE-välineiden käytön ympyrästä. Pääpiirteissään malli esittelee, kuinka tietyt käyttäjien käsitykset CASE-työkaluista näyttävät johtavan tiettyihin jatkokäsityksiin ja lopulta näiden käsityksien sarjasta muodostuu kehä. Iivarin mukaan positiiviset käsitykset CASE-välineiden tehokkuudesta vahvistavat käsityksiä niiden suhteellisesta edusta verrattuna perinteisiin järjestelmiin. Positiiviset käsitykset välineen suhteellisesta edusta Iivarin mukaan taas tuntuvat edistävän CASE-välineiden käyttöä. Kun CASE-välineitä käytetään, käyttäjien mielikuvat niiden tehokkuudesta kasvavat, ja tämän myötä ympyrä sulkeutuu.

Tutkimusten mukaan välineiden käyttö korreloi vahvasti vapaaehtoisuuden kanssa. Kun korrelaatiota mietitään CASE-välineiden käytön ympyrän tarjoamassa kontekstissa, huomataan, että ympyräänhan on jostain tultava sisään. Tällä ajatusmallilla on sikäli suuri merkitys, että pystyttäessä identifioimaan CASE-käytön ympyrään tietty sisääntuloreitti, voidaan suunnitella parempia menetelmiä CASE-välineiden omaksumisen avuksi. Sisääntuloväylä voisi olla näiden tulosten perusteella vapaaehtoisuus. Uudelleen konstruoitu ympyrä on esitetty kuviossa 3.



Kuvio 3. Kehittelemäni uusittu versio Iivarin CASE välineiden käyttö ympyrästä.

En halua millään muotoa väittää, että vapaaehtoisuus olisi ainoa oikea ratkaisu CASE-välineiden käytön menestykselliseen aloittamiseen, vaan haluan esitellä sen olevan yksi mahdollisuus. Koska CASE-välineiden käyttäjätyytyväisyyteen vaikuttaa suuri määrä muuttujia, ei ole koskaan haitaksi tietää useampia menestysreseptejä niiden sokkeloissa kulkemiseen.

Iivarin CASE-käytön ympyrää voidaan myös verrata Banduran [Järvinen ja Järvinen, 1996] sosiaaliseen kognitiiviseen teoriaan. Banduran teorian mukaan kognitiiviset tekijät ja ympäristö vaikuttavat käyttäytymiseen, tässä tapauksessa CASE-välineiden käyttöön. Lisäksi teorian mukaan käyttäytyminen ja ympäristö vaikuttavat ihmisten kognitiivisiin tekijöihin. Käyttäytyminen ja kognitiiviset tekijät taas vaikuttavat ympäristöön. Tästä saadaan aikaan Iivarin CASE-välineiden käytön kehää vastaava rakennelma. Kognitiivisista tekijöistä lisää tutkimuksen kohdassa 4.5, jossa puhutaan ihmisten yksilöllisten ominaisuuksien vaikutuksesta käyttäjätyytyväisyyteen.

Toisaalta Finniganin et al. tulkinnan mukaan vapaaehtoisuus ei välttämättä edesauta käyttäjätyytyväisyyttä, koska silloin suunnittelijat eivät välttämättä edes käytä uusia CASE-välineitä [Finnigan et al., 2000].

Tässä kohdassa siis tarkasteltiin CASE-välineiden käyttäjätyytyväisyyttä vapaaehtoisuuden kannalta. Vapaaehtoisuuden katsottiin olevan merkittävä tekijä käyttäjätyytyväisyyden muodostumisessa. Iivarin CASE-välineiden kehästä esiteltiin uusittu versio, jossa vapaaehtoisuus toimi sisääntuloreittinä ympyrään, ja näin sysäyksenä CASE-välineiden menestykselliselle käytölle ja käyttäjätyytyväisyyden muodostumiselle.

4.3.3 Organisaation rakenteelliset elementit

CASE-välineiden käyttäjätyytyväisyyttä arvioitaessa on syytä ottaa huomioon myös yrityksen tai organisaation rakenteelliset elementit, kuten palkitsemiskäytännöt, menettelytavat ja muut työskentelyyn liittyvät käytännöt. Aaen [Aaen, 1993] toteaa, että yrityksen koolla on merkitystä käyttäjien arvioihin CASE-välineistä. Aaen myös esittää, että yrityksissä, joissa CASE-välineen esittely sidotaan vastaavaan organisaation rakenteelliseen muutokseen, käyttäjät tuntuvat arvioivan CASE-välineitä positiivisemmin.

Tämä tutkimus ei huomioi normeja, jotka vaikuttavat ihmisten jokapäiväiseen vuorovaikutukseen. Päätin jättää normien tutkimisen tämän työn ulkopuolelle, koska ne eivät varsinaisesti kuulu aihealueeseen ja niiden tarkastelu vaatisi pitkällistä perehtymistä moneen eri tieteenalaan. Myös vuorovaikutuksella on myös oma vaikutuksensa normien muodostumisessa. Orlikowski pohtii näiden tekijöiden vaikutusta ryhmäohjelmien kohdalla [Orlikowski, 1992], mutta nämä seikat ovat yleistettävissä yhtä hyvin myös CASE-välineisiin.

4.3.4 Johdon tuki

Sumner [Sumner, 1995] toteaa, että johtajien tulee ymmärtää kaikki CASE-välineisiin liittyvät seikat perusteellisesti, jotta niiden käyttöönotosta tulisi menestyksellinen. Kuten kaikissa yrityksessä tapahtuvissa toiminnoissa, tulee myös CASE-välineiden käytöllä olla täysi ylimmän johdon tuki, jotta niitä voitaisiin menestyksellisesti käyttää. Menestyksellinen

käyttö puolestaan parantaa käyttäjätyytyväisyyttä ja käyttäjien arvioita välineestä. Myös Aenin [Aen, 1993] tutkimuksesta ilmenee, että johdon sitoutuneisuus on merkittävä seikka CASE-välineiden tuottamisissa vaikutuksissa yrityksessä.

Tämän kohdan päätarkoituksena oli tuoda esiin se, että käyttäjien tyytyväisyys välineiden käyttöön ei ole pelkästään riippuvainen välineen ominaisuuksista, vaan myös ympäristöllä on suuri vaikutus asiaan. Yksittäisistä tekijöistä voitaisiin nostaa esiin ehkä tärkeimpänä käyttäjien tarpeita vastaava koulutus. Kuitenkin on muistettava, että yhtä hyvin vääränlaisella koulutuksella voidaan heikentää tyytyväisyyttä hyvänkin välineen käytössä. Myös vapaaehtoisuus saattaa monissa tapauksissa olla merkittävä tekijä käyttäjätyytyväisyyden muodostumisessa.

4.4 CASE-välineiden ominaisuuksien vaikutus käyttäjätyytyväisyyteen

Perinteisesti CASE-välineiden tutkimusalueella suosituimpia tutkimuskohteita ovat olleet juuri yksittäisten välineiden ominaisuudet ja niiden arviointi. Esimerkiksi Crozier et al. luovat kriittisen katsauksen kolmen suosituksen välineen ominaisuuksiin [Crozier et al., 1989]. Nämä tutkimukset ovatkin hyödyllisiä niissä tapauksissa, joissa tarvitaan tietoa yksittäisistä välineistä tai niiden ominaisuuksista. Kuitenkin pidemmällä ajanjaksolla tarvitaan tietoa CASE-työkalujen yleisistä suuntaviivoista hahmottelemaan kokonaiskuvaa alan yleisestä kehityksestä.

CASE-välineitä voidaan käyttää hyväksi sekä ohjelmoinnissa että laajemman mittakaavan ohjelmistosuunnittelussa. Rajanveto näiden kahden eri asian välillä ei aina ole selvä. Mielestäni CASE-välineiden perimmäinen tarkoitus on kuitenkin tarjota laajempaa tukea ohjelmistosuunnittelun eri vaiheille. Tämä tarkoittaa käytännössä sitä, että ohjelmointi luetaan tämän tutkimuksen yhteydessä ohjelmistosuunnittelun osajoukoksi sen muiden osa-alueiden lisäksi.

Seuraavissa alakohdissa tarkoitukseni on tarkastella yleisesti CASE-välineiden ominaisuuksia käyttäjätyytyväisyyden kannalta. Tämän tutkimuksen tarkoituksenaan oli selvittää, mitkä tekijät vaikuttavat käyttäjätyytyväisyyden syntymiseen. Tässä kohdassa siis

kysymys on siitä, mitkä välineen ominaisuudet vaikuttavat käyttäjätyytyväisyyteen positiivisesti ja mitkä negatiivisesti. Ensin tarkastellaan välineiden kompleksisuutta, sitten pohditaan niiden yhteensopivuutta muiden välineiden kanssa ja sen jälkeen käyttöliittymän merkitystä. Lopuksi arvioidaan CASE-välineiden rakentumista käyttäjän tarpeiden mukaisiksi sekä niiden uudelleenkäyttömahdollisuuksia.

4.4.1 Kompleksisuus

Iivarin tutkimuksen mukaan CASE-välineet ovat usein liian monimutkaisia käyttää ja niissä on liikaa toimintoja. Tämän vuoksi käyttäjien on vaikea arvostaa niistä saatavia hyötyjä [Iivari, 1996]. Monimutkaisuuden kokemista voidaan huomattavasti helpottaa ja vähentää välineen käyttöönottovaiheessa antamalla esimerkiksi laadukasta välineen käyttöön liittyvää koulutusta. Kuten edellisessä kohdassa jo todettiin, tämän koulutuksen sisältö on tarkoin valittava käyttäjien tarpeiden mukaan, jotta koulutukselliset tavoitteet olisi mahdollista saavuttaa. On myös tärkeää huomioida kuulijoiden lähtötaso, jotta ei aiheutettaisi lisäharmia niille käyttäjille, joilla on jo vaikeuksia ymmärtää välineen perusominaisuuksiakin.

4.4.2 Käyttöliittymä

Jarabekin ja Huangin [Jarabek ja Huang, 1998] mukaan hyvän käyttöliittymän pitäisi tuottaa metafora, joka sulkee käsitteellisen kuilun ihmisen ajattelun ja tietokonejärjestelmän väliltä. Kuitenkin nykyisten välineiden käyttöliittymiä on rakennettu hyvin vähän tai ei ollenkaan tätä vaatimusta silmälläpitäen. Käyttöliittymän tulisi myös antaa ohjeistusta enemmän aloittelevalle kuin kokeneemmalle käyttäjälle. Jarabekin ja Huangin mukaan kuitenkin tämänkään tapaista käyttöliittymän räätälöintiä ei ole kovinkaan monessa työkalussa saatavilla.

CASE-välineiden ominaisuuksien määrän kasvaessa niiden käyttöliittymätkin muuttuvat yhä monimutkaisimmiksi ja erilaisten kontrollien määrä ruudulla kasvaa. Tunnettujen testien mukaan ihmisen käsityskyky kuitenkin on hyvin rajoittunut (7 + 2 periaate), joten kontrollien suuri määrä on kerralla liikaa tottuneemmallekin käyttäjälle. Onkin ehdotettu, että työkaluihin istutettaisiin aktiivisia toimintoja, jotka tarkkailisivat käyttäjää esimerkiksi Microsoftin 2000

tuoteperheen lailla. Nämä toiminnot piilottaisivat vähän käytettyjä kontrolleja ja toisivat esiin niitä, joita käyttäjät eniten tarvitsevat [Dabin, 1991; Jarzabek ja Huang, 1998].

Tutkimusten mukaan käyttöliittymän tulisi myös tukea paremmin käyttäjien mentaalisia (kognitiivisia) malleja, jotta suunnittelijan ajatukset voitaisiin tarkemmin esittää fyysisinä objekteina CASE-työkalun näkökulmasta katsottuna [Budgen et al., 1993]. Kusters ja Wijersin tutkimuksen mukaan käyttöliittymä ihmisen ja tietokoneen välillä kuuluu viiden CASE-välineiden tärkeimmän valintakriteerin joukkoon [Kusters ja Wijers, 1993]. Tutkimus on vuodelta 1993 ja käsitteli hollantilaisia CASE-välineiden käyttäjiä, ja suuren otoksen vuoksi voidaan siitä mielestäni vetää yleisempiäkin johtopäätöksiä. Vessey et al. ehdottavat tutkimuksessaan suunnittelijoiden avuksi eräänlaista sopeutuvaa apujärjestelmää, joka käyttäjän taitotason mukaan tarjoaisi erilaisia ohjeita käsillä olevan tehtävän ratkaisemiseksi. [Vessey et al., 1992]

Käyttöliittymästä ja sen käytettävyydestä puhutaan nykyään paljon, mutta hyvän käyttöliittymän määritelmä jää usein hämärän peittoon. Tässä tutkimuksessa puhutaan erityisesti käyttäjätyytyväisyydestä ja jos käyttöliittymän käytettävyys on hyvä, ovat käyttäjätkin yleensä tyytyväisiä. Käytettävyyteen tämä tutkimus ei juurikaan ota kantaa, sillä ohjelman käytettävyyden arviointiin tarvitaan aivan erilaisia lähestymistapoja [Nielsen, 1993]. Käytettävyyttä voidaan luonnehtia IEEE:n määrittelemään tapaan [IEEE, 1990], vapaasti suomennettuna: "Helppous, jolla käyttäjä oppii operoimaan järjestelmää tai komponenttia, antamaan sille syötteitä ja tulkitsemaan sen antamia vasteita." Käytettävyyttä voidaan myös määritellä ISO 9241:n näkökulmasta [ISO, 2001]: "Hyödyllisyys, tehokkuus ja tyytyväisyys, jolla käyttäjä saavuttaa määritellyt tavoitteet tietyissä järjestelmissä." Näistä jälkimmäinen tietysti sopii paremmin tähän tutkimukseen, koska siinä mainitaan tyytyväisyys.

Tämän kohdan tarkoituksena oli tuoda esiin käyttöliittymän tärkeys niin CASE-välineissä kuin kaikissa muissakin ohjelmistoissa. Tutkimusten mukaan käyttöliittymän vaikutus CASE-välineiden käyttäjätyytyväisyyteen on merkittävä. Nämä työkalut ovat kuitenkin toiminnaltaan niin monimutkaisia, että hyvän käyttöliittymän suunnittelu tarjoa mittavia haasteita. Erityisesti huolta aiheuttaa käyttöliittymän sopivuus niin vasta-alkajille kuin kokeneille ohjelmoijillekin. Tässä onkin varmasti lisätutkimuksen paikka selvittää, miten CASE-välineiden käyttöliittymät eroavat tavallisten ohjelmien käyttöliittymistä ja miten niitä voitaisiin parantaa.

4.4.3 Menetelmäkeskeinen rakenne

Tässä kohdassa selvitetään CASE-välineiden perustana olevan suunnittelumetodologian ja -menetelmien merkitystä käyttäjätyytyväisyyden muodostumisessa. Mielipiteet kirjallisuudessa jakaantuvat selvästi kahteen eri leiriin, joista toisen mielestä CASE-välineiden rakenteen tulisi seurata hyvin läheisesti jotain tiettyä suunnittelumenetelmää. Tämän avulla voitaisiin varmistaa käyttäjien tyytyväisyys, koska väline antaisi tukea jokaiselle suoritetulle työvaiheelle. Toiset puolestaan ovat sitä mieltä, että tietyn suunnittelumenetelmän seuraaminen rajoittaa suunnittelijoiden mahdollisuuksia ja sen vuoksi CASE-välineiden tulisi tukea paremmin suunnittelijoiden yksilöllisiä taipumuksia.

Jarzabekin ja Huangin tutkimusten [Jarzabek ja Huang, 1998] mukaan CASE-työkalut perustuvat vielä nykyisinkin ajatukseen, jossa työkalun on tuettava vain ohjelmistokehityksen menetelmiä. Heidän mukaansa välineissä on vähän tai hyvin vähän tukea ohjelmistokehityksen pehmeämmille osa-alueille. Heidän tutkimuksensa mukaan seuraavat ohjelmistokehityksen pehmeät elementit puuttuvat useista CASE-välineistä:

- luovuus
- ymmärtäminen
- ideoiden luominen
- rajoittamaton ajattelutapa
- ihmismielen tilat (virkeys, väsymys, tunnetilat)

King [Spurr ja Layzell, 1992] toteaa, että CASE-välineistä tuntuu puuttuvan asiakas-orientoitunut näkökulma. Välineet tukevat funktionaalisesti oikein toimivan järjestelmän tuottamista, mutta eivät ota huomioon laatuun liittyviä tekijöitä. Tämä on sikäli ristiriitaista, että useissa tutkimuksissa on todettu laadun päinvastoin kasvavan CASE-välineiden käytön myötä [Jones, 1992; Low ja Leenanuraksa, 1999; Stobart et al., 1993]. King kuitenkin tarkoittaa laatua asiakkaan näkökulmasta, mikä ei aina ole sama asia kuin virheetön ohjelmakoodi. Tämä tuntuukin paljon järkevämmältä, sillä myös Jarzabek ja Huang valittavat välineiden menetelmäkeskeisyyttä.

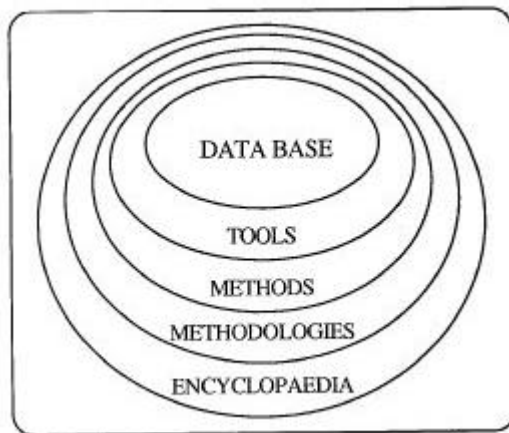
Olen samaa mieltä siitä, että CASE-työkaluista puuttuu todellakin monia tärkeitä ominaisuuksia, jotka voisivat toimiessaan vaikuttaa käyttäjäytyväisyyteen positiivisesti. En kuitenkaan olisi niinkään varma siitä, onko työkaluihin hyvä sisällyttää Jarzabekin ja Huangin ehdottamaa tukea käyttäjien eri tunnetiloille. Ensinnäkään tämänkaltaisia ominaisuuksia olisi vaikea toteuttaa kaikille käyttäjille sopiviksi, koska ihmiset ovat psyykkisiltä ja fyysisiltä ominaisuuksiltaan varsin erilaisia. Toiseksi tutkimusta ihmisten tunteiden liittymisestä tietokoneen käyttöön ei ole tehty tarpeeksi. Kolmanneksi jotkut käyttäjät voisivat tuntea tällaisen välineen liian tungettelevaksi ja yksityisyyttä loukkaavaksi. Kuitenkin esimerkiksi IBM on omistanut yhden projekteistaan ihmisten tunteiden tunnistamiseen liittyvään tutkimukseen. Tästä BlueEyes-projektista voikin olla tulevaisuudessa apua tunteiden yhdistämisessä käyttöliittymään [IBM, 2000].

On kuitenkin muistettava, että menetelmäkeskeisyys ja tuki olemassa oleville prosesseille eivät välttämättä ole lainkaan huono asia. Griffiths jopa ehdottaa tutkimuksessaan [Griffiths, 1994] entistä tiukempaa työkalujen integrointia, jotta voitaisiin muodostaa täysin ohjelmistotuotannon menetelmiä ja prosesseja vastaava työkalu. Uutena ideana Griffiths ehdottaa työkalun integroinnin ulottamista myös MIS-järjestelmiin saakka. Ilman tätä integrointia ei Griffithsin mukaan ohjelmistotuotannon projekteja voida hallita tehokkaasti. Mielestäni asia ei kuitenkaan ole näin yksinkertainen, sillä tietotekniikka-alan jatkuvasti muuttuvassa toimintaympäristössä kuvatus kaltainen järjestelmä olisi jo vanhentunut käyttöönnotossa, sillä menetelmät ja prosessit vaihtuvat hyvin tiuhaan tahtiin. Vanhoihin menetelmiin ja käytäntöihin perustuva väline tuskin saavuttaisi kovinkaan suurta menestystä käyttäjäytyväisyyden suhteen. Kuitenkin myös Stobart et al. ovat päätyneet tutkimuksissaan siihen tosiseikkaan, että CASE-välineiden ominaisuuksia pitäisi laajentaa johdon tukijärjestelmien suuntaan [Stobart et al., 1993].

Myös Nosek et al. ovat tutkimuksessaan [Nosek et al., 1992] sitä mieltä, että ylemmän tason CASE-välineet tarvitsevat jonkin metodologian ollakseen toimivia (CASE-välineiden luokista enemmän kohdassa 5.4). He kuitenkin haluavat muistuttaa, että tämä metodologia voi vaikeuttaa normaalia suunnittelijan ajatusprosessia, minkä vuoksi he ehdottavat uusien syöttövälineiden, kuten valokynän ja digitointipöydän hyväksikäyttöä kaavioiden piirtämisessä.

Vessey et al. ovat tutkineet eri CASE-välineitä niiden käyttämien metodologioiden kannalta. He ovat tutkimuksessaan havainneet, että liian tiukka ja suora metodologian seuraaminen saattaa aiheuttaa ongelmia välineen hyväksynnässä kokeneiden suunnittelijoiden taholta. Nämä omiin suunnittelutottumuksiinsa luottavat ammattilaiset eivät helposti hyväksy välinettä, joka pakottaa heidät luopumaan omista hyviksi havaituista menetelmistään. [Vessey et al., 1992] Vessey et al. ovat mielestäni ansiokkaasti tutkineet CASE-välineitä, mutta en ole heidän kanssaan samaa mieltä siitä, että CASE-välineiden tuottamat hyödyt olisivat olleet tutkimusten mukaan minimaalisia. Tässä tutkimuksessa esitellyt lähteet ovat kiistatta todistaneet CASE-välineiden tarjoamat edut, mutta aina on huomioitava se, että nämä edut eivät realisoidu itsestään vaan niiden saavuttamiseksi on tehtävä paljon työtä niin valinnan, käyttöönoton kuin koulutuksenkin saralla.

Goldstein kuuluttaa artikkelissaan [Goldstein, 1990] CASE-välineiden "puuttuvan renkaan", menetelmäkeskeisyyden perään. Kuviossa 4 on esitetty Goldsteinin malli CASE-välineen perustaksi.



Kuvio 4. Goldsteinin ehdotus CASE-välineen rakenteeksi.

Goldsteinin mukaan CASE-välineen ydin on tietokanta, joka sisältää keskitetysti kaiken projektiin liittyvän informaation mukaan lukien tekstit, kuvat, datan, terminologian ja ylipäätään kaiken tiedon, joka on kerätty projektin elinkaaren aikana. Tämän tietokannan tarkoituksena on erityisesti palvella dokumentoinnin asettamia vaatimuksia. Näitä tietoja käsitellään ja päivitetään puolestaan mallin seuraavalla kehällä olevilla kehitystyökaluilla ja -välineillä. Mallin seuraavan kehät metodit määrittelevät ne tavat, joilla työkaluja voidaan käyttää tietokannan tietojen käsittelyssä. Metodologiat puolestaan keräävät yhteen ne

prosessit ja menettelytavat, jotka on määritelty edellisellä kehällä. Mallin uloin osa kuvaa tietovarastoa, josta voidaan valita edellisiä kehiä määrittävät kokonaisvaltaiset tuotekehitysmallit.

Aluksi Goldsteinin malli vaikuttaa melko monimutkaiselta kaikkine eri tasoineen, mutta hetken tutkiskelun jälkeen selviää, että se itse asiassa on ainoastaan formalisoitu kuvaus kaikesta edellä kuvaamistani menetelmäkeskeisistä malleista. Tämän vuoksi halusin tuoda esiin sen tässä yhteydessä eräänlaisena yhteenvetona tässä kohdassa käsittelemistäni asioista.

Näissä kappaleissa olen yrittänyt luoda silmäyksen käyttäjätuottavuuden ja välineiden menetelmäkeskeiseen rakenteeseen. Toisaalta hyvin yrityksissä käytössä olevia menetelmiä tukeva CASE-väline melko varmasti indikoi tuottavampia käyttäjiä, mutta toisaalta taas välineen tulisi tukea ohjelmistokehityksen pehmeämpiä arvoja, kuten luovuutta ja yksilöllisyyttä.

4.4.4 Uudelleenkäyttö ja yhteensopivuus

Yksi CASE-työkalujen mainostetuimmista ominaisuuksista on uudelleenkäytön tehostaminen. Seuraavissa kappaleissa esitellään muutamia uudelleenkäytön mahdollisuuksia ja niiden toteuttamista koskevia seikkoja puhuttaessa CASE-välineistä.

Nykyvälineet eivät Jonesin mukaan toteuta uudelleenkäyttöä riittävän korkealla tasolla käyttäjätuottavuuden saavuttamiseksi. Jones sanoo, että CASE-työkalujen yhdistäminen olioperustaisiin ohjelmointimenetelmiin johtaa merkittävään uudelleenkäytön paranemiseen [Jones, 1992].

Mili et al. luokittelevat uudelleenkäytettävyyden kahteen lähestymistapaan: komponentti- ja generatiiviseen lähestymistapaan [Mili et al., 1995]. CASE-välineiden kohdalla kyseeseen voisi tulla lähinnä komponenttipohjainen lähestymistapa. Tavoitteena voisikin olla malli, jossa käyttäjän ei ole tarpeen keksiä yhä uudelleen ratkaisuja perusongelmiin. CASE-väline tarjoaisi laajan valikoiman malliratkaisuja kohdealueen yleisimpiin ongelmiin, ja suunnittelija voisi keskittyä todellisten ratkaisemattomien ongelmien ratkaisemiseen.

Dabinin mukaan [Dabin, 1991] CASE-työkalut tulisi nähdä pieninä ohjelmistotehtaina, jotka tarjoaisivat mahdollisuuden uudelleenkäyttöön. Jotta ajatus ohjelmistotehtaasta toteutuisi, tulisi CASE-työkalujen olla projektin ohella myös koko organisaation työkaluja. CASE-välineiden olisi myös oltava yhteensopivia jo olemassa olevien muiden ohjelmistojen kanssa. Yhteensopivuus aikaisempien järjestelmien kanssa on tärkeää, jotta saataisiin uudelleenkäytettyä jo aikaisemmin tehtyjä ratkaisuja. Käyttäjätyytyväisyyden voi olettaa kasvavan, jos jo kerran tehdyt suunnitelmat voidaan siirtää uuteen järjestelmään ilman mainittavia lisäponnisteluja. Myös Stobart et al. [Stobart et al., 1993] esittävät artikkelissaan, että käyttäjien mielestä välineet eivät tue tarpeeksi eri työvaiheissa käytettävien työkalujen integrointia.

Coallier asettaa artikkelissaan varsin kunnianhimoisia päämääriä CASE-välineille [Coallier, 1990]. Hänen mielestään CASE-työkalujen tulisi tulevaisuudessa täyttää tietyt standardoidut vaatimukset, joita ovat:

- Erityisien standardoitujen apuvälineiden luominen tekstipohjaisten alemman tason CASE-välineiden käyttöliittymän parantamiseksi
- CASE-välineiden käyttämien graafisten ja ei-graafisten yksiköiden ja objektien standardointi paremman tiedonsiirron varmistamiseksi eri välineiden välillä
- CASE-välineiden käyttämän tiedon formalisointi, jotta se olisi helposti käsiteltävissä henkilökohtaisilla tietokoneilla

Kuten jo sanoin, nämä Coallierin esittämät vaateet ovat varsin kunnianhimoisia siinä suhteessa, että nykyisin CASE-välineitä on valtavan paljon ja likipitäen jokaisella niistä on omat tallennusmuotonsa. Coallierin kirjoittama artikkeli on vuodelta 1990 ja näin reilun vuosikymmenen jälkeen edelleenkin ei ole nähtävissä merkkejä hänen mainitsemistaan standardointipyrkimyksistä. Tämä ei kuitenkaan mielestäni tarkoita sitä, että hänen esittämänsä ajatukset olisivat vanhentuneita tai muutenkaan epäkelpoja. Henkilökohtaisesti näen asian niin, että jos kyseisen kaltainen standardointimenettely otettaisiin käyttöön laajassa mittakaavassa, voitaisiin hyvin nopeasti päästä eroon CASE-välineitä vielä nykyäänkin vaivaavista yhteensopivuusongelmista. On kuitenkin totta, että tällainen projekti vaatisi varmasti niin rahallisesti kuin työmäärällisestikin varsin mittavasti resursseja. Kuitenkin mielestäni lopputulos olisi varmasti kaiken vaivannäön arvoinen. Coallier myös esittelee artikkelinsa loppuosassa varsin yksityiskohtaisesti eri standardeja, joihin tulevan CASE-

standardin tulisi pohjautua. Näin vuosikymmen eteenpäin nämä ehdotukset eivät enää tunnu kovinkaan hyviltä vaihtoehtoilta, joten idean toteuttamiseksi tarvittaisiin lisäkartoitusta nykyisten standardien piirissä.

Brown esittelee artikkelissaan käsitteen CASE-ympäristöistä, joka tarkoittavat yksittäisistä CASE-välineistä koottuja järjestelmiä, jotka on sovitettu toimimaan yhdessä joko räätälöinnin tai välittäjäohjelmien avulla. Hänen mielestään ongelmat, jotka aiheutuvat näiden välineiden asteittaisesta hankinnasta ja yhteensovittamisesta ovat suurimpia taakkoja välineiden käyttäjille. [Brown, 1994]

Brownin artikkeli käsittelee etupäässä sitä, miten CASE-ympäristöjen hankintaa voidaan erilaisten menetelmien avulla helpottaa, mutta olen hänen kanssaan samaa mieltä taakasta, joka suunnittelijoille aiheutuu epäyhteensopivista välineistä. Käyttäjien tyytyväisyys hyviinkin välineisiin laskee, jos tiedonsiirto niiden välillä vaatii mittavia manuaalisia, yksitoikkoisia tai työteliäitä toimenpiteitä.

Yhteensopivuutta voidaan ajatella myös hieman toisella tavalla, kuten esimerkiksi Sumner [Sumner, 1995] artikkelissaan tekee. Sumner esittelee tutkimuksessaan korkean sisäisen integraation käsitteen. Tämä käsite pitää sisällään sisäisen johdon tuen ja kypsän suunnittelu-ympäristön ohella myös työntekijöiden olemassa olevien kykyjen yhteensopivuuden uuden järjestelmän kanssa. Suuri sisäinen integraatio on yleensä indikaatio korkeasta onnistumisprosentista CASE-välineiden käytössä. Varsinaisestihan onnistuminen ei vielä kerro käyttäjätyytyväisyydestä, mutta kuten Iivari toteaa [Iivari, 1996], onnistumiset ja positiiviset käsitykset välineestä näyttävät johtavan suurempaan käyttäjätyytyväisyyteen. Yhteensopivuutta voidaan tietysti parantaa koulutuksen avulla, josta tarkemmin kohdassa 4.3.1.

Tarvetta uudelleenkäytölle on siis myöskin CASE-välineiden piirissä, mutta kuten Lee toteaa [Lee, 1993], ohjelmiston uudelleenkäyttö on vain rajoitettu ja vähemmän joustava muoto tiedon uudelleenkäytöstä. Sen vuoksi ohjelmistokomponenttien uudelleenkäyttöön tarvitaan paljon ylimääräistä tietoa, jota ei ole talletettuna itse uudelleenkäytettävään ohjelmiston osaan. Yksi CASE-työkalujen haasteista onkin kehittää parempia menetelmiä kaiken ohjelmointiprosessissa tarvittavan tiedon vangitsemiseen. Tällöin rajanveto

tietämyksenhallinta- ja CASE-välineiden välillä saattaa merkittävästi hämärtyä. Tämä on kuitenkin aihe erikseen ja lisätutkimusta täältä alalta tarvitaan.

4.4.5 Rääätälöitävyys ja tuki monelle käyttäjälle

Rääätälöitävyys tarkoittaa ohjelmiston ominaisuutta, joka mahdollistaa tiettyjen ohjelmiston ominaisuuksien muuttamisen erilaisten käyttäjäryhmien tarpeita vastaaviksi. Stiemerling et al. [Stiemerling et al., 1997] toteavat myös rääätälöitävyyden tarpeelliseksi jatkuvasti muuttuvien ja erilaistuvien työympäristöjen sekä käyttäjien tarpeiden vuoksi. Erityisesti rääätälöitävyyttä tarvitaan heidän mukaansa ryhmäohjelmien tapauksessa.

Finnigan et al. [Finnigan et al., 2000] puhuvat artikkelissaan siitä, kuinka CASE-välineiden tulisi mahdollistaa suunnittelijoille uusimpien tekniikoiden ja trendien seuraaminen. Tällä tavoin suunnittelijat voisivat rääätälöidä välinettään tukemaan yhä uudempia ja tehokkaampia työskentelymenetelmiä.

Stobartin et al. [Stobart et al., 1993] saamien tulosten mukaan käyttäjät ovat sitä mieltä, että CASE-välineet eivät ole tarpeeksi helposti muokattavissa organisaation tarpeita vastaaviksi. Lisäksi välineet tukevat usein vain yhtä kehitysmenetelmää, joka on usein erilainen kuin organisaatiossa jo käytössä olevat menetelmät. Stobart et al. esittävät myös artikkelinsa tutkimustuloksien perusteella, että käyttäjien mielestä CASE-työkalujen tulisi sisältää tukea useammalle käyttäjälle.

Ohjelmistoprojektien hallinta on nykypäivän suurien projektien myötä tullut yhä vaikeammaksi. Tätä mieltä on myös Griffiths, joka tutkimuksessaan ehdottaa ratkaisuiksi monen käyttäjän CASE-järjestelmiä, joiden käyttäjinä olisivat kaikki projektin työntekijät alimman tason ohjelmoijasta ylimpään johtoon saakka [Griffiths, 1994]. Sorensenin mukaan [Sorensen, 1995] suuri osa ohjelmoijan ajasta kuluu erilaisiin koordinoititehtäviin. Apuna tähän Sorensen esittää lyhyehkössä, mutta ytimekkäässä tutkimuksessaan Griffithsin tapaan monen käyttäjän järjestelmiä, joissa koordinoinnille olisi tuki jo luonnostaan.

Käyttäjätyytyväisyyttä epäilemättä lisäksi edellä kuvatun kaltaisten ominaisuuksien integroiminen CASE-välineisiin. Olen kuitenkin itse sitä mieltä, että näiden lisäyksien suorittamisessa tulisi noudattaa suurta varovaisuutta, jotta ei pudottaisi monimutkaisuuden sudenkuoppaan, jonka puolestaan tiedämme olennaisesti laskevan käyttäjien tyytyväisyyttä CASE-välineisiin.

Edellisten kohtien perusteella voidaan siis edelleenkin olla sitä mieltä, että on syytä tutkia eri CASE-välineiden ominaisuuksia, jotta voitaisiin muokata niitä paremmin käyttäjien tarpeita vastaaviksi. Erityisesti CASE-välineiden suunnittelussa olisi mielestäni keskityttävä käyttäjakeskeiseen lähestymistapaan. Tietysti erilaisia metodeja ja prosesseja kuvaavat lähestymistavat voivat olla tehokkaampia ja tuntuvat paremmilta ratkaisuilta. Kuitenkin on muistettava, että viime kädessä päätökset tekee ihminen ja ihmiset ovat usein varsin uskollisia työtavoilleen ja -tottumuksilleen. Sen vuoksi jokaisen CASE-välineen tulisi olla siinä määrin räätälöitävissä ja muokattavissa, että käyttäjät voisivat tuntea olonsa kotoisaksi sitä käyttäessään. Räätälöitävyys on myös eräs mahdollisuus Sumnerin [Sumner, 1995] mainitsemaan yhteensopivuuteen, jolloin uusi väline toimisi käyttäjien entisten kokemusten mukaisesti. Tämä puolestaan auttaisi CASE-välineen menestyksellistä käyttöä ja liivarin CASE-välineiden käytön ympyrän mukaisesti parantaisi käyttäjätyytyväisyyttä.

4.5 CASE-välineet ja yksilön ominaisuuksien vaikutus käyttäjätyytyväisyyteen

Seuraavissa kohdissa esitellään erilaisia yksilöiden ominaisuuksiin tekijöitä, joilla voi olla tutkimusten mukaan vaikutusta käyttäjien tyytyväisyyteen CASE-välineiden käytössä. Lista ei pyri olemaan täydellinen, mutta se yrittää herätellä huomaamaan kuinka erilaisia käsityksiä ja asenteita eri ihmisillä voikaan olla.

Aenin tutkimustulosten mukaan [Aen, 1993] käyttäjät antavat CASE-välineistä parempia arvioita, jos sen vasteaika on lyhyt. Tällä hän tarkoittaa sitä, että käyttäjät, jotka painottavat vasteajan merkitystä saattavat helposti antaa positiivisia arvioita välineistä niiden huonoista puolista riippumatta.

Orlikowskin artikkeli [Orlikowski, 1988] identifioi erilaisten ihmistyyppien asennoitumista uuteen CASE-järjestelmään ja sen käyttämiseen. Päinvastoin kuin johtavassa asemassa olevat työntekijät, yksittäiset käyttäjät eivät niinkään ole kiinnostuneita siitä, kuinka paljon jokin tietty työkalu tai väline nopeuttaa koko projektin valmistumista. Sen sijaan mielenkiinto kohdistuu heidän omiin henkilökohtaisiin päämääriinsä ja niiden täyttämiseen. Jos CASE-työkalut eivät tunnu täyttävän heidän suunnitelmiaan pitkällä tai edes lyhyelläkään aikavälillä, on heidän asennoitumisensa välineitä kohtaan negatiivinen.

Jotkut käyttäjät voivat kokea asiantuntija-asemansa heikentyvän CASE-välineiden tulon myötä, koska uusien välineiden myötä heidän vaivalla hankittu ammattitaitonsa muuttuu ikään kuin hyödyttömäksi [Orlikowski, 1988; Sumner, 1995]. Nämä käyttäjät saattavat myös pelätä, että heidän osaamisensa markkinoitavuus vähenee CASE-työkalujen pitkälle kehittyneiden automaatio-ominaisuuksien vuoksi.

Automaatio CASE-välineissä saattaa olla hyvä asia tehokkuuden kannalta, mutta jotkut käyttäjät saattavat tuntea sen myös aliarvioivan heidän kykyjään. Pettymyksiä saattaa tulla myös sellaisille käyttäjille, jotka odottavat saavansa oppia jotain uutta ja hienostunutta, mutta huomaavatkin pitkälle kehittyneen automatisoinnin tekevän suuren osan työstä heidän puolestaan. On kuitenkin olemassa sellainenkin käyttäjäryhmä, joka ajattelee enemmän suurten linjausten asioita. Nämä käyttäjät ovat tyytyväisiä automaation lisääntymiseen, koska se lisää heidän tuottavuuttaan ja vapauttaa heidät monimutkaisten asioiden näpertelystä.

Orlikowskin tutkimuksessa esiintyvät asennetyypit kuvastavat omien kokemusteni pohjalta melkoisen hyvin sitä persoonallisuuksien kirjoa, joka ohjelmistoalalla vallitsee. On kuitenkin muistettava, että edellä esitetyt ongelmaryhmät eivät ole enemmistönä CASE-välineiden käyttäjäkunnassa ja että suurin osa käyttäjistä suhtautuu enemmän tai vähemmän neutraalilla tavalla CASE-työkalujen asettamiin uusiin vaatimuksiin heidän osaamisessaan. Totta kuitenkin on, että valtavaa vauhtia kehittyvä ohjelmistoala kaipaa jatkuvasti enemmän osaajia ja selvää myöskin on, että heitä ei loputtomasti ole tarjolla [Dabin, 1991; Kemerer, 1992]. Koska CASE-välineiden käyttöön riittää lyhyempikin koulutus, on mielestäni perusteltua väittää, että CASE-välineet työvoiman tarvetta helpottaessaan saattavat asettaa vanhojen työntekijöiden vaivalla hankitut taidot kyseenalaiseen tilanteeseen.

Orlikowski [Orlikowski, 1992] on kehittänyt teknologisten kehysten käsitteen kuvaamaan ihmisten kognitiivisten elementtien vaikutusta uusien välineiden käyttöönottamiseen. Teknologisilla kehyksillä hän ymmärtää ihmisten mentaalisia malleja organisaation, työn ja ympäristön koostumuksesta. Hänen mukaansa nämä mallit ovat hyvin yksilöllisiä, mutta usein ne ovat hyvin samankaltaisia samanlaisessa työympäristössä työskentelevillä ihmisillä, jotka ovat saaneet samanlaisen koulutuksen ja ovat päivittäin toistensa kanssa vuorovaikutuksessa. Kohdatessaan uutta teknologiaa ihmiset pyrkivät ymmärtämään sitä omien teknologisten kehystensä avulla, usein pyrkien laajentamaan kehyksiä kattamaan uuden teknologian erityispiirteitä. Kuitenkin jos uuden teknologian ja olemassa olevien kehysten ero on liian suuri, saattaa vanha teknologinen kehys vaikeuttaa uuden teknologian ymmärtämistä ja ihmisten on muutettava radikaalisti omia käsityksiään. Tämä teknologisten kehyksien muuttaminen riippuu Orlikowskin mielestä toisaalta siitä, miten paljon tietoa tarjotaan saataville uudesta välineestä, ja toisaalta siitä, minkälaista koulutusta välineen käyttöön annetaan. Vaikka Orlikowskin artikkeli kosketteleekin ryhmäohjelmia, voidaan hänen tuloksensa helposti ulottaa koskemaan edellisten kohtien perusteella myös CASE-työkaluja.

Eräs huomioonotettava asia on Vessey et al. mukaan se, että tiukkaan metodologiseen perustaan nojaavat CASE-välineet eivät saavuta suosiota omiin metodeihinsa tottuneille kokeneille järjestelmäsuunnittelijoille. Heidän mielestään tällaisista välineistä saavat kaikkein eniten hyötyä juuri alalle tulleet käyttäjät, joilla ei ole kovinkaan paljon kokemusta CASE-välineistä.[Vessey et al., 1992]

Edellisissä kohdissa tarkoitukseni oli esittää perusteluja sen puolesta, että välineen ominaisuuksien ja ympäristövaikutusten ohella myös käyttäjien henkilökohtaisilla persoonallisuuksilla ja asenteilla on merkitystä. Tietenkin voidaan keskustella siitä, kuinka suuri tämä vaikutus kokonaisuudessaan on. Mielestäni henkilökohtaiset asenteet voivat kuitenkin varsinkin pienemmissä yrityksissä vaikuttaa ratkaisevasti CASE-välineiden hyväksyntään käyttäjien keskuudessa.

Näiden perustelujen myötä olemme käyneet läpi tämän tutkimuksen kirjallisuuskatsauksen. Aluksi pohdittiin käyttäjätyytyväisyyden mittaamisen vaikeutta ja sen jälkeen jaoimme käyttäjätyytyväisyyteen vaikuttavat tekijät eri luokkiin. Tämän jälkeen esiteltiin eri kategorioiden tekijöitä ja pohdittiin niiden vaikutuksia. Ympäristöllisten tekijöiden merkitystä painotettiin välineen ominaisuuksien ohella. Ympäristötekijöistä tärkeimpiä olivat

oikeanlainen koulutus ja vapaaehtoisuus, välineen ominaisuuksista puolestaan kaikki esitellyt olivat varsin tärkeitä. Lopuksi pohdittiin käyttäjien henkilökohtaisten ominaisuuksien vaikutusta ja todettiin sen olevan varsin merkittävässä asemassa ainakin pienimmissä CASE-välineitä käyttävissä yrityksissä. Seuraavaksi vuorossa on empiirisen kyselytutkimuksen tulosten analysointia ja niiden heijastamista edellä saatuihin tuloksiin.

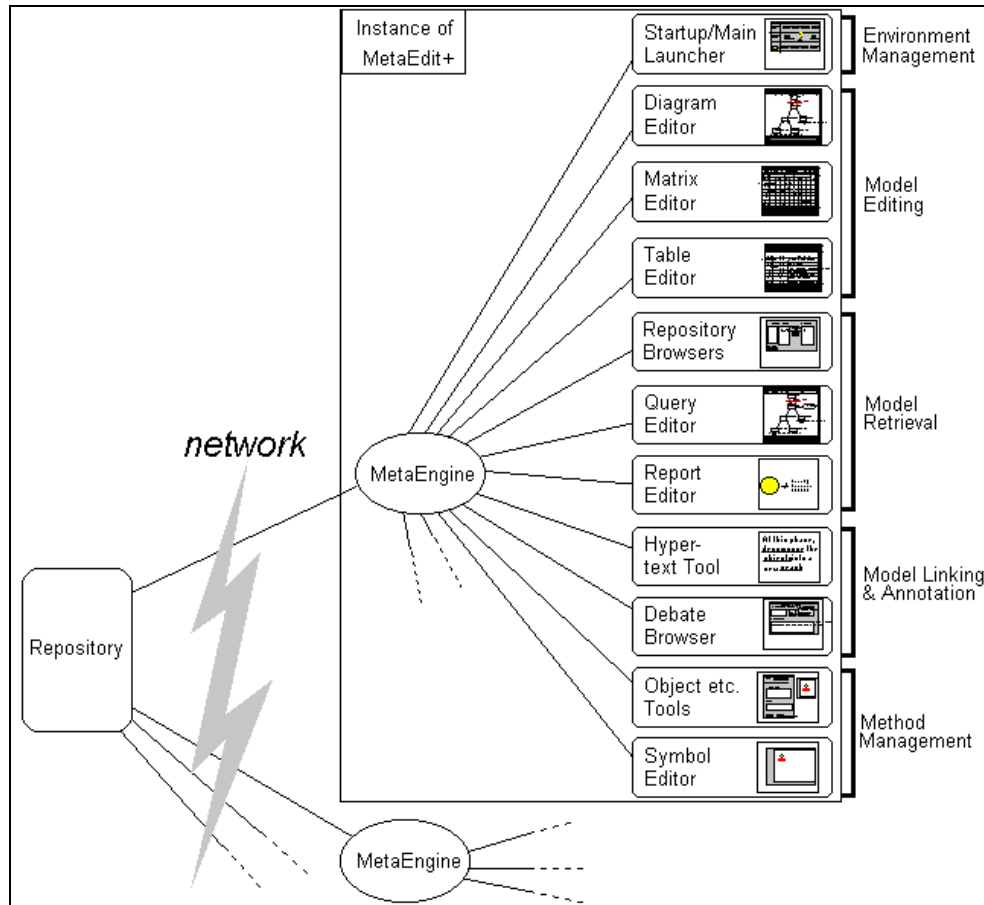
5 MetaEdit+ ja käyttäjätyytyväisyys suuressa tietoliikennealan yrityksessä

5.1 MetaEdit+:n esittely

Valitsemassani yrityksessä käytettiin tutkimushetkellä (helmikuu 2001) suomalaisen Metacase Consulting -yrityksen (<http://www.metacase.com>) kehittämää MetaEdit+-nimistä CASE-välinettä. Välineestä oli käytössä kaksi eri versiota, joista puhutaan tarkemmin tuonempana. CASE-ympäristöön kuului myös MetaEdit+:n tietokannoista koodia generoiva koodigeneraattori. MetaEdit+ ei varsinaisesti ole alkuperäisessä muodossaan kokonainen CASE-väline, vaan se pikemminkin käytännössä mahdollistaa oman CASE-välineen luonnin tietyn, ennalta määrätyn arkkitehtuurin pohjalta. Tässäkin tapauksessa yrityksen sisällä oli luotu oma, tiettyyn tarkoitukseen sopiva arkkitehtuuri. Tämän vuoksi on otettava huomioon myös se, että käyttäjien mielipiteet riippuvat myös tästä arkkitehtuurista ja sen erityispiirteistä. Valitettavasti en voi tässä yhteydessä vaitiolovelvollisuuteni vuoksi eritellä näitä piirteitä, mutta haluan korostaa, että tässä tutkimukseni osiossa käsitellään MetaEdit+ -välinettä ja yrityksen siihen luomaa arkkitehtuuria yhtenä kokonaisena yksikkönä. Käytän tästä kokonaisesta yksiköstä puhuttaessa yleisnimitystä MetaEdit+. Tästä johtuen kaikki tekemäni johtopäätökset eivät ole yleistettävissä muihin MetaEdit+:n avulla suunniteltuihin CASE-välineisiin. MetaEdit+:n erityispiirteistä ja toimintaan liittyvistä vaiheista puhutaan enemmän kohdassa 5.4.

Muistutan vielä tässä yhteydessä, että tarkoitukseni ei varsinaisesti ole arvioida kyseisen CASE-välineen ominaisuuksia, vaan selvittää, mitkä tekijät vaikuttavat

käyttäjätyytyväisyyden syntymiseen. En kuitenkaan kokonaan pidäyt arvioimasta välinettä niissä yhteyksissä, joissa se palvelee myös tarkoitustani käyttäjätyytyväisyyden arvioinnissa.



Kuvio 5. MetaEdit+:-n arkkitehtuuri [MetaPHOR Group, 2001].

Itse henkilökohtaisesti olen tutustunut työkaluun noin kahden vuoden ajan käyttäen sitä lähes päivittäin. Voidaankin sanoa, että tunnen sen hyvät ja huonot puolet varsin kattavasti. Mielestäni järjestelmätuntemuksestani on suurimmaksi osaksi hyötyä tutkimukselleni, mutta voidaan myös ajatella sen mahdollisesti aiheuttavan ennakkokäsityksiä ja puolueellisuutta. Petersonin mukaan kaikki yrityksen sisäiset tutkimukset ovat vinoutuneita johonkin suuntaan [Peterson, 1996]. Ymmärrän, että saatan huomaamattani kadottaa objektiivisuuteni joissain tilanteissa, mutta pyrin seuraavissa kappaleissa erittelemään, mitkä seikat ovat omia kokemuksiani ja mitkä taas tutkimuksessa esiin tullutta aineistoa. On myös huomattava, että tämän tutkimuksen ei ole tarkoitus arvioida käsiteltyä CASE-välinettä, vaan tuoda esiin subjektiivisia käyttäjien kokemuksia sen käytöstä. Käyttäjien kokemusten avulla puolestaan pyritään etsimään tekijöitä, jotka luonnehtivat CASE-välineitä yleensä. Katson henkilökohtaisten kokemusten palvelevan tässä yhteydessä koko tutkimuksen etua, sillä tällä

tavalla pohdiskeluni ei jää pelkäksi teoreettiseksi sanahelinäksi, vaan sillä on myös vankka perusta todellisuudessa. Kokemukseni voi myös tarjota aiheeseen sellaisia näkökulmia, joita välineelle täysin vieras tutkija ei osaisi välttämättä ajatella.

5.2 *MetaEdit+:*n käyttöönottoon johtaneet syyt

Arvioitaessa CASE-välineen käyttäjätyytyväisyyttä on mielestäni tärkeää ottaa myös huomioon ne tekijät, jotka johtivat kyseisen välineen käyttöönottoon. Voidaan perustellusti väittää, että nämä seikat ne eivät suoranaisesti vaikuta siihen, miten käyttäjät välineiden käytön kokevat, mutta on varmasti totta, että ne saattavat luoda käyttäjille tiettyjä ennako-odotuksia käyttöönotettavan työkalun suhteen. Nämä ennako-odotukset voivat puolestaan johtaa yhtä hyvin tyytyväisiin ja tyytymättömiin välineiden käyttäjiin. Tutkimukseni alkuvaiheissa haastattelin joitakin johtavassa asemassa olevia henkilöitä CASE-välineen käyttöön johtaneista tekijöistä. Esittelen seuraavissa alakohdissa haastattelujeni tuloksia, soveltaen niihin edellisten kohtien tarjoamaa teoreettista pohjaa.

5.2.1 Tuotetun ohjelmakoodin modulaarisuus

Eräänä vahvana syynä siirtymisessä perinteisestä ohjelmakoodin tuotannosta CASE-työkalujen käyttöön ilmeni tarve erilaiselle tuotteiden räätälöinnille. CASE-välineiden avulla pystyttiin graafisia apuvälineitä käyttäen eriyttämään helposti samasta kokonaisuudesta eri tuotteille suunnitellut ominaisuudet. Erilaisia ominaisuuksia pystyttiin lisäämään ja poistamaan häiritsemättä tuotteen muiden osien toimintaa. Huomioitavaa kuitenkin on, että pelkkä modulaarisuutta tukeva CASE-väline ei tietystikään riitä, vaan koko tuotteen suunnitteluarkkitehtuurin tulee tukea modulaarista suunnittelutapaa.

5.2.2 Projektin parempi hallinta

Sumnerin tavoin [Sumner, 1995] kohdeyrityksen entisten projektien aikana oli huomattu, että aikataulujen ja koko projektin hallinta oli vaikeaa. Projektien kestoa oli vaikeaa arvioida ja se

aiheutti taloudellisia rasitteita. Uusien välineiden avulla pyrittiinkin entistä parempaan aikataulujen arviointiin ja sitä kautta koko projektin hallinnan helpottumiseen.

Sumnerin mukaan CASE-välineen käyttöönotto tuo järjestelmäsuunnittelijoiden työhön tiettyä kurinalaisuutta, jonka avulla he voivat paremmin hallita ja koordinoida työtään. Myös Stobartin et al. mukaan yritykset odottavat CASE-välineisiin sijoitetun rahasummien tuottavan vastaavasti tuloksia projektin parempana hallittavuutena [Stobart et al., 1993].

5.2.3 Suunnitteludokumenttien tuottamisen automatisointi

Suunnitteludokumenttien tuottamiseen kului yrityksessä huomattavan paljon aikaa ja uuden välineen avulla haluttiin automatisoida suunnitteludokumenttien tuottaminen. Ajatuksena oli, että MetaEdit+:lla tuotetut graafiset mallit toimisivat samalla suunnitteludokumentteina, joten erillisiä ponnisteluja niiden valmistumiseksi ei enää tarvittaisi.

5.3 *MetaEdit+:*n ennakko-odotusten toteutuminen johdon näkökulmasta

Haastattelin eräitä yrityksen johtohenkilöitä sähköpostin välityksellä siitä, miten he olivat asettaneet ennakko-odotuksensa ja miten ne olivat lopulta toteutuneet. Lisäksi kysyin, oliko käyttöönoton jälkeen ilmennyt mitään negatiivisia tai positiivisia yllätyksiä.

Johdon mukaan heidän ennakko-odotuksensa olivat olleet jonkin verran liian optimistisia. Kokonaisuutena MetaEdit+:n hyödyt eivät olleet sillä tasolla, jota oli odotettu. Erityisesti pettuneitä oltiin siihen, että manuaalisesti erikseen tuotetuista suunnitteludokumenteista ei pystyttykään täysin luopumaan, koska MetaEdit+:n suunnittelumallit eivät täyttäneet suunnitteludokumenteille asetettuja vaatimuksia. Kuten seuraavassa luvussa kerrotaan, eivät käyttäjäkään olleet tyytyväisiä MetaEdit+:n suunnittelulle antamaan tukeen.

Ennakko-odotuksista poiketen myöskään koodin tuottamista ei pystytty täysin automatisoimaan, koska koodigeneraattorin tuottama koodi oli hyvin vaikeaselkoista ja se saattoi sisältää virheitä. Ylipäätänsä johdon antama palaute noudatteli pääpiirteissään tuonnempana esiteltäviä kyselytutkimuksen tuloksia. Esimerkiksi versionhallinta koettiin

vaikeaksi, koska selvää yhteyttä generoidun koodin ja graafisten mallien välille oli vaikea hahmottaa.

5.4 *MetaEdit+ ja CASE-välineiden luokat*

Esittelen MetaEdit+:n pääpiirteissään seuraavissa kappaleissa, jotta lukijat voivat paremmin ymmärtää sitä koskevien argumenttien luonteen. Käsittelen välinettä ensin suhteessa CASE-välineiden jaotteluihin ja siirryn sitten välineen taustalla olevien periaatteiden tarkasteluun.

Useimmissa jaotteluissa CASE-välineet jaetaan seuraavalla tavalla [Lending ja Chervany, 1998]:

1. Korkean tason CASE –välineet (upper CASE)
2. Matalan tason CASE –välineet (lower CASE)
3. Yhdistetyt CASE-välineet (integrated CASE)

Ensimmäinen luokka tarkoittaa välineitä, jotka tukevat järjestelmäsuunnittelun alkuvaiheita, kuten vaatimusmäärittelyä ja suunnittelua. Matalan tason CASE-välineet puolestaan keskittyvät myöhempisiin vaiheisiin, kuten koodin tuottamiseen ja sen testaamiseen. Yhdistetyt välineet taas yhdistävät edellä kuvattujen työkalujen ominaisuuksia ja tukevat ohjelmistosuunnittelun kaikkia vaiheita. MetaEdit+ voidaan lukea matalan tason välineisiin kuuluvaksi, sillä se tukee oikeastaan suoranaisesti vain koodin tuottamista. Toisaalta taas siinä on joitain suunnittelua tukevia elementtejä, joten tätä välinettä voitaisiin myös jossain mielessä kutsua yhdistetyksi välineeksi, tosin varsin rajoittuneeksi sellaiseksi.

MetaEdit+ ei sinällään ole kokonainen CASE-väline ainakaan johdannon määritelmään tukeutuen, vaan se tarjoaa ikään kuin perustan tällaisen välineen rakentamiselle ja sen soveltamiselle käsillä olevaan ongelmaan. Tutkittavassa yrityksessä on kehitetty MetaEdit+:saan liittyen oma tietokanta, jossa on tällä CASE-työkalulla suoritettavien tehtävien vaatimat komponentit. Myös komponenttien esitystapa ja ominaisuudet ovat konfiguroitavissa. Finnigan et al. kuvaavat MetaEdit+:n kaltaisia välineitä meta-CASE-välineiksi [Finnigan et al., 2000]. Edellä kuvattua jaottelua voitaisiinkin täydentää neljännellä

kohdalla, mutta toisaalta taas näiden meta-CASE-välineiden avulla tuotetaan kuitenkin aina jokin jaottelun kolmen ensimmäisen kohdan välineistä.

Halutut sovellukset suunnitellaan tilapohjaisella mallilla graafisessa ympäristössä. Ympäristö jakautuu erilaisiin osiin käyttötarkoituksen perusteella. Kaikki sovellusta kuvaavaan malliin lisätyt komponentit tallentuvat tietokantaan. Kuten Prather artikkelissaan toteaa [Prather, 1993], CASE-välineet jakautuvat yleensä kahteen luokkaan. Toiseen kategoriaan kuuluvat suuret monen käyttäjän yhtäaikaiset tietokannat ja toiseen yhden käyttäjän istuntopohjaiset työkalut. MetaEdit+ -ympäristössä sovellusten suunnittelu tapahtuu istunnoissa. Jokaisen istunnon lopuksi ohjelma tallentaa istunnon aikana tietokantaan tehdyt muutokset. Kun halutaan tuottaa tiettyä sovellusta varten ohjelmakoodi, tuotetaan ensin tietokannasta tätä varten prolog-ohjelmointikielinen tiedosto. Tämän tiedoston perusteella juuri tätä tarkoitusta varten kehitetyt C-kielen koodigeneraattorit tuottavat käännettävän ohjelmakoodin.

Yhteenvetona edellisistä kappaleista MetaEdit+:n avulla luodun sovelluksen kehityskaari siis muodostuu seuraavista vaiheista. Ensin käyttäjä kirjautuu sisään tunnuksellaan ja salasanallaan MetaEdit+ -istuntoon, jossa hän haluaa uutta sovellusta kuvaavan mallin luoda. Tämän jälkeen käyttäjä määrittelee suunnittelukomponenttien avulla graafisesti haluamansa toiminnot. Sitten ohjelmakoodi realisoidaan prolog-kielisiksi ilmaisuiksi, joista koodigeneraattorilla tuotetaan varsinainen c-kielinen käännettävä ohjelmakoodi.

Näiden kappaleiden tarkoituksena oli selvittää lyhyesti MetaEdit+:n toimintaperiaatteet. Tarkemmasta esittelystä en katso olevan juurikaan hyötyä käyttäjäytyväisyyden kannalta, koska seuraavissa kohdissa kuitenkin esitellään melko tarkasti siihen liittyviä tekijöitä.

5.5 MetaEdit+:n ominaisuuksien hyödyntäminen

Kuten Lending ja Chervany [Lending ja Chervany, 1998] artikkelissaan toteavat, monissa yrityksissä CASE-välineen hankintaa perusteltiin sen tarjoamalla monilla erilaisilla ominaisuuksilla. Eräässä yrityksessä jopa valittiin markkinoiden monipuolisin väline, jota sitten loppujen lopuksi käytettiin ainoastaan kommunikaatiovälineenä suunnittelijoiden välillä. CASE-välineitä arvioitaessa on kuitenkin ennen kaikkea kiinnitettävä huomiota siihen, mitä välineen ominaisuuksia käytetään ja millä tavalla.

Kuten edellä esiteltiin, MetaEdit+ ei varsinaisesti ole kokonainen CASE-väline vaan oikeammin väline sellaisen suunnittelulle. Yrityksessä luotu arkkitehtuuri saattaa käyttää tai olla käyttämättä kaikkia MetaEdit+:n tarjoamia mahdollisuuksia. Yrityksen kuitenkin seuraavassa hahmotella, minkälaisia ominaisuuksia juuri tässä yrityksessä toteutetussa välineessä käytettiin.

Välineen suunnitteluominaisuuksia käytettiin hyväksi jonkin verran, mutta mitään suunnitelmallista hyväksikäyttöä ei esiintynyt. Suunnittelijat kyllä rakensivat sovelluksiaan CASE-välineen avulla, mutta tämä tapahtui enemmänkin yrityksen ja erehdyksen kautta. On kuitenkin huomattava, että MetaEdit+, eikä sen avulla suunniteltu erikoisvälinekään tarjoa alun alkaenkaan kovinkaan laajoja suunnittelun apuvälineitä.

Välineen tarjoamia varsinaisia ohjelmakoodin tuottamiseen tarkoitettuja ominaisuuksia taas käytettiin varsin laajasti, ja luonnollisesti koodigeneraattorit olivat kaikkien käytössä. Kuten jäljempänä todetaan, ei kaikkia uudelleenkäyttöön liittyviä ominaisuuksia käytetty niiden kömpelön toteutuksen vuoksi.

Tämän kohdan tarkoituksena oli selvittää, mitä ominaisuuksia käyttäjät CASE-välineestä käyttivät. Melko suppean integroidun CASE-toteutuksen vuoksi MetaEdit+ ei tarjoa kovinkaan hienostuneita suunnittelun apuvälineitä. Ehkä tästä johtuen MetaEdit+:n kaikkia ominaisuuksia käytettiin melko tasapuolisesti hyväksi, lukuun ottamatta kaikkia uudelleenkäyttöominaisuuksia, jotka käyttäjät kokivat joiltain osin epämiellyttävästi toteutetuiksi.

5.6 MetaEdit+:n käyttäjätyytyväisyyttä tutkiva empiirinen tutkimus

Seuraavissa kohdissa käsittelen järjestetyn kyselytutkimuksen läpivientä ja esitän perusteluja menettelytapoja koskeville valinnoille. Ensin keskitytään tutkimusmenetelmiin, sen jälkeen kartoitetaan tutkimukseen osallistuneiden taustoja ja valintaperusteita. Tämän jälkeen esitellään tutkimuksen peruspilarit, eli asetetut hypoteesit, joiden avulla arvioidaan tulevissa kohdissa kyselyn avulla saavutettuja tuloksia.

5.6.1 Tutkimusmenetelmät

Tutkimus suoritettiin yrityksessä kaksiosaisena kyselytutkimuksena. Ensimmäisessä osassa lähetettiin 27 työntekijälle sähköpostina kysely, jonka pohjalta laadittiin tarkentavat kysymykset kyselyn toiseen osaan. Toinen osa suoritettiin haastattelututkimuksena, jossa yrityksen työntekijöitä haastateltiin heidän työpaikoillaan. Tutkimuksen luotettavuuden lisäämiseksi pyrittiin aikaansaamaan tutkijan ja haastateltavien välille jatkuva dialogi, jotta välttyttäisiin mahdollisilta väärinkäsityksiltä. Tässä etuna oli myös asiantuntemukseni käsiteltävästä järjestelmästä. Petersonin mukaan [Peterson, 1996] kaikki kyselytutkimukset kärsivät vastaajien vastausten vinoutuneisuudesta. Erityisesti haastattelutilanteissa haastatellut saattavat helpommin vastata haastattelijan odottamia vastauksia. Tätä vinoumaa yritettiin torjua lähettämällä haastattelulomakkeet ensin haastatelluille sähköpostitse ja turvautumalla henkilökohtaisiin haastatteluihin tarvittaessa vasta sen jälkeen.

Hartrum et al. mukaan [Hartrum et al., 1989] CASE-välineen tutkimiseen kuuluu kuusi vaihetta:

1. Määrittele tutkimuksen tavoitteet
2. Laadi tutkimuskysymykset
3. Suunnittele kyselykaavake
4. Kerää tiedot
5. Analysoi kerätyt tiedot
6. Vahvista tutkimuksen luotettavuus

Mielestäni Hartrum et al. ovat kiteyttäneet oivasti CASE-välineen arviointiin liittyvät tekijät. Sen vuoksi tämä kyselytutkimus pyrki seuraamaan yllä kuvattuja vaiheita mahdollisimman tarkasti, mutta kuitenkin tietyiltä osin soveltaen niitä käsillä olevaan tilanteeseen. Hartrum et al. metodologia perustuu itse asiassa Bailey ja Pearsonin lähestymistapaan [Bailey ja Pearson, 1983], mutta tämän lähestymistavan käsittely ei enää ole relevantti tämän tutkimuksen kannalta, koska kyseessä on kuitenkin suhteellisen pieni otos ja melko pieni kysymysten määrä.

Tutkimuksen tavoitteet määriteltiin hypoteesien avulla, joista kerrotaan enemmän kohdassa 5.6.3. Tämän jälkeen laadittiin kyselylomake Microsoft Excel –ohjelmalla ja se lähetettiin sähköpostitse kyselyyn valituille henkilöille. Tämä kyselytutkimus eroaa jonkin verran Hartrumin et al. määrittelemistä puitteista kysymysten asettelun suhteen. Heidän ohjeensa mukaan jokaisen kysymyksen vastausvaihtoehtona tulisi olla 7-portainen asteikko, mutta tämän kyselyn pienen koon ja osallistujamäärän vuoksi vastausvaihtoehdot rajoitettiin viiteen eri vaihtoehtoon. Tämä ratkaisu helpotti toisaalta tulosten analysointia, mutta toisaalta taas se vei mahdollisuuden arvioida vastausten luotettavuutta käyttämällä Hartrumin et al. määrittelemiä keinoja. Katson kuitenkin, että pystyn tästä huolimatta arvioimaan saamiani tuloksia luotettavasti. Jälkeenpäin arvioituna olisi kuitenkin ollut viisasta sisällyttää mukaan Hartrumin et al. tapaan jokaisen kysymyksen jälkeen jatkokysymys siitä, kuinka tärkeäksi vastaaja arvioi kyseisen seikan merkityksen työnsä kannalta. Tämä olisi voinut parantaa kuvaa siitä, mitkä asiat vastaajille todella olivat tärkeitä. Tämä osa kuitenkin jätettiin pois lopullisesta versiosta, jotta kyselystä saataisiin helpommin käsiteltävä kokonaisuus.

Kysymysten määrä pidettiin tarkoituksella alhaisena, jotta saataisiin mahdollisimman suuri vastausprosentti. Erilaisia mittausvirheitä pyrittiin poistamaan selvittämällä vastaajien työhistoriaa koskevia seikkoja, kuten työskentelyaikaa yrityksessä, MetaEdit+:n käyttöaikaa ja käyttötiheyttä, ikää ja sukupuolta.

Kysymyksiä kyselyssä oli kaikkiaan 25 kappaletta, joista viimeiset kolme olivat avoimia. Useimpiin kysymyksiin oli valmiiksi laadittu viisi mahdollista vaihtoehtoa kuvaamaan suhtautumista kyseiseen asiaan. Vastaukset kysymyksiin koskien välineen käyttöaikaa, oppimisen kestoa ja työskentelyaikaa yrityksessä koodattiin tulosten analysoinnin yhteydessä kukin tilanteeseen sopivan jaottelun mukaan. Käyttöaikaa ja työskentelyaikaa yrityksessä arvioitiin puolen vuoden tarkkuudella, oppimiseen kulunutta aikaa puolestaan kuukauden tarkkuudella.

5.6.2 Tutkimuksen osallistujat

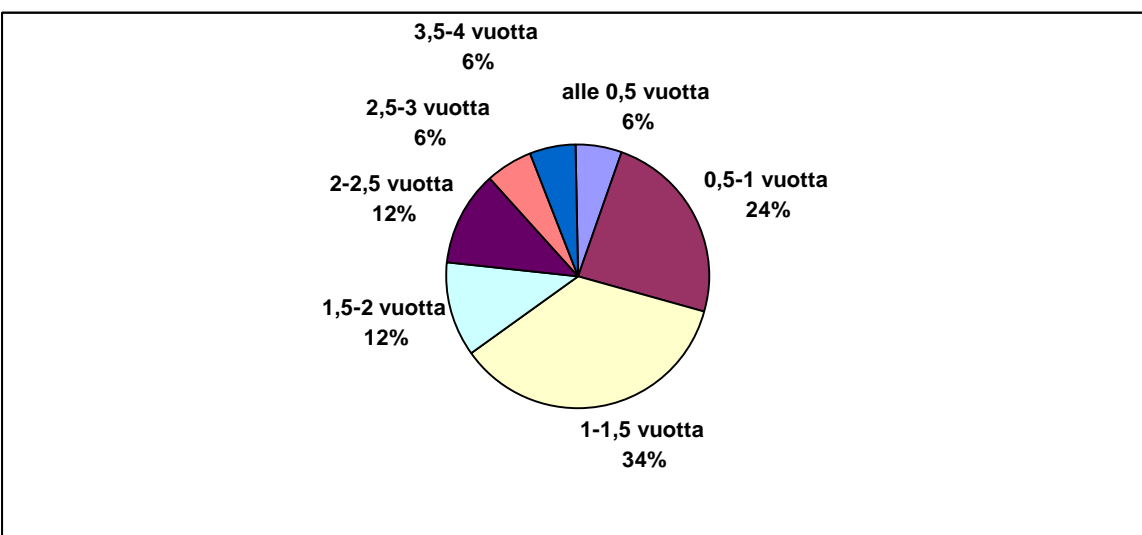
Vastaajat valittiin satunnaisesti yrityksen suomenkielisten MetaEdit+ -käyttäjien keskuudesta. Kysely lähetettiin kaikkiaan yhteensä 27 henkilölle, joista 17 vastasi siihen määräaikaan mennessä. Vastaajista kaikki olivat miehiä, iältään 24 – 42 vuotiaita. MetaEdit+ -ohjelmiston

käyttöaika vaihteli kolmesta kuukaudesta kolmeen vuoteen. Suurin osa vastaajista käytti työkalua päivittäin (11 henkilöä) ja loput vastaajista ainakin kerran viikossa, joten voidaan perustellusti todeta vastaajien olevan hyvin perillä ohjelmistosta ja sen käytöstä.

Yrityksessä vastaajat olivat olleet vuodesta kuuteen vuoteen. Vastaajien keskimääräiset tiedot on esitelty taulukossa 1.

Ikä	29 vuotta
Työssä yrityksessä	2 vuotta 9 kk
Käyttäneet välinettä	1 vuosi 8 kk

***Taulukko 1.** Tutkimukseen osallistuneiden keskimääräinen ikä, työskentelyaika yrityksessä sekä välineen käyttöaika.*



***Kuvio 6.** Välineen käyttöaika.*

Juuri kukaan ei ollut käyttänyt mitään muita CASE-välineitä, ainoastaan yksi vastaajista kertoi käyttäneensä Prosa-nimistä työkalua. Hartrum et al. totesivat tutkimuksensa valossa näyttävän jonkin verran siltä, että jotain muuta CASE-välinettä ensin käyttäneet antavat positiivisempia arvioita tutkittavasta välineestä [Hartrum et al., 1989]. Myös Aaen on havainnut tutkimuksessaan vastaavia trendejä [Aaen, 1993]. Tämän tutkimuksen yhteydessä tätä väitettä ei kuitenkaan päästy tutkimaan, koska vastaajilla ei yhtä lukuun ottamatta ollut kokemusta muista CASE-välineistä. Toinen huomionarvoinen seikka on, että yli puolet

vastaajista (9 henkilöä) olivat heti yritykseen tullessaan alkaneet käyttää MetaEdit+ -ohjelmistoa, eivätkä olleet käyttäneet mitään muita vaihtoehtoisia välineitä. Tämä on mielestäni toisaalta hyvä, sillä näin käyttäjien mielipiteet ovat todella vapaita ennakkokäsityksistä, mutta toisaalta taas heillä ei ole vertailupohjaa mihinkään muihin tapoihin tuottaa ohjelmakoodia. Tästä lisää kohdassa 6.2.3.

Kuten aikaisemmin mainitsin, MetaEdit+ -välineestä on yrityksessä käytössä kaksi eri versiota, 2.5 ja 3.0, joista jälkimmäinen tarjoaa eräitä parannuksia edelliseen verrattuna. Koska tämän tutkimuksen tarkoitus ei ole varsinaisesti eri versioiden ominaisuuksien vertailu, sivuutan nämä eroavaisuudet tässä yhteydessä ja palaan niihin myöhemmin käsiteltäessä kyselytutkimuksen tuloksia. Vastaajista 12 henkilöä käytti uudempaa versiota työkalusta ja viisi versiota 2.5.

5.6.3 Tutkimushypoteesit

Ennen kyselytutkimusta asetettiin viisi hypoteesia, joiden toteumista haluttiin seurata kyselytutkimuksen avulla. Näitä olivat:

- H1** MetaEdit+ version 3.0 käyttäjät ovat tyytyväisempiä kuin version 2.5 käyttäjät
- H2** Enemmän koulutusta saaneet ovat tyytyväisempiä
- H3** Vapaaehtoisesti välinettä käyttävät ovat tyytyväisempiä
- H4** Mitä monimutkaisemmaksi käyttäjä kokee välineen käytön, sitä tyytymättömämpi hän on
- H5** Mitä nopeammin käyttäjä omaksuu välineen käytön, sitä tyytyväisempi hän siihen on

Huomioina hypoteeseista haluaisin sanoa sen verran, että ensimmäinen hypoteesi oli enemmänkin itseäni kiinnostava kuin relevantti CASE-välineiden yleisen käyttäjätyytyväisyyden kannalta. Tämä johtuu siitä, että omassa käytössäni on tosiaan vanhempi versio MetaEdit+ -välineestä ja halusin tietää, olisiko uudempi versio oleellisesti erilainen käyttäjätyytyväisyyden kannalta tutkittuna. Lisäksi, kuten myös Hartrum et al. korostavat [Hartrum et al., 1989], on tärkeää tutkia yrityksen investointien kannalta uuden version edut vanhaan verrattuna. Muut hypoteesit muodostin kirjallisuuskatsauksessa esiin

tulleiden yleisten trendien perusteella, nähdäkseni noudattiko tutkimani yrityksen tilanne muiden tutkimusten kautta saatuja tuloksia.

6 Kyselytutkimuksen tulokset ja niiden analysointi

Tässä luvussa tarkoitukseni on käydä yksityiskohtaisesti läpi edellä kirjallisuuskatsauksessa esiin tulleiden käyttäjätyytyväisyyteen vaikuttavien tekijöiden esiintyminen todellisessa CASE-välineen käytössä. Lisäksi pohditaan joitain kirjallisuuskatsauksen ulkopuolisia seikkoja, jotka eivät tulleet esille edellisissä kappaleissa. Näitä ovat esimerkiksi tuki prosessille ja oppimiskynnys. Pyrin arvioimaan jokaisessa kohdassa asetettujen tutkimushypoteesien toteutumista ja syitä saamiini tuloksiin.

Käsittelen ensin MetaEdit+:n ominaisuuksia ja siirryn sen jälkeen pohtimaan ympäristön vaikutusta käyttäjätyytyväisyyteen tässä yrityksessä. Luvun loppupuolella puolestaan keskitytään miettimään, miten tässä tapauksessa yksilöiden erot vaikuttivat heidän suhtautumiseensa välineeseen. Lopuksi käsitellään yhteenvetona kunkin hypoteesin toteutuminen ja arvioidaan tuloksiin johtaneita syitä.

6.1 MetaEdit+:n käyttäjätyytyväisyyteen vaikuttavat ominaisuudet

6.1.1 Yleistä

Seuraavissa kappaleissa aion esitellä lyhyesti tärkeimmät MetaEdit+ -työkalun käyttäjätyytyväisyyttä muokkaavat elementit. Välineen ominaisuuksista voisi kirjoittaa huomattavasti laveamminkin, mutta mielestäni ne ovat tutkimuskohde erikseen, ja kuten olen aiemmin todennutkin, tämän tutkimuksen tarkoitus ei ole olla tuotearviointi. Esittelen kuitenkin lyhyesti ne MetaEdit+:n ominaisuudet, joilla on merkitystä käyttäjätyytyväisyyden muodostumisessa. Kerron myös hiukan itse välineen ominaisuuksista kyselyn ulkopuoleltakin, jotta lukija voisi paremmin ymmärtää missä kontekstissa asioita tässä kyselytutkimuksessa käsitellään.

Jos pohditaan tämän kyselyn tuottamia tuloksia suhteessa muiden tutkimusten tuottamiin tuloksiin, voidaan sanoa että eroja ilmeni jonkin verran. Esimerkiksi yhteensopivuus muiden käytettyjen työkalujen kanssa koettiin verrattain hyväksi, kun Stobart et al. taas totesivat tämän olevan yleinen ongelma CASE-välineiden käytössä [Stobart et al., 1993]. Toisaalta taas Stobartin et al. mukaan välineen liian varhainen käyttöönotto johtaa epävakautteen ohjelmistojen käytössä, mikä ilmeni myös tässä tutkimuksessa.

Toteuttamassani kyselyssä (liite 1) osallistujilta kysyttiin muuan muassa MetaEdit+:n ominaisuuksien vaikutuksista heidän päivittäiseen työmotivaatioonsa. Tulokset olivat kuitenkin jokseenkin mitään sanomattomia, koska suurin osa vastaajista ei osannut sanoa, miten välineen ominaisuudet heidän työhönsä vaikuttivat. Vastaajista kolme oli sitä mieltä, että ominaisuudet vaikuttivat negatiivisesti heidän työhalukkuuteensa ja viisi puolestaan kertoi niiden vaikuttavan positiivisesti. Kenties kysymysten asettelu olisi voinut olla hieman tarkempi, sillä ainakaan näillä vastauksilla ei ole juurikaan informatiivista arvoa.

Kyselyn lopussa oli varattu tilaa kolmelle avoimelle kysymykselle, joiden tulokset olivatkin varsin hedelmällisiä koskien varsinkin käyttöliittymän ja käytetyn koodigeneraattorin puutteita. Seuraavassa ensin lyhyt yhteenveto käyttäjien yleisistä mielipiteistä välineestä.

Ylipäätänsä käyttäjät suhtautuivat melko positiivisesti välineen ominaisuuksiin. Esimerkiksi hyväksi ominaisuudeksi koettiin se, että MetaEdit+ piilottaa todellisen ohjelmakoodin toteutuksen ja näin ohjelmoija ei välttämättä tarvitse niin hyvää ohjelmointitaitoa. Tämä asia on ollut esillä monissa muissakin tutkimuksissa [Dabin, 1991; Kemerer, 1992; Sumner, 1995]. Tämän tutkimuksen yhteydessä se ei välttämättä pitänyt paikkaansa, sillä koodigeneraattorin virhetilanteista selviytyminen vaatii hyvää tuotetun ohjelmakoodin hallintaa. Tästä lisää kohdassa 6.1.6.

Kyselytutkimukseen vastanneiden mielestä MetaEdit+:n perimmäinen idea oli varsin hyvä, mutta toteutus ei ollut paras mahdollinen. Ohjelmakoodin tuottaminen oli helppoa, kun oli ensin oppinut välineen käytön ja sen omat kiemurat. Jotkut olivat kuitenkin sitä mieltä, että joidenkin välineen ongelmien kiertämiseksi joutui tekemään liikaa työtä. Lisäksi MetaEdit+:lla tehtyjen toteutuksien vaikeaselkoisuuteen vaikuttivat käyttäjien mukaan myös vanhempien versioiden puutteelliset ominaisuudet ja käyttäjien kokemattomuus.

6.1.2 Kompleksisuus

Seuraavissa kappaleissa pohditaan MetaEdit+ -ohjelmiston kompleksisuutta ja siihen liittyvän hypoteesin H4 toteutumista tämän tutkimuksen valossa. Hypoteesin havaittiin toteutuneen vain osittain, mutta kompleksisuuden kokeminen näytti korreloivan myös koulutuksen määrän ja yrityksessä työskennellyn ajan suhteen. Myös hypoteesi H1 näytti pitävän osittain paikkansa, sillä vanhemman version käyttäjät antoivat negatiivisempia arvioita välineestä.

MetaEdit+:n kompleksisuutta mitattiin erikseen sitä varten laaditulla kysymyksellä (liite 1, kysymys 19). Kysymyksessä vastaajilla kysyttiin, kuinka hyvin he kokivat tuntevansa MetaEdit+:n toimintaperiaatteet, eli tuntuiko järjestelmän käyttäminen monimutkaiselta vai helpolta. Vastaukset osoittivat, että välineen toimintaperiaatteet koettiin suhteellisen helposti ymmärrettäviksi, sillä yhdeksän vastaajista oli sitä mieltä, toimintaperiaatteet olivat melko helposti tai helposti ymmärrettäviä. Vain viisi vastaajista oli sitä mieltä, että toimintaperiaatteet olivat vaikeasti tai melko vaikeasti ymmärrettäviä.

Jos pohditaan hypoteesin H4 toteutumista (*Mitä monimutkaisemmaksi käyttäjä kokee välineen käytön, sitä tyytymättömämpi hän on*), on tulos hiukan kaksijakoinen. Kolme viidestä kompleksisuudesta valittaneista vastaajista oli sitä mieltä, että he suoriutuisivat työstään paremmin käyttämällä perinteisiä menetelmiä (ohjelmoimalla käsin). Heistä kahden mielestä MetaEdit+:n käyttö oli vastenmielistä ja sen ominaisuudet vaikuttivat heidän työmotivaatioonsa negatiivisesti. Kolmas perinteisien menetelmien kannattaja taas oli sitä mieltä, että ohjelman käyttö oli hankalaa. Nämä tulokset näyttävät tukevan hypoteesia H4.

Toisen puolen tästä viiden hengen ryhmästä puolestaan muodostivat kaksi muuta vastaajaa, joiden mielestä välineen käyttö oli mieluisaa ja jotka kokivat sen avulla suoriutuvansa työstään perinteisiä menetelmiä paremmin. Toisen mielestä välineen ominaisuudet jopa vaikuttivat hänen työmoraaliinsa kohottavasti.

Tässä tutkimuksessa siis hypoteesi H4 ei näyttänyt pitävän paikkaansa kuin osittain. Huomionarvoista kuitenkin oli se, että kumpikin kaikkein epäsuotuisimpien arvioiden antajasta käytti MetaEdit+ versiota 2.5. Tämä puolestaan näyttäisi tukevan hypoteesia H1, eli version 3.0 käyttäjät näyttivät tässä suhteessa olevan tyytyväisempiä.

MetaEdit+ -työkalun monimutkaisuuden kokeminen näytti korreloivan jonkin verran sen kanssa, kuinka kauan vastaajat olivat työskennelleet yrityksessä, sillä viidestä kompleksisuudesta valittaneen joukosta pisimpään työskennellyt oli ollut yrityksessä kuusi vuotta, kaksi seuraavaa vastaajista neljä vuotta, neljäs henkilö kolme vuotta ja vähiten ollut yhden vuoden. Kaikkien vastaajien keskuudessa mediaaniaika oli puolestatoista vuodesta kahteen vuoteen ja keskimääräinen aika kaksi vuotta ja yhdeksän kuukautta. Osa näistä tiedoista on nähtävissä taulukossa 1. Näiden havaintojen valossa siis pitempään yrityksessä työskennelleet näyttivät antavan negatiivisempia arvioita välineestä.

Iivarin [Iivari, 1996] mukaan välineen kompleksisuuden kokemista voidaan vähentää antamalla hyvää ja tarkoituksenmukaista koulutusta. Saamani tulokset eivät ainakaan sodi tätä ajatusta vastaan, sillä kaikki viidestä henkilöstä, joiden mielestä MetaEdit+:n käyttö oli monimutkaista, olivat mielestään saaneet liian vähän koulutusta välineen käytöstä. Tuntimäärältäänkin heidän saamansa koulutus oli melko vähäistä, tosin eräs vastaajista oli saanut koulutusta ylitse keskiarvon, mutta tästä kysyttäessä hän kertoi koulutuksen olleen väärin kohdennettua. Eräs negatiivisia arvioita antaneista vastaajista myös perusteli negatiivista asennettaan olemattomalla koulutuksella ja huonolla välineen dokumentaatiolla.

Edellisten kappaleiden tarkoitus oli tuoda ilmi, että välineen monimutkaisuuden kokeminen vaikuttaa suoraan välineen käyttäjätyytyväisyyteen. Lisäksi monimutkaisuuteen vaikuttaa tämän ja myöskin Iivarin [Iivari, 1996] tutkimuksen mukaan koulutuksen laatu ja määrä. Myös työskentelyaika yrityksessä näytti vaikuttavan käyttäjien arvioihin välineen monimutkaisuudesta. Pitempään yrityksessä olleet antoivat negatiivisempia arvioita välineestä. Hypoteesi H4 näytti tässä tapauksessa pitävän paikkansa vain osittain, sillä vain osa monimutkaisuudesta valittaneista käyttäjistä antoi negatiivisia arvioita koko välineestä. Myös hypoteesi H1 näytti pitävän paikkansa, sillä kaikkein negatiivisimpia arvioita antaneet käyttäjät työskentelivät ohjelman vanhemmalla versiolla.

6.1.3 Käyttöliittymä ja käyttäjätyytyväisyys

Käyttöliittymä yleensä on erittäin tärkeä elementti puhuttaessa käyttäjätyytyväisyydestä, sillä käyttöliittymä on ohjelmiston osa, jonka kanssa käyttäjä jatkuvasti on vuorovaikutuksessa. Sen vuoksi erittelen seuraavissa kappaleissa varsin tarkasti saamiani vastauksia ja lisään mukaan omia kokemuksiani välineen käytöstä. Hypoteesi H1 näytti pitävän paikkansa käyttöliittymän selkeyden kokemisen suhteen, koska kaikki vanhemman version 2.5 käyttäjät antoivat negatiivisia arvioita käyttöliittymän selkeydestä.

Koska yritykseni haluaa pitää välineen tarkan rakenteen liikesalaisuutenaan, on huomioitava, että en joka kerta pysty puuttumaan tietyn käyttöliittymäkomponentin rakenteeseen sen alimmalla tasolla, vaan on tyydyttävä tarkastelemaan sitä abstrahoiden.

Tekemäni kysely tuotti käyttöliittymän kannalta hyvin kahtiajakautunutta informaatiota. Kokonaisuudessaan vastaajat jakautuivat jälleen kahteen ryhmään, joista toisen mielestä käyttöliittymä on melko vaikeaselkoinen ja toisen mielestä puolestaan melko selkeä. Yksi vastaajista ei osannut sanoa kantaansa ja yhden mielestä käyttöliittymä oli todella vaikeaselkoinen.

Mielenkiintoista kuitenkin oli se, että kaikkien MetaEdit+:n vanhempaa versiota käyttävien mielestä käyttöliittymä oli melko vaikeaselkoinen, joka tukee selvästi hypoteesiamme H1 myöhemmän version tyytyväisemmistä käyttäjistä. Kuitenkin näyttää siltä, että uudemman 3.0-versioon siirtyminen ei tuottaisi välttämättä sen tyytyväisempiä käyttäjiä, sillä käyttöliittymää koskevia negatiivisia kommentteja vastaanotettiin yhtä lailla molempien versioiden käyttäjäkunnalta.

Yleisesti ottaen erityisesti huonoja puolia tuntui löytyvän varsin paljon, ja niistä suurimpana mainittakoon epäyhtenäisyys Windows-ympäristön kanssa ja totutun käyttölogiikan puuttuminen. Tästä huolimatta suurin osa käyttäjistä oli kuitenkin sitä mieltä, että he suoriutuivat paremmin työtehtävistään MetaEdit+:n avulla. Vain viisi kyselyyn osallistuneista 17 vastaajasta oli sitä mieltä, että he suoriutuisivat tehtävistään paremmin käsin koodaamalla. Lisää näistä vaihtoehtoisista suoritustavoista kohdissa 6.2.3 ja 6.3. Saavutetut tulokset

käyttöliittymän puutteista ovat jonkin verran ristiriidassa Finniganin et al. raportoimien tulosten kanssa, joiden mukaan uusiseelantilaisissa yrityksissä vuosina 1999 ja 2000 käytetyissä CASE-välineissä ei havaittu merkittäviä puutteita käyttöliittymän suhteen [Finnigan et al., 2000].

Seuraavissa kohdissa pohditaan ensin käyttöliittymän niitä osia, jotka häiritsivät eniten käyttäjiä, ja sen jälkeen taas niitä, joihin käyttäjät olivat tyytyväisiä. Yritän sitoa mahdollisuuksien mukaan kaikki havaintoni tämän tuotoksen alkuosassa esiteltyihin teorioihin ja muihin tutkimuksiin.

6.1.4 Käyttöliittymän puutteet

Tämän kappaleen tarkoituksena on tutustuttaa lukija lyhyesti niihin olennaisiin MetaEdit+:n käyttöliittymän elementteihin, joihin kyselyyn vastanneet käyttäjät eivät olleet tyytyväisiä. Haluan uudelleen tässä yhteydessä muistuttaa, että kyseessä ei ole yksittäisen välineen arviointi, vaan tarkoitukseni on tutkia käyttäjätyytyväisyyteen vaikuttavia elementtejä MetaEdit+:n avustuksella, jotta ne voitaisiin ottaa paremmin huomioon suunniteltaessa uusia CASE-välineitä. Haluan lisäksi mainita tässä yhteydessä sen, että tarkoitukseni ei ole seuraavissa kappaleissa painottaa Windows-käyttöjärjestelmän erinomaisuutta, vaan tuoda esiin MetaEdit+:n käyttöliittymän puutteet verrattuna siihen ympäristöön, missä sitä käytetään. Tämä siis yksinkertaisesti siksi, että tavallinen Windows-käyttäjä on tottunut tiettyihin käyttöjärjestelmän tarjoamiin toiminnallisuuksiin. Käyttäjä saattaa olla hyvinkin tyytymätön, kun jokin tälle järjestelmälle ohjelmoitu ohjelma ei näitä toiminnallisuuksia tue.

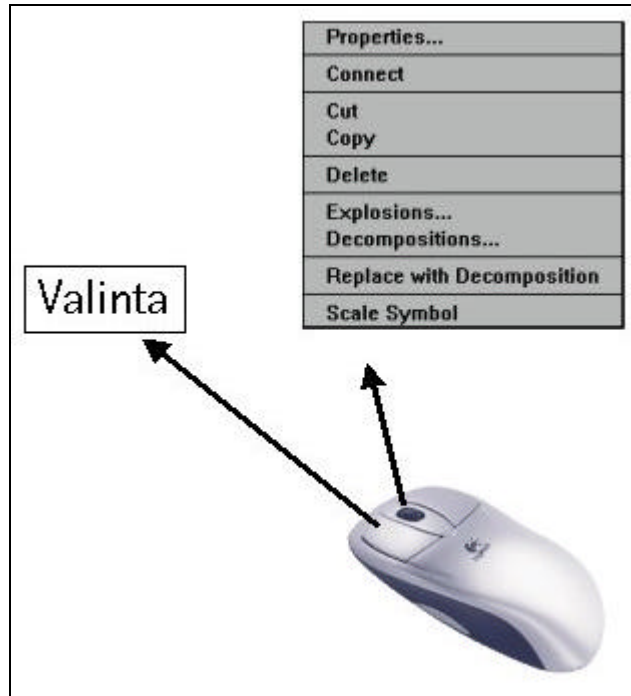
Jotkut haluavat varmasti argumentoida, että subjektiiviset näkemykset, omat näkemykseni mukaan lukien, eivät voi tarjota aina luotettavaa tietoa. Haluankin korostaa, että koko käyttöliittymän kokeminen on aina subjektiivista ja siksi keskustelu tiedon luotettavuudesta on irrelevanttia tässä yhteydessä.

Kaikkein suurimmaksi käyttöliittymän puutteeksi nousi selvästi sen puutteellinen Windows-standardien toteuttaminen. Tämä on mielestäni varsin vakava puute ohjelmassa, joka on

suunniteltu parantamaan työtehoa ja tuottamaan tuloksia. Kyselyn vapaassa osiossa viisi vastaajista nosti Windows-yhteensopimattomuuden kolmen eniten ärsyttävimmän piirteen joukkoon.

Syynä MetaEdit+:n yhteensopimattomuuteen Windows-standardien kanssa lienee se, että se on ilmeisesti siirretty Windows-ympäristöön UNIX-maailmasta. Tästä kielivät esimerkiksi ikkunoiden epästandardi ulkoasu normaaleihin Windows-sovelluksiin nähden ja puutteellinen tuki uusimmille tekniikoille, kuten esimerkiksi rullahiiren käytölle. MetaEdit+ ei myöskään noudata normaaleja syöttömetodeja. Suurin ero verrattuna standardiin Windows-syöttötekniikkaan on käyttöliittymän pohjautuminen hiiren keskimmäisen napin käyttöön. Hiiren nappien käyttö on kokonaisuudessaan mielestäni varsin epäloogista ja vaikeasti opittavaa tavalliselle Windows-standardeihin [Microsoft, 2001] tottuneelle käyttäjälle.

Hiiren vasen painike toimii melko lailla normaalia vastaavalla tavalla, sillä voidaan valita objekteja ja painaa erilaisia painikkeita. Keskimäinen nappi taas suorittaa normaalisti oikealle napille kuuluvia toimintoja. Tästä huolimatta oikealla napilla toteutetaan edelleen osalle normaalisti kuuluvista toiminnoista. Mielestäni olisi ollut huomattavasti järkevämpää pitäytyä kokonaan keskimmäisen napin käytöstä, koska huomattava osa hiiristä on edelleenkin vain kaksinappisia. Kolmannen napin painallushan on mahdollista suorittaa kaksinappisilla hiirillä molempien nappien yhtäaikaisella painalluksella, mutta tämä on usein hankalaa tarkkaa työtä tehtäessä. Lisäksi uusimmissa hiirijureissa on mahdollisuus asettaa keskimmäiselle napille jokin erikoistoiminto, joka jää käyttämättömäksi, koska MetaEdit+:n käyttö on mahdotonta ilman keskimmäistä nappia. Mielestäni kaikki keskimmäisen napin toiminnot olisi ollut helppoa toteuttaa normaalilla oikean napin kontekstipohjaisella valikkoratkaisulla.



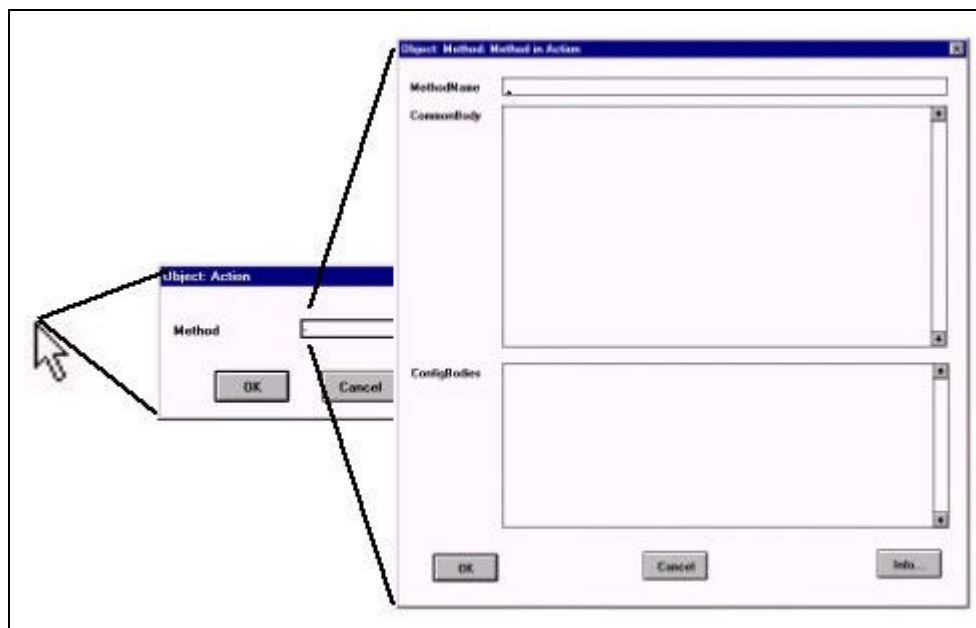
Kuvio 7. Hiiren nappien konfiguraatio MetaEdit+ -ohjelmistossa.

Myös Windows-ohjelmista tuttu peruuta-toiminto on toteutettu MetaEdit+:ssa varsin puutteellisesti, ja myös kyselyn perusteella monet vastanneista kaipasivat juuri tätä toimintoa. Tarkalleen ottaen peruuta-toiminto kyllä on olemassa, sillä ohjelma antaa mahdollisuuden hylätä kaikki edellisen talletuskerran jälkeen tehdyt muutokset. Tämä peruuta-toiminto kuitenkin edellyttäisi välitalletusta jokaisen pienimmänkin muutoksen jälkeen, mikä ei ainakaan kyselytutkimuksen valossa parantanut käyttäjätyytyväisyyttä.

Eräs Windowsin eniten käytetyistä toiminnoista on varmasti kopioi-liitä -ominaisuus, jossa käyttöjärjestelmä kopioi valitun objektin tai tekstin osan omalle leikepöydälleen, josta se sitten on liitettävissä mihin tahansa leikepöydän käyttöä tukevaan ohjelmaan. Useat kyselyyn vastanneista valittivat tämän ominaisuuden puutteellista toteutusta MetaEdit+ -ympäristössä. Heidän mielestään ohjelma toimi epäloogisesti siinä suhteessa, että joitain elementtejä oli mahdollista kopioida ja joitain taas ei. Itse asiassa asian laita on niin, että ainoastaan yhden suunnitteluelementin kopiointi on mahdollista rikkomatta tuotetun ohjelmakoodin toiminnallisuutta. Muiden elementtien kopiointi on toteutettu monimutkaisella "File in patches" -menetelmällä. Tässä menetelmässä käyttäjä avaa erillisen valikon, jossa näkyvät kaikki tietokannan suunnitteluelementit. Näistä elementeistä on sitten valittava ne, jotka haluaa kopioiduksi ja ohjelma generoi lopputuloksena .pat-tunnisteisen tiedoston. Jos haluaa siirtää tietoja eri tietokantojen välillä, on tämä -pat-tiedosto tuotava kohdetietokantaan

erillisen valikon kautta. Prosessissa kuluu huomattavan paljon enemmän aikaa normaaliin kopioi-liitä -toiminnon suorittamiseen verrattuna, ja lisäksi tietokantojen stabiilius kärsii huomattavasti tämän toiminnon käyttämisen yhteydessä. Edellä mainittujen seikkojen vuoksi kopiointi eri tietokantojen välillä olikin haastattelujen perusteella hyvin harvinaista. Yleisempää ja nopeampaa oli yksinkertaisesti luoda toteutukset aina uudelleen jokaisen uuden mallin yhteydessä. Haluan muistuttaa tässä yhteydessä, että nämä epäloogisuudet kopiointitoiminnoissa saattavat yhtä hyvin johtua MetaEdit+:n pohjalta luodusta yrityksen omasta suunnitteluympäristöstä, eli nämä tulokset eivät välttämättä ole yleistettävissä muihin tämän välineen pohjalta tehtyihin järjestelmiin.

Tutkimukseen vastanneista viisi valitti ohjelman valtaavan koko näytön käyttöönsä ja jättävän Windows-palkin omien ikkunoidensa alle. Käyttäjien mukaan ainoa tapa saada ikkunat pois alapalkin päältä oli suurentaa ne koko ruudun kokoisiksi. Käyttäjät myös valittelivat standardien vasemman yläkulman nappien puuttumista joissain tapauksissa.



Kuvio 8. MetaEdit+:n objektien manipulointi-ikkunat.

Ohjelman toimiessa oletusasetuksilla se avaa käyttäjälle neljä erilaista ikkunaa, joilla ei ole mitään tiettyä asemointia toisiinsa nähden. Joku voisi nähdä tämän skenaarion ohjelmoijaa vapauttavana ja paremman räätälöinnin mahdollistavana tekijänä, mutta omien kokemusteni mukaan siitä seuraa enemmän haittaa kuin hyötyä. Esittelen seuraavassa lyhyen esimerkin.

Tämän esimerkin tarkoituksena on kuvata, kuinka MetaEdit+:n ikkunointikäytännöt toimivat ja osoittaa lukijalle niiden monimutkainen rakenne. Kirjoitaututtaessa MetaEdit+:n istuntoon sisään käyttäjälle avautuu ensimmäiseksi ohjelman perusikkuna. Tämän perusikkunan avulla käyttäjällä on mahdollisuus suorittaa tietokannalle erilaisia toimintoja. Perusikkunan lisäksi ohjelma avaa oletusarvoisesti pienen ikkunan, josta käyttäjä näkee valitun tietokannan ja sen tilan kyseisellä hetkellä. Oletetaan, että käyttäjällä on perusikkunan lisäksi vielä auki kolme sovellusikkunaa, joissa hän manipuloi mallin tiettyjä osia. Lisäksi käyttäjällä täytyy olla auki ikkuna, josta hän voi valita luomiaan malleja muokattavaksi. Tällä hetkellä siis ikkunoiden yhteismäärä on kuusi. Kaiken tämän lisäksi jokaisella sovellusikkunalla on oma ikkunansa työkalupalkille, josta malliin lisättävät komponentit valitaan. Tämä tekee jo yhteensä yhdeksän erilaista ikkunaa. Tilannetta vaikeuttaa se, että ikkunoilla ei ole mitään erilaisia nimiä vaan ne kaikki on nimetty geneerisesti. Nämä havainnot tukevat Jarzabekin ja Huangin ajatuksia siitä, että CASE-välineiden käyttöliittymät ovat liian monimutkaisia [Jarzabek ja Huang, 1998].

Kyselyn ja omien kokemusteni perusteella ikkunoiden hallintaa vaikeuttaa erityisesti se, että tarkasteltaessa minkä tahansa komponentin ominaisuuksia, avaa ohjelma kaksi uutta ikkunaa tätä varten, eikä näiden ikkunoiden ollessa auki käyttäjä pääse tekemään mitään muita operaatioita tietokannassa tai muissa ikkunoissa. Tämä rajoitus saattaa tietysti osaltaan johtua tietokantaoperaatioihin kohdistetuista rajoituksista, mutta kunnollisen kopiointitoiminnon puuttuessa käyttäjän muistia kuormitetaan huomattavan paljon. Ikkunoiden toimintaa on havainnollistettu kuviossa 8. Kyselyssä nousi myös esille MetaEdit+:n tietokantaoperaatioiden ja käyttöliittymän hidas toiminta. Tämä onkin totta erityisesti suurempien tietokantojen ollessa kyseessä. Käyttäjien mukaan ohjelman nopea toiminta vaatii kohtuuttoman nopeiden PC-koneiden käyttöä.

MetaEdit+:n työkaluvalikoimaan kuuluu "Action"-komponentti, johon ohjelmoija voi lisätä omia aliohjelmiaan. Järjestetyn kyselyn mukaan tämän käyttöliittymäkomponentin käyttö oli joillekin epämiellyttävää, koska sen käyttö poikkeaa ohjelman muiden komponenttien käytöstä tietyiltä osin. Kun käyttäjällä on tavallisesti mahdollisuus valita muuttuja suoraan listasta suunnittelupöydälle, on suunnittelijan näiden aliohjelmakomponenttien toteuttamisessa tyydyttävä omaan muistiinsa, koska aliohjelmiin ei voi valita muualla esiteltyjä komponentteja suoraan. MetaEdit+ ei millään tavalla tarkasta ohjelmoijan

aliohjelmiin lisäämiä muuttujia. Tämän vuoksi käyttäjät tekivät usein virheitä muuttujien nimissä. Tilannetta vaikeutti erityisesti se, että MetaEdit+ ei anna käyttäjän manipuloida editori-ikkunan alla olevia komponentteja, ja näin tarkistaa muuttujien nimiä. MetaEdit+:n mukana toimitettava editori on käyttäjien mukaan ominaisuuksiltaan Windowsin Muistiota alkeellisempi ja sen koko on perusasetuksilla noin kuudesosa kuvaruudun koosta. Editori on kiinteä, eli ohjelmisto ei tarjoa mahdollisuuksia sen korvaamiseksi jonkun muun valmistajan editorilla. Kuten Jarzabek ja Huang toteavat [Jarzabek ja Huang, 1998], räätälöitävyys on ominaisuus, joka CASE-välineistä usein puuttuu. MetaEdit+:n räätälöitävyydestä puhutaan tarkemmin kohdassa 6.1.5.

Käyttäjät myös valittivat edellä mainittujen attribuuttien epäloogisesta käytöstä. Vaikka käytetyt attribuutit tallentuivat tietokantaan, saattoi niiden sieltä esiin saaminen olla varsin hankalaa. Kun tietokannan oli kerran käynnistänyt, ei ollut itsestään selvää, että kaikki muuttujat olisivat heti saatavilla. Useiden käyttäjien havaintona oli ollut se, että jonkin tietyn muuttujan löytyminen saattoi edellyttää sellaisen kuvan avaamista, jossa kyseistä muuttujaa oli jo käytetty.

Myös MetaEdit+:n objektien käsittely aiheutti kyselyyn vastanneille tyytymättömyyden tunteita. Lähekkäin olevista objekteista oli heidän mukaansa vaikea saada valittua oikeaa ja sen vuoksi objektit saattoivat siirtyä helposti vastoin käyttäjän tahtoa väärin paikkoihin. Lisäksi normaalisti maalaamalla ei voitu valita useampaa objektia kerralla, vaan tämän toiminnon suorittamiseen oli käytettävä hiiren napin lisäksi näppäimistön *control*-näppäintä. Tutkimuksessa kävi myös ilmi se, että tuotetun mallin eri versioiden vertailu kuvien perusteella oli vaikeaa. Käyttäjät kaipasivat parempia menetelmiä tähän tarkoitukseen.

Eräs MetaEdit+:n heikoimmista ominaisuuksista oli kyselyn mukaan sen epävakaa toiminta, joka on minulle henkilökohtaisestikin tuttua vanhemman 2.5-version kohdalla. Käyttäjien mukaan ohjelma suoritti usein virhetoimintoja, joiden seurauksena tietokanta saattoi vioittua jopa korjauskelvottomaksi. Usein täysin ennalta varoittamatta ja tallennustoimintoa tarjoamatta tietokanta saattoi esittää virheilmoituksen ja sulkea itsensä. Tällöin ei käyttäjien mukaan ollut muuta vaihtoehtoa kuin yrittää rekonstruoida tietokanta ja toivoa, että edellisen session lopetusta ennen talletetut tiedot olisivat vielä tallella. Käyttäjät kokivat erityisen ärsyttäväksi jatkuvien varatalletusten tekemisen.

Edellisissä kappaleissa tarkoitukseni oli tuoda esiin niitä piirteitä, joihin MetaEdit+:n käyttäjät olivat kyselyn ja haastatteluiden mukaan tyytymättömiä. Kuten kohdan pituudesta voidaan päätellä, ei käyttöliittymään oltu kovinkaan tyytyväisiä ainakaan suorittamani kyselyn perusteella. Käyttäjien mielestä ohjelma ei aina toiminut, vaikka se siltä näyttäisikin ja jotkut kokivat eri suunnitteluelementtejä kuvaavat ikonit epähavainnollisiksi. Eräiden mielestä ohjelmalla toteutettuja malleja sai tutkia liian kauan, ennen kuin niiden toiminta oli ymmärrettävää. Toisaalta taas 12 vastaajista koki selviytyvänsä paremmin työtehtävistään MetaEdit+:n avulla, joten ei käyttöliittymä näyttänyt kuitenkaan kokonaan estävän käyttäjätyytyväisyyden muodostumista. Vaikka käyttöliittymän kokeminen on hyvin subjektiivinen asia, on olemassa tiettyjä hyvän käyttöliittymäsuunnittelun elementtejä, kuten esimerkiksi ihmisen käsityskyvyn 7+2 sääntö. MetaEdit+:n käyttöliittymän hyvistä puolista enemmän seuraavassa kohdassa.

6.1.5 Käyttöliittymän vahvat puolet

Edellä kävimme läpi niitä käyttöliittymän epämiellyttäviä ominaisuuksia, jotka estivät käyttäjätyytyväisyyden syntymistä. Seuraavissa kappaleissa luomme katsauksen niihin MetaEdit+:n ominaisuuksiin, jotka tutkimukseni mukaan ovat käyttäjille eniten mieleen. Näitä ominaisuuksia olivat esimerkiksi käyttöliittymän graafisuus ja tiettyjen perustoimintojen nopeus ja helppous.

Ehdottomasti tyytyväisimpiä käyttäjät olivat MetaEdit+:n havainnollisuuteen ja graafiseen toteutukseen. Kyselyyn vastanneista 12 listasi tämän ominaisuuden MetaEdit+:n kolmen parhaimman ominaisuuden joukkoon. Heidän mielestään graafinen toteutus auttaa hahmottamaan tuotetun ohjelmakoodin toiminnallisuutta ja ymmärtämään tavallista nopeammin muiden tuottaman koodin toimintaa. Suurien kokonaisuuksien ymmärtäminen ja muokkaaminen oli myös vastaajien mielestä helpompaa graafisessa ympäristössä. Myös täysin vapaata komponenttien asettelua pidettiin hyvänä ja luovuutta edistävänä tekijänä.

Kyselyyn vastanneista 11 oli sitä mieltä, että MetaEdit+:n avulla heidän koodin tuottamisensa oli nopeampaa, helpompaa ja aikaa säästy muihin työtehtäviin. Myös erilaisten muutosten teko ohjelmakoodin toimintaan oli helpompaa MetaEdit+:n avulla kuin perinteisesti käsin koodaamalla. Vastaajien mielestä Ohjelman käytön oppimisen jälkeen sillä oli todella helppoa

ja nopeaa luoda rutiininomaisia ja yksinkertaisia toteutuksia. Oppimiskynnyksestä ja oppimiseen kuluneesta ajasta ja niiden merkityksestä käyttäjätyytyväisyyteen puhutaan enemmän kohdissa 6.2.2 sekä 6.3.1.

Näiden kappaleiden myötä olemme käyneet läpi MetaEdit+:n käyttäjätyytyväisyyteen ja käyttöliittymään vaikuttavat seikat, jotka tulivat esiin tässä tutkimuksessa. Yhteenvetona hyvistä puolista voidaan sanoa se, että vaikka niitä määrällisesti ei ollutkaan yhtä paljon kuin huonoja, olivat ne kuitenkin laadultaan parempia ja painoarvoltaan negatiivisia puolia painavampia, koska suurin osa käyttäjistä kuitenkin koki välineen käytön mieluisaksi ja kertoi sen parantavan heidän työsuoritustaan.

6.1.6 Koodigeneraattorin toiminta

CASE-välineiden olennaisista osista yksi on koodigeneraattori, sillä juuri se luo yhteyden todellisen ohjelmakoodin ja tuotettujen graafisten suunnittelumallien välille. MetaEdit+ ei sisällä itsessään koodigeneraattoria vaan se tuottaa prolog-kielisiä tiedostoja, joista varsinainen käännettävä ohjelmakoodi on koostettava. Tutkimassani yrityksessä oli luotu juuri heidän tarpeisiinsa ja arkkitehtuuriinsa sopiva koodigeneraattori. Kyselyssä tästä generaattorista nousi esiin monenlaisia mielipiteitä, sekä huonoja että hyviä. Pohdin seuraavissa kappaleissa näistä tärkeimpiä.

Parhaimpina koodigeneraattorin ominaisuuksina kyselyyn vastanneet käyttäjät pitivät sen standardoitua rakennetta, jonka vuoksi kaikkien tuottama ohjelmakoodi oli yhdenmukaista ja täytti samat vaatimukset. Useat vastanneista olivat myös sitä mieltä, että tällainen lähestymistapa poistaa systemaattisia ohjelmointivirheitä.

Generaattorin toiminnassa oli kuitenkin monien vastanneiden mielestä selviä puutteitakin. Erityisen hankalaa vastaajien mielestä oli se, että koodi generoituu hieman eri tavalla eri generointikerroilla. Tästä seuraa se, että kahden eri version kooditason vertailu on mahdotonta, koska ne ovat joka tapauksessa aivan erilaisia. Myöskin virhetilanteiden hallinta oli kyselyn perusteella toteutettu käyttäjiä tyydyttämättömällä tavalla. Erityisesti virheilmoitusten epäinformatiivisuus koettiin häiritsevänä tekijänä. Käyttäjien mukaan

virheilmoitusten perusteella oli täysin mahdoton selvittää vian todellista olinpaikkaa, koska virheilmoitukset viittasivat johonkin dokumentoimattomaan virhekoodiin.

Ohjelmoijan kannalta kaikkein ikävimmäksi puoleksi koettiin se, että koodi ja kuvat eivät aina vastanneet toisiaan. Tämä saattaa tietysti johtua MetaEdit+:n ominaisuuksista, mutta se uhkaa kuitenkin romuttaa koko CASE-välineen idean, koska virhetilanteissa koodia on kuitenkin osattava lukea.

Koodigeneraattorin luoma koodi koettiin usein hankalaksi ymmärtää. Itse syötetyistä funktioista generaattori tekee oman kooditiedostonsa, jossa useamman kuin kerran käytetyt aliohjelmat on epäloogisesti generoitu useampaan kertaan. Kääntäjähän eivät toki käännä jokaista funktiota kuin kerran, mutta tämä kasvattaa kooditiedostojen kokoa ja vaikeuttaa generoidun koodin hallittavuutta ja luettavuutta. Toinen koodigeneraattorin ikävä puoli oli se, että parametrit generoituvat satunnaisesti eri järjestykseen eri generointikerroilla, joten jokaisen generointikerran jälkeen kaikki generoidut tiedostot on otettava käyttöön, vaikka kaikkiin niihin ei olisikaan tullut muutoksia. Erityisesti eri versioiden vertailu koettiin hankalaksi juuri näiden ongelmien vuoksi. Useat vastaajista valittivat myös objektisuuntautuneen C-kielen (OOC) vaikeuttavan generoidun ohjelmakoodin ymmärtämistä.

Tässä kohdassa käsiteltiin siis koodigeneraattorin toimintaa käyttäjätyytyväisyyden näkökulmasta katsottuna. Pahimmiksi ongelmiksi nousivat käyttäjien palautteen perusteella generaattorin vaikeaselkoiset virheilmoitukset ja generaattorin tuottaman koodin vaikeaselkoisuus. Lisäksi käyttäjille tuotti vaikeuksia vertailla eri versioiden koodia, koska koodigeneraattorin tuottaman koodin järjestys vaihteli generointikerroittain. Kokonaisuudessaan koodigeneraattori koettiin hyvin toimivaksi pienistä ongelmista huolimatta.

6.1.7 Uudelleenkäyttö ja yhteensopivuus

Tässä kohdassa tarkoituksena on arvioida MetaEdit+:n uudelleenkäyttömahdollisuuksia ja yhteensopivuutta muiden yrityksessä käytössä olevien työkalujen kannalta. Lisäksi tutkitaan

hypoteesia H1, eli oliko eri versioiden käyttäjillä erilainen mielipide uudelleenkäytöstä ja yhteensopivuudesta.

MetaEdit+ -ohjelmointiympäristö on varsin avoin ympäristö, johon jokainen yritys voi erikoistaa omien tarpeidensa mukaisen CASE-välineen. Tästä johtuen voisimme olettaa, väline olisi varsin yhteensopiva yrityksen käyttäminen muiden välineiden kanssa. Kyselyn avulla saatujen tulosten mukaan tämä olikin tilanne melkein kaikkien vastaajien mielestä. Tämä poikkeaa jonkin verran Stobartin et al. saamista tuloksista [Stobart et al., 1993], joissa eräänä suurimpana ongelmana CASE-välineiden käytössä tuli esiin juuri eri työvaiheisiin tarkoitettujen välineiden yhteensopimattomuus toistensa kanssa. Syynä tähän voidaan ehkä pitää yksinkertaisesti sitä, että välineet ovat ainakin tämän tutkimuksen valossa kehittyneet merkittävästi tässä suhteessa.

Kyselyyn vastanneista kolme oli sitä mieltä, että työkalu tukee hyvin uudelleenkäyttöä ja kymmenen henkilöä vastasi sen tukevan melko hyvin ohjelmakoodin uudelleenkäyttöä. Yhteensä yli kaksi kolmasosaa vastaajista antoi positiivisia arvioita uudelleenkäytöstä. Vain kaksi vastaajaa oli sitä mieltä, että MetaEdit+ tukee melko huonosti uudelleenkäyttöä. Kaksi kyselyyn vastanneista ei puolestaan osannut ilmaista kantaansa tämän kysymyksen suhteen.

Uudelleenkäytettävyyteen liittyviä ominaisuuksia väline ei kuitenkaan toteuta kovinkaan monipuolisesti. Edellä esitelty "*File in patches*" -ominaisuus on ainoa mahdollisuus siirtää tietoja eri tietokantojen välillä. Ohjelma tarjoaa myös kopiointimahdollisuuden, mutta tämä ominaisuus ei ole kovinkaan johdonmukainen, sillä se luo ainoastaan uudet instanssit tietokannan olioista, eikä suinkaan uusia olioita tietokantaan. Kopioiden ominaisuuksia muuttaessa myös alkuperäisten olioiden ominaisuudet muuttuvat, mikä tekee kopioinnista melko vaikeasti hyödynnettävän toiminnon. Uudelleenkäyttöön liittyvissä asioissa ei havaittu mitään eroja eri MetaEdit+ -versioiden käyttäjien välillä, joten emme saaneet tukea hypoteesille H1 ainakaan tässä kohdassa.

Vastaajien mielestä MetaEdit+:n tulisi olla vielä enemmän integroitu muihin välineisiin, jotta työvaiheiden määrää suunnitteluvaiheesta itse tuotettuun koodin voitaisiin vähentää. Nykyisessä toteutuksessa kuvan suunitellaan ensin graafisella välineellä, sitten niistä generoidaan prolog-ohjelmakoodi, josta puolestaan tuotetaan kolmannen sukupolven ohjelmointikielen koodia. Tämä koodi on sitten käännettävä jollain siihen sopivalla

kääntäjällä. Myös Finnigan et al. kertovat tutkimuksessaan käyttäjien olleen tyytymättömiä muiden heidän käyttämiensä välineiden kanssa [Finnigan et al., 2000].

Griffithsin [Griffiths, 1994] mukaan ensimmäinen askel parempien CASE-välineiden tuottamisessa on eri välineiden integroiminen toimivaksi kokonaisuudeksi. Kun vastaajilta kysyttiin, kuinka MetaEdit+ heidän mielestään toimii yhteistyössä muiden välineiden kanssa, olivat vastaukset jälleen kerran melko kirjavia. Vastaukset jakautuivat tasaisesti kahteen leiriin, kuusi vastaajista oli sitä mieltä, että yhteensopivuus muiden työkalujen kanssa oli huono tai melko huono, mutta seitsemän puolestaan sanoi yhteensopivuuden olevan melko hyvä. Kolme henkilöä ei osannut ilmaista kantaansa tässä asiassa. Vastausten vaihtelevuuteen voidaan identifioida useitakin syitä. Ensinnäkin eri projekteissa saattaa olla käytössä erilaisia välineitä, jotka voivat olla tai olla olematta hyvin yhteensopivia MetaEdit+:n kanssa. Toiseksi version 2.5 käyttäjistä ainoastaan yksi oli sitä mieltä, että yhteensopivuus oli melko hyvä, mikä puolestaan tukee hypoteesia H1, jonka mukaan uudemman version käyttäjät ovat tyytyväisempiä.

Kyselyn tulosten mukaan suurin osa vastaajista antoi positiivisia arvioita MetaEdit+:n uudelleenkäyttöominaisuuksista. Myös Finnigan et al. raportoivat tutkimuksessaan vastaavista havainnoista [Finnigan et al., 2000]. Yhteensopivuudesta muiden välineiden kanssa oltiin kuitenkin jonkin verran kriittisempiä ja erityisesti vanhemman version käyttäjät olivat tyytymättömiä yhteensopivuuteen, minkä todettiin tukevan Hypoteesia H1. Käyttäjät olisivat kaivanneet suoraviivaisempaa menetelmää suunnitelluista malleista ohjelmankoodin tuotantoon ilman välivaiheita.

6.1.8 Räätelöitävyys ja tuki monelle käyttäjälle

Räätelöitävyys tarkoittaa ohjelmiston ominaisuutta, joka mahdollistaa tiettyjen ohjelmiston ominaisuuksien muuttamisen erilaisten käyttäjäryhmien tarpeita vastaaviksi. Tässä kohdassa arvioidaan MetaEdit+:n käyttäjätyytyväisyyttä räätelöinnin ja monen käyttäjän tuen kannalta. Myös asettamamme hypoteesin H1 todetaan pitävän paikkansa räätelöinnin kohdalla.

Stobartin et al. [Stobart et al., 1993] saamien tulosten mukaan käyttäjät ovat tyytyväisempiä silloin, kun CASE-välineet tarjoavat mahdollisuuden räätelöidä kehitysympäristöään

mieleisekseen. MetaEdit+ ei varsinaisesti tarjoa kovinkaan useaa mahdollisuutta tällaisten toimenpiteiden suorittamiseen. Käyttäjien haastatteluissa kävi ilmi, että nekin niukat mahdollisuudet jäivät usein käyttämättä ohjelman pikkuvikojen vuoksi.

MetaEdit+ tarjoaa mahdollisuuden käyttää X-Window -ympäristön tapaan erilaisia ikkunoiden tyyliä. Tarjolla on normaalin Windowsin lisäksi Motif- ja Mac-tyyliset vaihtoehdot. Kuitenkin käytettäessä kahta jälkimmäistä tyyliä tietyt objektien tekstit katoavat näkyvistä, joten tyylien käyttäminen ei ole kaikesta huolimatta kovinkaan käytännöllistä.

Käyttäjien kaipaamista räätälöintivaihtoehdoista kaikkein halutuin oli mahdollisuus käyttää jotakin muuta editoria kuin MetaEdit+:n peruseditoria. Tämä seikka koski erityisesti version 2.5 käyttäjiä, koska puute on korjattu versiossa 3.0. Hypoteesi H1 näyttää näiden tietojen valossa pitävän jonkin verran paikkansa räätälöinnistä puhuttaessa.

Stobartin et al. [Stobart et al., 1993] mainitsemaa tukea monelle käyttäjälle ei tekemässäni tutkimuksessa tullut lainkaan ilmi. Tosin tätä ominaisuutta varten ei oltu muodostettu omaa testikysymystä, mikä saattaa vaikuttaa asiaan. Kuitenkaan kukaan käyttäjistä ei toivonut avoimissakaan kysymyksissä tällaista toiminnallisuutta MetaEdit+ -ympäristöön. Olen itsekin sitä mieltä, että CASE-välineisiin ei tulisi lisätä ryhmäohjelmien piirteitä, koska CASE-työkalut ovat jo ennestäänkin liian monimutkaisia. Huomiota tulisi ennemminkin kiinnittää käyttäjätyytyväisyyttä parantaviin tekijöihin.

MetaEdit+ tarjoaa jonkin verran mahdollisuuksia käyttöliittymän räätälöimiseen, mutta nämä ominaisuudet jäivät käyttäjiltä käyttämättä käyttöliittymän virheiden vuoksi. Hypoteesille H1 saatiin jonkin verran tukea, koska version 2.5 käyttäjät olivat uudemman version käyttäjiä tyytymättömämpiä esimerkiksi editorin räätälöintimahdollisuuksien puuttumisen johdosta. Monen käyttäjän järjestelmiä koskevia kommentteja tai toiveita ei tässä kyselyssä tullut esiin lainkaan.

6.1.9 Menetelmäkeskeinen rakenne

Menetelmäkeskeisen rakenteen tarpeellisuudesta tai tarpeettomuudesta keskusteltiin luvussa 4.4.3. Koulukuntia oli kaksi, joista toisen mielestä menetelmäkeskeisyys on edellytys CASE-

välineille ja toisten mielestä taas se rajoittaa käyttäjien luovaa työskentelyä. Menetelmäkeskeisyydellä ymmärrän tässä yhteydessä tilannetta, jossa CASE-väline seuraa rakenteeltaan jotain tiettyä suunnittelumenetelmää.

Kuten edellä todettiin, Jarzabekin ja Huangin tutkimusten [Jarzabek ja Huang, 1998] mukaan CASE-työkalut perustuvat vielä nykyisinkin ajatukseen, jossa työkalun on tuettava vain ohjelmistokehityksen menetelmiä. MetaEdit+ on kuitenkin tässä suhteessa poikkeus, ainakin siinä muodossa, missä se on kohdeyrityksessäni toteutettu. Osaltaan tämä johtuu välineen keskittymisestä ainoastaan ydintoimintoihin (esimerkiksi varsinaista tukea suunnittelulle ei ole), mutta toisaalta voidaan perustellusti sanoa, että väline ei tarjoa mitään erityisominaisuuksia minkään menetelmän tukemiseksi.

Mielestäni tämä suunnittelutoimintojen puute saattaa kostautua projektin loppuvaiheissa, jossa usein jo toteutettuja ohjelmakoodeja on korjattava eri tavoin. Tämän vuoksi CASE-välineen tuki riittävien suunnitteludokumenttien tuottamiselle olisi ensiarvoisen tärkeää, jotta myös pidemmän ajanjakson jälkeen muistettaisiin suunnitteluratkaisuihin johtaneet syyt.

Tässä kohdassa tarkasteltiin MetaEdit+:n menetelmäkeskeisyyttä käyttäjätyytyväisyyden kannalta. MetaEdit+ ei varsinaisesti tue mitään tiettyä menetelmää, joten menetelmäkeskeisyyttä ei päästy tutkimaan sen varsinaisessa merkityksessä. Käyttäjät tuntuivat olevan tyytyväisiä ohjelman tarjoamaan vapauten suunnittelun ja toteutuksen suhteen.

6.1.10 Ehdotuksia MetaEdit+:n ominaisuuksien parantamiseksi

Vaikka tämän tutkimuksen tarkoituksena ei ole varsinaisesti arvostella mitään tiettyä välinettä, haluan tuoda julki muutaman henkilökohtaisen mielipiteeni niistä ominaisuuksista, joilla katson olevan merkitystä käyttäjätyytyväisyyden kannalta.

Ohjelman tarjoamia tarkistus- ja virheenkorjausmenetelmiä tulisi mielestäni kehittää, jotta voitaisiin saavuttaa parempi käyttäjätyytyväisyys. Tärkeimpänä mielestäni olisi toteuttaa

jonkinlainen kontrolli sen suhteen, että käyttäjä olisi kirjoittanut kaikkien muuttujien nimet oikein ja huolehtinut niiden dynaamisesta varauksesta ja vapautuksesta.

Jos aletaan visioida asioita pidemmälle, voitaisiin kehittää jopa jonkin kolmannen sukupolven ohjelmointikielten standardi, jota kyseinen CASE-väline osaisi lukea ja ymmärtää ja rekonstruoida suunnittelukuvat sen pohjalta. Tällainen käänteinen yhteensopivuus mahdollistaisi tiettyjen asioiden helpomman suorittamisen ja antaisi käyttäjille vapauden muokata koodia vuoroittaisesti sekä välineen avulla että manuaalisesti tilanteen niin vaatiessa. Tässä ajatusmallissa toteutuisivat Stobartin et. al. [Stobart et al., 1993] ajatukset välineen räätälöimisestä. Myöskin käyttäjien kommentit liian monista välivaiheista ja koko koodin uudelleengeneroimisesta pienten muutosten vuoksi puoltavat tällaista lähestymistapaa. Kuten edellä todettiin, Coallier on myös esittänyt tämän suuntaisia ajatuksia [Coallier, 1990].

Työkalun tuskastuttavimpia puolia on mielestäni puute yleisien ratkaisujen "kirjastosta". Tämä kirjasto sisältäisi malliratkaisut yleisimpien ongelmien ratkaisemiseksi. Nykymuodossa kaikki suunnitteluratkaisut on luotava jokaiseen erilliseen ohjelmaan aina uudelleen.

Koska en ole tutustunut kovinkaan läheisesti MetaEdit+:n versioon 3.0, en osaa sanoa kuinka suuri osa näistä selkeistä suunnitteluvirheistä on korjattu uudemmassa versiossa. Kaikki edellä kuvatut tekijät eivät sinänsä liity MetaEdit+:n käyttäjättyytyväisyyteen, mutta toteutuessaan esittelemäni seikat parantaisivat käyttäjättyytyväisyyttä ainakin minun kohdaltani.

6.2 MetaEdit+ ja ympäristön vaikutus

Huipputeknologiaa kehittävässä yrityksissä aikatauluilla ja tuotteiden oikea-aikaisella lanseeraamisella on hyvin suuri merkitys. Ainaisella kiireellä saattaa olla vaikutusta moniin tärkeisiin CASE-välineitä koskeviin tekijöihin. Vaikka MetaEdit+:n puutteellista dokumentointia, liian nopeata käyttöönottoa ja riittämätöntä koulutusta voidaan selittää huipputeknologisen yrityksen toimintaympäristön vaativuudella, ei näitä laiminlyöntejä voida kuitenkaan sillä puolustella. Mielestäni tämä on vain ongelman siirtämistä eteenpäin ehkä vielä kriittisemmille ajanjaksoille.

Seuraavissa kappaleissa keskitytään analysoimaan tulosten perusteella tutkimani yrityksen tarjoaman ympäristön vaikutuksia MetaEdit+:n käyttäjien tyytyväisyyteen. Selvitän asioita myös tarkemmin omien kokemusteni perusteella niissä kohdissa, joissa katson sen perehdyttävän lukijaa paremmin yrityksessä vallitseviin olosuhteisiin. Ensin pohditaan dokumentoinnin merkitystä, sitten koulutusta ja käyttöönottoa. Näiden aihepiirien jälkeen luodaan katsaus välineen käytön vapaaehtoisuuteen, organisaation rakenteellisiin elementteihin sekä johdon antamaan tukeen välineen käytölle.

6.2.1 Dokumentointi

Kuten Matthieu Dabin [Dabin, 1991] toteaa, on dokumentoinnilla koulutuksen ohella suuri merkitys käyttäjätyytyväisyyden muodostumiseen. Tämän tutkimuksen puitteissa tutkittavan CASE-välineen dokumentaatio ei juurikaan vastaajia tyydyttänyt. Seuraavissa kappaleissa pureudutaan MetaEdit+:n dokumentoinnin laatuun ja käyttäjien mielipiteisiin siitä. Lisäksi hypoteesia H1 arvioidaan dokumentoinnin kannalta.

Suurin osa vastaajista, 82 prosenttia, oli sitä mieltä, että dokumentaation taso oli ainoastaan välttävä tai peräti huono. Ainoastaan kolme henkilöä oli sitä mieltä, että dokumentaatio oli riittävää tai hyvää. Kyselytutkimuksen mukaan projektin alkuvaiheissa dokumentointi oli minimaalista ja sen oikeellisuus oli kyseenalaista. Vielä nykyäänkin paras dokumentointi on käyttäjien mielestä muiden koodien tarkasteleminen ja niiden toteutusten kopioiminen.

Mitä tulee hypoteesiin H1 eri versioiden käyttäjätyytyväisyyden eroista, voidaan todeta, että vanhemman version 2.5 käyttäjistä kukaan ei ollut sitä mieltä, että dokumentaatio olisi ollut hyvää tai edes riittävää. Kuitenkin ylipäätään dokumentaatioon tyytyväisiä olleiden vastaajien määrä oli niin pieni, että en katso tällä tuloksella olevan kovinkaan suurta merkitystä hypoteesin H1 toteutumisen kannalta.

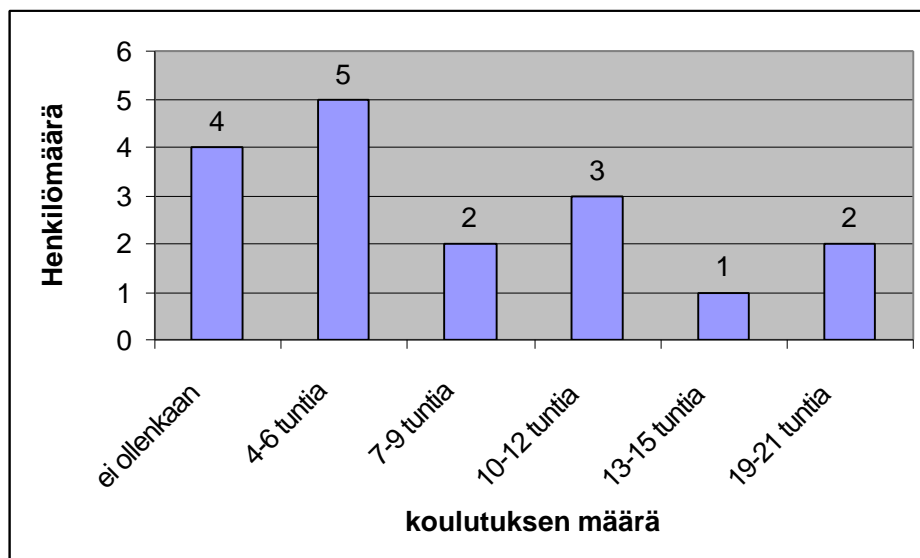
Edellisten kohtien keskeinen sisältö oli pitkälti se, että kohdeyrityksessä käytössä olevan MetaEdit+:n pohjalta tehdyn CASE-välineen dokumentoinnin taso oli varsin huono. Mitään eroja eri versioiden välillä ei havittu, joten hypoteesi H1 ei tässä yhteydessä saa perusteluja suuntaan tai toiseen.

6.2.2 Koulutus ja käyttöönotto

Edellisissä kappaleissa on puhuttu paljon koulutuksen merkityksestä käyttäjätuottavuuden muodostumisessa ja todettu sen olevan suorastaan ehto CASE-välineen menestykselliselle hyväksikäytölle. Myös Orlikowskin mukaan [Orlikowski, 1992] välineiden käyttöönotolla ja tähän varatulla ajalla suuri merkitys lopullisen käyttäjätuottavuuden muodostumiseen. Käyttäjille tulisi varata riittävästi aikaa uuden tekniikan ja metodien sisäistämiseen. Seuraavissa kappaleissa esittelen MetaEdit+⁺:n käyttöönottoon liittyviä seikkoja ja arvioin hypoteesia H2, jonka mukaan enemmän koulutusta saaneet käyttäjät olisivat tuottavampia.

MetaEdit+ työkalun käyttöönottoa ei oltu käyttäjien mukaan suunniteltu tarpeeksi. Työkalu otettiin käyttöön liian aikaisessa vaiheessa sen ollessa vielä epästabiilissa tilassa. Puutteellisesta dokumentoinnista johtuen käyttäjillä oli suuria ongelmia ohjelman käytössä. Kuten eräs vastaajista kyselyn avoimessa osiossa totesikin, alkuvaikeudet olivat suurimpana tekijänä vaikuttamassa negatiivisen kuvan syntymiseen välineestä.

Kyselyyn vastanneista yli puolet oli mielestään saanut liian vähän koulutusta välineen käyttöön. Lisäksi vastaajista neljä ei ollut saanut koulutusta lainkaan. Eniten koulutusta saaneet olivat saaneet sitä noin 20 tuntia. Useimmissa tapauksissa koulutusta oli annettu välineen käytössä neljästä kuuteen tuntia. Koulutuksen määrää kuvataan tarkemmin kuviossa 9.



Kuvio 9. Saadun koulutuksen määrä.

Asetimme edellä hypoteesin H2, jonka mukaan enemmän koulutusta saaneet olisivat tyytyväisempiä välineen käyttöön. Tämä hypoteesi ei näyttänyt ainakaan kyseisen välineen kohdalla pitävän paikkaansa. Viisi vastaajista oli sitä mieltä, että he eivät suoriudu työstänsä paremmin MetaEdit+:n avulla. Kuitenkin kaikki näistä viidestä olivat saaneet koulutusta välineen käytössä, jotkut jopa paljon keskimääräistä enemmän. Toisaalta taas kaikki ne vastaajista, jotka eivät olleet saaneet koulutusta välineen käytössä olivat melko tyytyväisiä. Tässä onkin syytä pohtia edellisten tutkimusten valossa ehkä enemmän vastaajien henkilökohtaisia ominaisuuksia, jotta saisimme selville syyn eräiden vastaajien negatiiviseen suhtautumiseen välineeseen. Näistä seikoista lisää kohdassa 6.3.

Työkalun käyttäminen oli pakollista, vaikka sen rajoitukset tunnettiin. Yhdistettynä monimutkaiseen perusarkkitehtuuriin työkalusta tuli erityisen hankala oppia ja turhauttava käyttää. On kuitenkin huomattava, että loppuvaiheessa välineen käyttö oli jo opittu niin hyvin, että pikkuvikoihin ja omutuisuuksiin ei enää kiinnitetty huomiota. Käyttäjien mielestä väline oli periaatteessa helppo oppia, mutta niin sanottua hiljaista tietoa oli käyttäjien mielestä oppimisen kannalta liikaa.

Aloittaessani MetaEdit+:n käytön, yrityksen siitä erikoistama CASE-väline oli vielä varsin keskeneräinen. Kaikkia komponentteja ei oltu vielä kehitetty ja käyttöliittymäkin oli vielä kokeiluasteella. Ajan myötä sovellus kuitenkin kehittyi ja kypsyi lopulta nykyiseen muotoonsa. MetaEdit+:n versiolle 2.5 tarkoitettua sovellusta ei olla päivitetty enää vuoden 2000 alun jälkeen.

Edellisissä kappaleissa käsiteltiin MetaEdit+:n käyttöönottoa tutkittavassa yrityksessä ja siitä annettua koulutusta. Työkalu oli vastaajien mielestä otettu liian aikaisessa vaiheessa käyttöön, mikä näkyi sen keskeneräisyytenä sekä epäluotettavana toimintana. Hypoteesimme H2, jonka mukaan enemmän koulutusta saaneet käyttäjät olisivat tyytyväisempiä, ei tuntunut pitävän paikkaansa.

6.2.3 Vapaaehtoisuus

Tutkimukseni kirjallisuuskatsauksessa esitettiin, että monissa tutkimuksissa on havaittu vapaaehtoisuuden olevan merkittävä käyttäjätyytyväisyyteen vaikuttava tekijä [Iivari, 1996; Iivari ja Maansaari, 1997]. Seuraavissa kappaleissa käsitellään vapaaehtoisuutta ja sitä koskevaa hypoteesia H3, jonka mukaan vapaaehtoisesti välinettä käyttävät olisivat tyytyväisempiä.

Kyselyyn vastanneista 17 henkilöstä neljällä oli käytössään vaihtoehtoinen väline työnsä suorittamiseen. Useimmissa tapauksissa tämä väline oli yksinkertaisesti koodin tuottaminen perinteisesti käsin ohjelmoimalla. Näistä henkilöistä kuitenkin vain kaksi oli sitä mieltä, että he työskentelisivät mieluummin ilman MetaEdit+ -välinettä. Voidaankin sanoa, että tämä tutkimus ei antanut niin kieltävää kuin puoltavaakaan aineistoa vapaaehtoisuuden merkityksestä.

Pohdittaessa edellä asetettua hypoteesia H3, jonka mukaan vapaaehtoisesti välinettä käyttävät ovat tyytyväisempiä, ei siitä kyselyn tulosten perusteella voida sanoa mitään yleispätevää. Koska ainoastaan neljällä henkilöllä oli käytössään vaihtoehtoisia välineitä, ja heistä puolet olisi mieluummin käyttänyt niitä, jää tämän hypoteesin paikkansapitävyys ratkaisemattomaksi.

Edellisten tutkimustulosten perusteella [Iivari, 1996; Iivari ja Maansaari, 1997] vapaaehtoisuus kuitenkin on esittänyt merkittävää roolia CASE-välineiden käyttäjätyytyväisyyteen liittyvissä seikoissa. Mielestäni tämän tutkimuksen tulosta saattaa osaltaan selittää vastaajien kokemattomuus muista MetaEdit+:n kaltaisista välineistä, koska ainoastaan yksi henkilö oli käyttänyt muuta CASE-välinettä. Lisäksi useat henkilöt olivat olleet yrityksen palveluksessa vain muutamia vuosia ja käyttäneet koko ajan samaa CASE-välinettä, MetaEdit+:ssa. Tästä on varmasti seurauksena asiantuntemuksen lisäksi tietynlainen kapea-alainen perspektiivi, jonka puristuksessa käyttäjät eivät välttämättä osaa kuvitella muunlaisia mahdollisuuksia.

Edellisissä kohdissa käsiteltiin vapaaehtoisuuden merkitystä kohdeyrityksessä MetaEdit+:n käytön suhteen. Valitettavasti asettamamme hypoteesit H2 ja H3 eivät saaneet vahvistusta vastausten tasaisen jakautumisen vuoksi.

6.2.4 Organisaation rakenteelliset elementit

Kirjallisuuskatsauksessa todettiin, että CASE-välineiden vaikutus on huomattavasti positiivisempi, jos käyttöönottoon liittyy myös organisaation muutoksia. Kyselytutkimukseni ei kuitenkaan ottanut kantaa tähän seikkaan, mutta esittelen seuraavassa joitain seikkoja CASE-välineen ympärille kehitetystä organisaatiosta. Valitettavasti en vaitiolovelvollisuuteni vuoksi voi käsitellä tässä yhteydessä organisaation palkitsemiskäytäntöjen vaikutusta käyttäjätyytyväisyyteen.

Käyttäjien kannalta tärkein elementti on varmasti CASE-välineen käyttöön liittyvä tutor-toiminta ja asiantuntijafoorumi. Yrityksen jokaisessa yksikössä on nimetty CASE-välineen käyttöön liittyviin kysymyksiin vastaava henkilö. Jos tämä henkilö ei osaa vastata kysymykseen, on hänen mahdollista eskaloida kysymys kaikkien yksiköiden tutor-henkilöistä koostuvaan foorumiin, jossa yleensä kaikki ongelmat löytävät ratkaisunsa. MetaEdit+:n esittelemisen yhteydessä yrityksessä ei toteutettu mitään radikaaleja muutoksia organisaation suhteen, vaan muutokset ovat tulleet enemmänkin itseohjautuvasti tarpeen niin vaatiessa.

Tämän kohdan tarkoituksena oli selvittää ne organisaation elementit, joilla on merkitystä käyttäjätyytyväisyyden kannalta. Valitettavasti en kuitenkaan voi esitellä kaikkia tähän liittyviä tekijöitä vaitiolovelvollisuuteni vuoksi. Käyttäjätyytyväisyyttä kuitenkin tässä yrityksessä nosti yksikkökohtaisen tutor-henkilön nimeäminen ja eri yksiköiden välinen tutor-henkilöiden yhteisö.

6.2.5 Johdon tuki

Johdon tukea koskevaa kysymystä en sisällyttänyt kyselylomakkeeseen, koska henkilökohtaisesti olen kokenut johdon olleen aina välineen käyttöä tukemassa. Samaa mieltä olivat myös haastattelemani käyttäjät. Käyttäjien mukaan johtoasemassa olevat henkilöt ovat

esimerkillisesti tiedostaneet uuden järjestelmän käyttöön liittyneet ongelmat ja varautuneet niiden selvittämiseen kuluvaan aikaan myös aikataulun laadinnassa. Käyttäjätyytyväisyyden kannalta johdon tuen siis voidaan siis katsoa siis olleen ainoastaan positiivinen tekijä.

6.2.6 Tuki prosessille

Ohjelmistotuotannossa on järkevää muodostaa tuotteen tyypistä riippuva prosessi tuotteen laadun ja aikataulun säilyttämisen varmistamiseksi. En tässä yhteydessä voi salassapitovelvollisuuteni vuoksi esitellä juuri kyseisessä yrityksessä käytettäviä ohjelmistotuotannon menetelmiä, mutta muutama sana välineen suhteesta prosessiin on paikallaan. CASE-välineiden menetelmäkeskeisestä rakenteesta on keskusteltu luvuissa 4.4.3 ja 6.1.9. Seuraavissa kappaleissa luon katsauksen kyselytutkimuksen tuottamiin tuloksiin käyttäen apuvälineenäni hypoteesia H1.

Tutkimassani yrityksessä tekemäni kyselytutkimus ei paljastanut juurikaan MetaEdit+:n tuesta olemassa oleville prosesseille. Vastaajista kahdeksan oli sitä mieltä, että MetaEdit+ tuki hyvin tai melko hyvin prosesseja, mutta kuusi oli puolestaan sitä mieltä, että ohjelma tuki melko huonosti käytössä olevia prosesseja. Kolme henkilöä ei osannut ilmaista kantaansa tästä asiasta.

Hypoteesin H1 (*MetaEdit+ version 3.0 käyttäjät ovat tyytyväisempiä kuin version 2.5 käyttäjät*) valossa tarkasteltaessa voidaan sanoa, että hypoteesi pitää paikkansa, sillä viidestä MetaEdit+:n version 2.5 käyttäjästä neljä oli sitä mieltä, että väline tuki olemassa olevia prosesseja melko huonosti. Syiksi puutteelliselle prosessien tuelle mainittiin katselmointipöytäkirjojen tuottamisen hankaluus ja tehtyjen muutosten vaikea dokumentoiminen.

Toisaalta on huomattava, että kaikki MetaEdit+:n vanhemman version käyttäjät työskentelivät samassa projektissa, joten jos tämän projektin menettelytavat poikkeavat merkittävästi muiden projektien vastaavista, voidaan tutkimuksessa ilmenneiden eroavaisuuksien ajatella johtuvan pelkästään tästä.

Tuki prosessille on hyvin paljon samankaltainen asia kuin CASE-välineiden menetelmäkeskeinen rakenne, mutta eri näkökulmasta katsottuna, minkä vuoksi halusin erotella nämä asiat toisistaan. Yrityksessä käytössä olevat ohjelmistokehityksen prosessit saivat jonkin verran tukea välineeltä ainakin uudemman version tapauksessa. Tämän todettiin tukevan hypoteesia H1. Menetelmäkeskeisestä rakenteesta edellä keskusteltaessa tuli ilmi, että MetaEdit+ ei varsinaisesti tukeudu rakenteessaan mihinkään tiettyyn metodologiaan ainakaan vanhemman version kohdalla, johon itse olen tutustunut. Ehkäpä uudempi versio seuraa tarkemmin jotain tiettyä menetelmää, koska sen käyttäjät antoivat positiivisempia arvioita välineen tuesta yrityksen olemassa oleville prosesseille.

6.3 MetaEdit+ ja yksilölliset tekijät

Kirjallisuuskatsauksessa todettiin, että käyttäjäytyvyäisyyteen vaikuttavien monien ulkoisten tekijöiden ohella myös käyttäjien henkilökohtaiset tekijät ovat vaikuttamassa lopputulokseen. Seuraavissa kohdissa pohditaan kyselytutkimukseen vastanneiden käyttäjien henkilökohtaisten ominaisuuksien vaikutusta käyttäjäytyvyäisyyden muodostumisessa. Ensin pohditaan oppimiskynnystä ja hypoteesien H1 ja H5 toteutumista sen tarjoamassa viitekehyksessä. Tämän jälkeen käsitellään kokeneiden käyttäjien vanhojen taitojen muuttumista arvottomiksi uuden välineen käyttöönoton jälkeen.

6.3.1 Oppimiskynnys

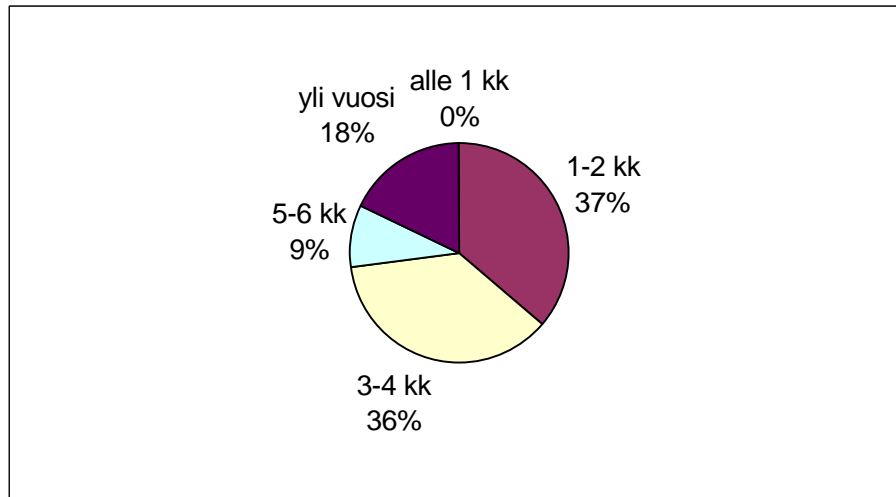
Oppimiskynnystä on käsitteenä vaikeaa arvioida ilman selkeitä kriteereitä oppimisen määrittelylle. Järjestämässäni kyselytutkimuksessa ei kuitenkaan määritely kovin tarkkaan, mitä välineen käytön oppimisella tarkoitettiin. Tämä johti osaltaan varsin hajautuneisiin vastauksiin ja tuloksen tarkkuuden kärsimiseen. Olen kuitenkin sitä mieltä, että tästäkin huolimatta tulokset tarjoavat arvokasta tietoa siitä, miten välineen käytön oppiminen vaikuttaa käyttäjäytyvyäisyyteen. Tarkoitukseni oli myös arvioida MetaEdit+:n käytön oppimista kokonaisuutena mukaan lukien MetaEdit+ ja sen pohjalta luotu yritykselle erikoistettu arkkitehtuuri. Kuitenkin väärinkäsityksien vuoksi jotkut vastaajista arvioivat vain toista näistä osista, mikä saattaa vinouttaa tutkimustuloksia. Olen kuitenkin esimerkiksi keskiarvoja laskiessani jättänyt pois nämä selkeästi muista poikkeavat arvot. Seuraavissa kappaleissa

käsitellään seikkoja, jotka liittyvät MetaEdit+:n ja siihen perustuvan yrityksessä kehitetyn arkkitehtuurin oppimiseen ja oppimiskynnykseen. Arviointikriteereinä käytetään asettamiamme hypoteeseja H1 ja H5.

Kyselystä saatujen tulosten perusteella välineen käytön oppiminen vei aikaa useimmilta yhdestä neljään kuukautta. Kuitenkin mukana oli henkilöitä joilla oppiminen kesti jopa vuoden verran. Eri versioiden välillä ei havaittu mitään eroja. Hypoteesissa H5 oletettiin välineen käytön nopeasti oppineiden olevan tyytyväisempiä, mutta tulosten perusteella tämä hypoteesi ei saanut vankkaa todellisuuspohjaa. Joissain tapauksissa nopeasti välineen käytön oppineet antoivat positiivisia arvioita käyttäjätyytyväisyyden kannalta, mutta taas toisissa tapauksissa he eivät olleet lainkaan tyytyväisiä.

Jotta voitaisiin paremmin ymmärtää mitä hypoteesissa H5 tarkoitetaan nopeasti oppimisella, määrittelen nopeasti oppineet sellaisiksi käyttäjiksi, jotka oppivat välineen käytön keskimääräistä nopeammin eli alle neljässä kuukaudessa. Hitaammin oppineet määritellään tässä tutkimuksessa sellaisiksi käyttäjiksi, jotka oppivat välineen käytön keskimääräistä hitaammin eli he olivat käyttäneet aikaa oppimiseen yli puoli vuotta.

Alle neljässä kuukaudessa välineen käytön oppineita oli kyselyn perusteella kaikkiaan kahdeksan henkilöä ja yli puoli vuotta oli kulunut kahdella vastaajista. Käytön nopeasti oppineista neljä antoi varsin positiivisia arvioita välineestä ja sen käytöstä. He suoriutuivat työtehtävistään mielestään perinteisiä menetelmiä paremmin ja MetaEdit+:n käyttö oli heille mieluisaa. Kuitenkin yksi käytön nopeasti oppineista oli sitä mieltä, että hän selviytyi paremmin käsin koodaamalla.

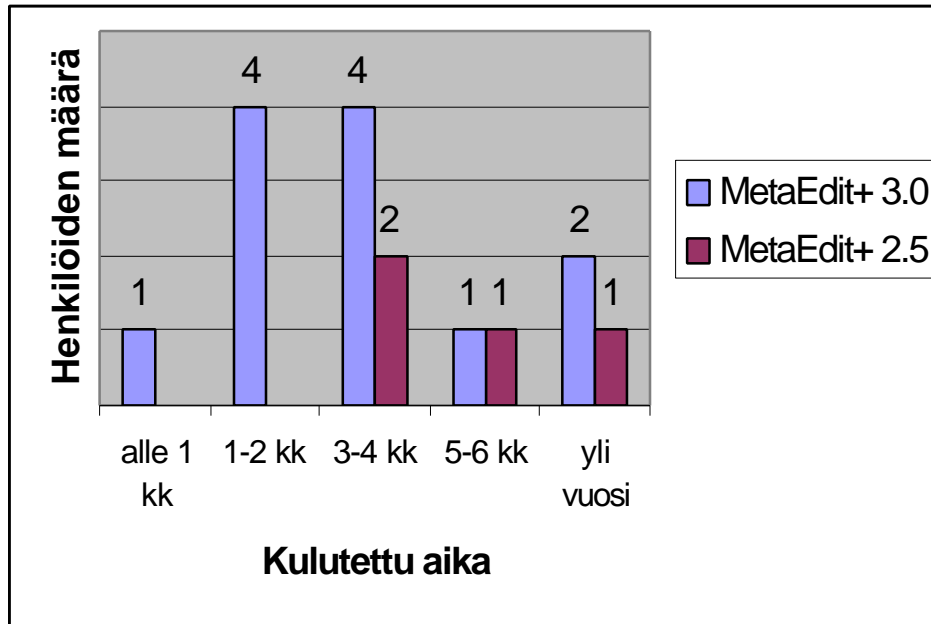


Kuvio 10. Oppimiseen kulunut aika.

Yli puoli vuotta välineen käyttöä opetelleista toinen oli sitä mieltä, että hän ei selviydy MetaEdit+:n avulla työtehtävistään perinteisiä menetelmiä paremmin. Toisen mielestä MetaEdit+ paransi työsuoritusta ja oli jopa mieluisaa käyttää. Oppimismetodit vaihtelivat kaikkien vastaajien kesken melko paljon. Suurin osa vastaajista oli oppinut välineen käyttöä koulutuksessa, itseopiskelulla ja työtovereiden opastuksella.

Edellä mainittujen tulosten perusteella ei voida mielestäni todistaa hypoteesia H5 varauksetta todeksi ainakaan tässä tutkimusotteessa. Kuitenkin näyttää siltä, että nopeammin käytön oppineet ovat suurimmalta osin tyytyväisiä välineen käyttöön.

Verrattaessa eri versioiden oppimiseen kulutettua aikaa, voidaan havaita, että uudemman version käytön oppimiseen näyttää kuluvan jonkin verran vähemmän aikaa kuin vanhemman version. Uudemman version käyttäjistä neljä oli oppinut välineen käytön alle neljässä kuukaudessa, mutta vanhemman version käyttäjistä kukaan ei ollut oppinut käyttöä niin nopeasti. Kautta linjan uudemman 3.0 version käyttäjät näyttivät keskimäärin oppineen käytön noin kaksi kuukautta nopeammin kuin vanhemman 2.5 version käyttäjät. Tutkimustulosten perusteella uudemman version käytön oppimiseen kului keskimäärin 3,5 kuukautta ja vanhemman vastaavasti hieman alle 5,5 kuukautta. Lisäksi eräs vanhemman version käyttäjistä oli todennut tarkemmin yksilöimättä oppimisen kestäneen "liian kauan". Jos oppimisen nopeutta pidetään käyttäjätyytyväisyyden kriteerinä, mitä se mielestäni on, voidaan todeta uudemman version käyttäjien olevan tyytyväisempiä tässä suhteessa. Tämä tukee asetettua hypoteesia H1.



Kuvio 11. Eri versioiden oppimiseen kulunut aika

Edellisissä kappaleissa tarkoitukseni oli tuoda esiin, miten MetaEdit+:n käyttäjät oppivat CASE-välineen käytön. Hypoteesi H1 näytti pitävän paikkansa, sillä vanhemman version käyttäjät oppivat keskimäärin hitaammin välineen käytön kuin version 3.0 käyttäjät. Hypoteesin H5 toteutumisesta ei näiden tietojen perusteella voi perustellusti sanoa mitään, sillä nopeammin oppineet eivät välttämättä olleet tyytyväisempiä kuin hitaammin edenneet. Oppimistavat olivat kaikilla melko lailla samat, sillä useimmin käyttö opittiin koulutuksen, työtovereiden opastuksen ja itseopiskelun yhdistelmänä.

6.3.2 Opittujen taitojen muuttuminen arvottomiksi

Edellä kuvailtiin, miten uuden teknologian esittely saattaa herättää negatiivisia tunteita kokeneemmissa työntekijöissä. Nämä työntekijät saattava ajatella asiantuntija-asemansa heikentyvän ja teknisen osaamisensa muuttuvan arvottomaksi [Orlikowski, 1988]. Seuraavissa kappaleissa kerrotaan, miten nämä teoriat realisoituivat kohdeyrityksessä.

Kyselyyn vastanneista suurin osa oli sitä mieltä, että MetaEdit+:n avulla he suoriutuivat työstään entistä paremmin. Kuitenkin viiden henkilön mielestä MetaEdit+ ei parantanut heidän työtehoaan, pikemminkin päin vastoin. Näistä viidestä kaksi oli ollut yrityksen

palveluksessa yli neljä vuotta ja yksi jopa yli kuusi vuotta. Mielestäni heidän negatiivista suhtautumistaan työkaluun voidaan selittää osittain Orlikowskin [Orlikowski, 1988] ja Sumnerin [Sumner, 1995] tavoin sillä, että heidän kovalla työllä hankkimansa taidot tuntuvat arvottomilta uuden välineen käyttöönoton jälkeen.

On tunnettua, että muutokset aiheuttavat ihmisissä usein vastustusta ja näin arvelen tapahtuneen myös tutkittavassa yrityksessä kyselytutkimusten vastausten perusteella. Ihmiset ovat erilaisia ja tämä tulisi mielestäni ottaa huomioon CASE-välinettä esiteltäessä. Ihmisiä tulisi motivoida uuden välineen käyttöön ja ehkä myös vapaaehtoisuus olisi avain käyttäjätyytyväisyyteen tällaisissa tapauksissa. On kuitenkin huomattava, että tutkimukseni ei aukottomasti todista syyksi juuri edellä mainittua taitojen muuttumista arvottomaksi, vaan syy voi yhtä hyvin olla varsin toisenlainenkin. Esitettyjen teorioiden valossa omat arvioni näyttävät kuitenkin pitävän paikkansa.

6.4 Tutkimushypoteesien toteutuminen

Kohdassa 5.6.3 määrittelin tutkimukselleni olennaiset tutkimushypoteesit, joiden avulla tutkittavan yrityksen käyttämän MetaEdit+ -välineen käyttäjätyytyväisyyttä arvioitiin. Hypoteesit olivat seuraavat:

- H1** MetaEdit+ version 3.0 käyttäjät ovat tyytyväisempiä kuin version 2.5 käyttäjät
- H2** Enemmän koulutusta saaneet ovat tyytyväisempiä
- H3** Vapaaehtoisesti välinettä käyttävät ovat tyytyväisempiä
- H4** Mitä monimutkaisemmaksi käyttäjä kokee välineen käytön, sitä tyytymättömämpi hän on
- H5** Mitä nopeammin käyttäjä omaksuu välineen käytön, sitä tyytyväisempi hän siihen on

Edellisissä kohdissa olen käsitellyt kunkin näiden hypoteesien toteutumista monissakin eri yhteyksissä. Tämän kohdan tarkoituksena on kerätä kaikki nämä langanpätkät yhteen ja koota hypoteesien toteutumisesta kattava kokonaiskuva. Etenemme kronologisessa järjestyksessä ensimmäisestä hypoteesista aina viimeiseen viidenteen hypoteesiin saakka.

6.4.1 Hypoteesin H1 toteutuminen

Ensimmäisen asettamani hypoteesin tarkoitus oli enemmänkin oman uteliaisuuteni tyydyttäminen kuin yleiset CASE-välineiden käyttäjätyytyväisyyteen liittyvät asiat. Vaikka tämän tutkimuksen tarkoitus ei ole eri välineiden analysointi, katsoin tarpeelliseksi tutkia, oliko kahden eri version käyttäjien mielipiteissä eroavaisuuksia käyttäjätyytyväisyyden suhteen. Erityisesti asia kiinnosti itseäni henkilökohtaisesti sen vuoksi, että tähän mennessä olen käyttänyt ainoastaan vanhempaa versiota.

Tutkimustulosten perusteella MetaEdit+:n kahden eri version välillä todellakin oli joitain merkittäviä ja todellisia eroja käyttäjätyytyväisyyden kannalta mitattuna. Suurimpana erona uudemman version hyväksi paljastui tutkimuksessa käyttöliittymän selkeyden kokeminen. Kaikki vanhemman version käyttäjät pitivät käyttöliittymää melko vaikeaselkoisena, kun taas uudemman version käyttäjistä negatiivisia arvioita antoi ainoastaan 25 prosenttia. Käyttöliittymä ylipäättänsä koettiin melko lailla tasaisesti samoilta osa-alueilta puutteelliseksi kummankin version käyttäjäkunnan piirissä. Näistä osa-alueista keskusteltiin jo tarkemmin kohdassa 6.1.4. Taulukossa 2 on esitetty ristiintaulukoituna eri versioiden käyttäjien mielipiteet MetaEdit+:n käyttöliittymän selkeydestä.

	Versio 3.0	Versio 2.5
selkeä	0 0.00%	0 0.00%
melko selkeä	8 66.67%	0 0.00%
en osaa sanoa	1 8.33%	0 0.00%
melko vaikeaselkoinen	2 16.67%	5 100.00%
vaikeaselkoinen	1 8.33%	0 0.00%
yhteensä	12	5

Taulukko 2. MetaEdit+:n käyttöliittymän selkeyden kokeminen. Eri vastausvaihtoehtojen frekvenssit ja suhteelliset frekvenssit on ristiintaulukoitu eri versioiden välillä.

Vanhemman version 2.5 käyttäjät näyttivät myös kokevan välineen käytön monimutkaisemmaksi kuin uudemman version käyttäjät (Taulukko 3). Lisäksi vanhemman version käyttäjiltä kului selvästi kauemmin aikaa omaksua välineen käyttöön liittyvät tiedot ja taidot kuin uudemman version käyttäjillä. Keskiarvolla mitattuna uuden version käyttäjät omaksuivat välineen keskimäärin 3,5 kuukaudessa, kun taas vanhemman version käyttäjät kuluttivat aikaa noin kaksi kuukautta enemmän.

	Versio 3.0	Versio 2.5
vaikeita ymmärtää	1 8.33%	0 0.00%
melko vaikeita ymmärtää	2 16.67%	2 40.00%
en osaa sanoa	2 16.67%	1 20.00%
melko helppoja ymmärtää	6 50.00%	2 40.00%
helppoja ymmärtää	1 8.33%	0 0.00%
yhteensä	12	5

***Taulukko 3.** MetaEdit+*:n* monimutkaisuuden kokeminen. Eri vastausvaihtoehtojen frekvenssit ja suhteelliset frekvenssit on ristiintaulukoitu eri versioiden välillä.*

Yhteensopivuutta arvioivien kysymysten valossa uudemman version käyttäjät olivat tyytyväisempiä. Vanhemman version käyttäjistä ainoastaan yhden mielestä yhteensopivuus oli melko hyvä muiden yrityksessä käytettyjen välineiden kanssa, kun taas uudemman version käyttäjistä puolet oli tätä mieltä.

	Versio 3.0	Versio 2.5
huono	0 0.00%	2 40.00%
melko huono	3 25.00%	1 20.00%
en osaa sanoa	3 25.00%	1 20.00%
melko hyvä	6 50.00%	1 20.00%
hyvä	0 0.00%	0 0.00%
yhteensä	12	5

***Taulukko 4.** MetaEdit+ :n yhteensopivuuden kokeminen. Eri vastausvaihtoehtojen frekvenssit ja suhteelliset frekvenssit on ristiintaulukoitu eri versioiden välillä.*

Hypoteesi H1 tuntui pitävän paikkansa myös käsiteltäessä MetaEdit+ :n tukea yrityksessä käytössä oleville prosesseille, sillä 80 prosenttia vanhemman version käyttäjistä piti yhteensopivuutta melko huonona, kun taas uudemman version käyttäjistä tätä mieltä oli ainoastaan 17 prosenttia. Vastaukset on esitetty ristiintaulukoituna eri versioiden välillä taulukossa 5.

	Versio 3.0	Versio 2.5
hyvin	2 16.67%	0 0.00%
melko hyvin	5 41.67%	1 20.00%
en osaa sanoa	3 25.00%	0 0.00%
melko huonosti	2 16.67%	4 80.00%
huonosti	0 0.00%	0 0.00%
yhteensä	12	5

***Taulukko 5.** MetaEdit+ :n tuki yrityksessä käytössä oleville prosesseille ja käytännöille. Eri vastausvaihtoehtojen frekvenssit ja suhteelliset frekvenssit on ristiintaulukoitu eri versioiden välillä.*

Yhteenvedon hypoteesin H1 toteutumisesta voidaan sanoa, että se piti monilla osa-alueilla paikkansa. Kuitenkin on edelleenkin huomioitava, että kaikki eroavaisuudet eri versioiden välillä olivat enemmän tai vähemmän kosmeettisia käyttöliittymään ja ulkoisiin ominaisuuksiin liittyviä tekijöitä. Ohjelman perustoiminta ja ydin ovat säilyneet kuta kuinkin muuttumattomina, samoin kuin suurimmat ongelmat käyttöliittymän toteutuksessa.

6.4.2 Hypoteesin H2 toteutuminen

Hypoteesi H2:n mukaan oletimme enemmän koulutusta saaneiden käyttäjien olevan tyytyväisempiä välineen käyttöön. Kuitenkaan ainakaan tällä kyseisellä otoksella emme saaneet tätä hypoteesia tukevia tuloksia, sillä vähemmän koulutusta saaneet eivät olleet yleisesti ottaen yhtään tyytymättömämpiä kuin enemmän koulutusta saaneet. Käyttäjät, jotka eivät olleet saaneet laisinkaan koulutusta välineen käyttöön näyttivät jopa joissain tapauksissa olevan huomattavasti tyytyväisempiä kuin jonkin verran koulutusta saaneet käyttäjät. Käyttäjien mielipiteet välineen käyttämisestä ja sen vaikutuksesta työsuoritukseen on esitetty taulukoissa 6 ja 7 ristiintaulukoituna annetun koulutuksen suhteen.

	0 h	0 - 10 h	11 - 21 h
hyvin mieluisaa	0 0.00%	0 0.00%	0 0.00%
mieluisaa	3 75.00%	3 37.50%	3 60.00%
en osaa sanoa	1 25.00%	4 50.00%	1 20.00%
vastenmielistä	0 0.00%	1 12.50%	1 20.00%
hyvin vastenmielistä	0 0.00%	0 0.00%	0 0.00%
Yhteensä	4	8	5

Taulukko 6. MetaEdit+ :n käyttämisen kokeminen. Eri vastausvaihtoehtojen frekvenssit ja suhteelliset frekvenssit ristiintaulukoituna annetun koulutuksen suhteen.

	0 h	0 - 10 h	11 - 21 h
hyvin negatiivisesti	0 0.00%	0 0.00%	0 0.00%
negatiivisesti	0 0.00%	2 25.00%	1 20.00%
en osaa sanoa	2 50.00%	5 62.50%	2 40.00%
positiivisesti	2 50.00%	1 12.50%	2 40.00%
hyvin positiivisesti	0 0.00%	0 0.00%	0 0.00%
Yhteensä	4	8	5

Taulukko 7. MetaEdit+ :n vaikutus työsuoritukseen. Eri vastausvaihtoehtojen frekvenssit ja suhteelliset frekvenssit ristiintaulukoituna annetun koulutuksen suhteen.

Koska monissa tutkimuksissa on havaittu koulutuksen lisäävän käyttäjätyytyväisyyttä [Ho, 1992; Jones, 1992], oli syytä miettiä, mitkä seikat johtivat tässä tutkimuksessa siihen, että koulutus ei välttämättä näyttänyt lisäävän käyttäjätyytyväisyyttä. Ensimmäisenä esiin nousivat kyselyyn osallistuneiden yksilölliset seikat, etenkin koska kyse oli varsin pienestä otoksesta. Näistä seikoista keskusteltiin kohdassa 6.3.2. Toisena mahdollisena selityksenä asialle voitaisiin pitää Iivarin [Iivari, 1996] huomautusta siitä, että väärin toteutettu koulutus saattaa lisätä koulutettujen negatiivista suhtautumista välineeseen. Eräs haastatelluista myös toi asian esiin haastattelun yhteydessä.

Tämän kohdan tarkoituksena oli tuoda esiin ne tekijät, jotka vaikuttivat hypoteesin H2 toteutumiseen. Näiden perusteella enemmän koulutusta saaneet eivät olleet välttämättä tyytyväisempiä välineen käyttöön johtuen henkilökohtaisista syistä ja koulutuksen laadusta.

6.4.3 Hypoteesin H3 toteutuminen

Hypoteesi H3 oli tarkoitettu mittaamaan sitä, pitikö esittelemäni uusittu versio (kuvio 2) Iivarin [Iivari, 1996] CASE-käytön ympyrästä paikkansa tutkitussa yrityksessä. Valitettavasti saaduista tuloksista ei kuitenkaan voida tehdä yleisiä johtopäätöksiä pienen otoksen ja tasaisesti jakautuneiden vastausten vuoksi. Ainoastaan neljällä kyselyyn vastanneista käyttäjistä oli mahdollisuus vaihtoehdoisen välineen käyttöön. Kaikissa tapauksissa tämä

väline oli ohjelmoinnin suorittaminen käsin. Näiden neljän kesken vastaukset jakautuivat tasan sen suhteen, olivatko he tyytyväisiä MetaEdit+:n käyttöön. Lisäksi on huomioitava, että kellään vastaajista ei tosiaan ollut toista vaihtoehtoista CASE-välinettä valittavanaan, mikä vaikeuttaa asian arvioimista CASE-välineiden näkökulmasta. Tähän hypoteesiin ei siis löydetty vastausta tämän tutkimuksen puitteissa.

6.4.4 Hypoteesin H4 toteutuminen

Hypoteesin H4 avulla haluttiin tutkia vaikuttiko välineen monimutkaisuuden kokeminen käyttäjätyytyväisyyteen, kuten se muissa tutkimuksissa oli vaikuttanut [Iivari, 1996]. Tutkimustulokset olivat kuitenkin kahtiajakautuneita. Osa vastaajista, joiden mielestä väline oli monimutkainen, käytti sitä kuitenkin mielellään ja oli sitä mieltä, että MetaEdit+:n avulla he suoriutuivat paremmin työstään. Toiset puolestaan olivat ehdottomasti sitä mieltä, että he tekisivät ohjelmakoodin mieluummin käsin ja kokivat välineen vaikeuttavan työtehtävien hoitoa. Hypoteesia H4 ei siis voida perustellusti sanoa pitävän paikkansa, mutta joissain tilanteissa välineen monimutkaiseksi kokevat käyttäjät näyttävät suhtautuvan negatiivisemmin koko välineeseen.

6.4.5 Hypoteesin H5 toteutuminen

Hypoteesi H5 liittyi välineen käytön oppimisprosessiin. Sen mukaan välineen käytön nopeasti oppineet ovat tyytyväisempiä. Tämä oletus ei näyttänyt kiistämättä pitävän paikkaansa ainakaan tutkitussa otoksessa, sillä asetettujen parametrien puitteissa nopeasti ja hitaasti oppineet käyttäjät olivat miltei yhtä lailla tyytyväisiä tai tyytymättömiä välineen käyttöön. Erityisesti hitaammin oppineiden vastaukset jakaantuivat tasapuolisesti negatiivisiin ja positiivisiin arvioihin. Nopeasti välineen käytön oppineet määriteltiin sellaisiksi, jotka olivat oppineet välineen käytön keskimääräistä nopeammin, alle neljässä kuukaudessa. Hitaammin välineen käytön oppineet puolestaan määriteltiin sellaisiksi käyttäjiksi, jotka oppivat välineen käytön keskimääräistä hitaammin, yli kuudessa kuukaudessa. Näiden jaotteluiden perusteella näytti siis siltä, että useat nopeammin oppineet ovat tyytyväisiä välineeseen. Hitaammin oppineista ei voida varmuudella sanoa mitään, koska vastaukset jakaantuivat tasaisesti kahden vaihtoehdon välille. On kuitenkin jälleen kerran huomioitava, että kyseessä on

rajoittunut otos, jonka perusteella ei välttämättä voida vetää mitään yleistettyjä johtopäätöksiä. Käyttäjien mielipiteet välineen käyttämisestä ja sen vaikutuksesta työsuoritukseen on esitetty taulukoissa 8 ja 9 ristiintaulukoituna oppimiseen kulutetun ajan suhteen.

	nopeimmin oppineet	keskitaso	hitaimmin oppineet
hyvin mieluisaa	0 0.00%	0 0.00%	0 0.00%
mieluisaa	5 62.50%	1 50.00%	1 50.00%
en osaa sanoa	3 37.50%	1 50.00%	0 0.00%
vastennielistä	0 0.00%	0 0.00%	1 50.00%
hyvin vastennielistä	0 0.00%	0 0.00%	0
Yhteensä	8	2	2

Taulukko 8. MetaEdit+ :n käyttämisen kokeminen. Eri vastausvaihtoehtojen frekvenssit ja suhteelliset frekvenssit ristiintaulukoituna oppimiseen kulutetun ajan suhteen.

	Nopeimmin Oppineet	keskitaso	hitaimmin oppineet
hyvin negatiivisesti	0 0.00%	0 0.00%	0 0.00%
negatiivisesti	1 12.50%	0 0.00%	1 50.00%
en osaa sanoa	5 62.50%	1 50.00%	0 0.00%
positiivisesti	2 25.00%	1 50.00%	1 50.00%
hyvin postiiivisesti	0 0.00%	0 0.00%	0 0.00%
Yhteensä	8	2	2

Taulukko 9. MetaEdit+ :n vaikutus työsuoritukseen. Eri vastausvaihtoehtojen frekvenssit ja suhteelliset frekvenssit ristiintaulukoituna oppimiseen kulutetun ajan suhteen.

7 Päätelmät ja yhteenveto

Tässä kohdassa tarkoitukseni on esitellä saadut tulokset ja analysoida niitä. Käsittelen ensin tärkeimmät kirjallisuuskatsauksessa esiin tulleet päätelmät ja sen jälkeen siirrytään analysoimaan kyselytutkimuksen avulla saatuja tuloksia.

Saatujen tulosten perusteella voidaan huomata, että CASE-välineiden menestyksellisessä käyttöönotossa ja käyttäjätuottavuuden aikaansaamisessa on huomioitava paljon muitakin seikkoja kuin pelkästään käytettävän välineen ominaisuudet. Tässä onkin yksi CASE-välineiden käytön suurimmista ongelmista, sillä kehitettävän välineen ominaisuudet eivät ole riittävä tae menestykselle, vaan on luotava myös ihanteelliset olosuhteet sen käyttöönotolle ja käyttämiselle. Vaikka saatujen tulosten perusteella nykyisistä välineistä puuttuukin useita käyttäjien kaipaamia ominaisuuksia, on koulutuksella ja välineiden käytön vapaaehtoisuudella mahdollista helpottaa käyttäjätuottavuuden syntymistä, vaikka välineen ominaisuudet eivät täysin vastaisikaan käyttäjien odotuksia. Pohtimisen arvoista kuitenkin on, kuinka käyttöönotossa voidaan painottaa vapaaehtoisuutta nykyisessä kiristyneessä kilpailutilanteessa. Tämä onkin varmasti lisätutkimuksen arvoinen osa-alue.

Tulosten mukaan on tunnistettavissa myös selvää kehityksen tarvetta CASE-välineiden pehmeämmissä ominaisuuksissa. Kuten Jarzabek ja Huang tutkimuksissaan toteavat, välineiden menetelmäkeskeisen suunnittelun sijaan niiden ominaisuuksia olisi rakennettava enemmän ihmisten käyttöön, sillä ihmisethän näillä välineillä työtä tekevät [Jarzabek ja Huang, 1998], [Vessey et al., 1992]. Kuitenkin toisaalta useissa yhteyksissä painotettiin, että CASE-välineiden tulisi tukeutua vahvasti tiettyyn metodologiaan, jotta niistä olisi käyttäjien kannalta hyötyä [Stobart et al., 1993; Griffiths, 1994; Nosek et al., 1992; Goldstein, 1990]. Jarzabekin ja Huangin mukaan myös ihmisten tunteet olisi paremmin otettava huomioon CASE-välineiden käyttöliittymän suunnittelussa. Tunteiden käytöstä käyttöliittymissä tarvitaan kuitenkin vielä lisätutkimusta, ennen kuin sen kaltaisia toimintoja voidaan laajamittaisesti käyttää hyväksi suunnittelussa.

CASE-välineiden synnyttämässä tyytyväisyydessä on myös suuria ihmisten yksilökohtaisia eroja. Nämä erot olisi käyttöönottovaiheessa pyrittävä tunnistamaan, jotta ne eivät pääse

häiritsemään itse välineen käyttöä. Erityisen tärkeää olisi identifioida ne työntekijät, jotka kokevat uudet välineet uhkana raskaalla työllä hankitulle ammattitaidolleen tai taitojensa markkinoitavuudelle. Heidän kohdallaan erityisesti vapaaehtoisuus näyttelee suurta osaa uuden työkalun hyväksymisessä ja sen tehokkaassa käyttämisessä. Työntekijän tarpeisiin ja lähtötasoon räätälöidyn koulutuksen on myös todettu auttavan työntekijän sopeutumista uuden välineen käyttöön.

CASE-työkalujen uudelleenkäyttöominaisuuksia pidetään myös tärkeinä käyttäjien keskuudessa. Näihin liittyy kuitenkin ohjelmistoteollisuuden yleinen hiljaisen tiedon ongelma, koska kaikkia ohjelmiston tuotantoprosessin tietoja ei saada vangittua välineeseen. Tutkimusta parempien uudelleenkäyttömenetelmien kehittämisestä tarvitaankin siis edelleen.

CASE-välineiden suunnittelijat joutuvat kohtaamaan monia ongelmia suunnitellessaan uusia välineitä ohjelmistoteollisuuden käyttöön. Perusongelmana tuntuu olevan se, että hyvän CASE-välineen lisäksi tarvitaan myös erittäin pätevää koulutusta sen käyttöön, jotta sitä voitaisiin tehokkaasti käyttää. Tämä onkin alue, joka kaipaa paljon lisätutkimusta. On selvitetävä, miten saadaan paketoitua CASE-väline koulutuksen kanssa ohjelmoijia houkuttelevaksi pakettiratkaisuksi.

Yleisenä huomiona järjestetystä empiirisestä tutkimuksesta voidaan sanoa, että MetaEdit+:n käyttäjät olivat melko tyytyväisiä välineen käyttöön. Suurimpana ongelmana koettiin kautta linjan ohjelmiston käyttöliittymään liittyvät ongelmat. Tämä on jonkin verran ristiriidassa muiden tulosten kanssa, koska niissä käyttöliittymään liittyviin tekijöihin oltiin tyytyväisiä ja niiden katsottiin olevan CASE-välineiden muiden ominaisuuksien tasolla. [Finnigan et al., 2000]. Lisäksi käyttäjät olivat jonkin verran tyytymättömiä koodigeneraattoriin ja sen tuottaman ohjelmakoodin selkeyteen.

Kyselytutkimuksen tulokset olivat suhteessa asetettuihin tutkimushypoteeseihin osittain odotettuja, mutta joiltain osin myös yllättäviä. Suurimpia yllätyksiä oli se, että ympäristön vaikutus ei tutkimuksessa yrityksessä ollut ehkä niin suuri kuin ennalta odotettiin. Asetettujen hypoteesien valossa näyttää siltä, että MetaEdit+ ja sen käyttäjien käyttäjätyytyväisyys noudattelee yleisesti eri tutkimuksissa havaittuja linjoja. Asetetuista hypoteeseista näyttivät toteutuvan ainakin hypoteesit H1 ja H4. MetaEdit+:n vanhemman version käyttäjät näyttivät olevan useilla osa-alueilla tyytymättömämpiä kuin uudemman 3.0 version käyttäjät. Tämähän

on tietysti toisaalta hyväkin asia, koska tällöin CASE-välineen kehitystyössä on menty oikeaan suuntaan. Hypoteesin H4 suhteen tilanne oli melko lailla samankaltainen ja kyselytutkimuksen tulokset osoittivat, että välineen monimutkaisuuden kokeminen vaikuttaa usein käyttäjätyytyväisyyteen negatiivisesti. Hypoteesi H2 ei kuitenkaan näyttänyt ainakaan tällä otoksella pitävän paikkaansa, sillä enemmän koulutusta välineen käytössä saaneet henkilöt eivät välttämättä olleet tyytyväisempiä. Tähän kuitenkin löydettiin mahdollinen selitys henkilökohtaisten CASE-välineiden käyttäjätyytyväisyyteen vaikuttavien seikkojen joukosta. Hypoteesi H3:n kohtalo oli tässä tutkimuksessa jäädä ratkaisemattomaksi, sillä otoksen pieni määrä ja sen piirissä syntynyt tasajako eivät anna todisteita puoleen eikä toiseen. Hypoteesi H5:n oikeellisuudesta ei myöskään saatu kiistatonta näyttöä. Tämän otoksen perusteella välineen oppimisen nopeudella ei näyttänyt olevan merkitystä välineen käyttäjätyytyväisyyden suhteen.

Tällä tutkimuksella on myös rajoituksensa. Suuri osa tutkimusaineistosta pohjautuu subjektiiviseen aineistoon käyttäjien kommenteista ja haastatteluista. Vaikka subjektiivista aineistoa on pidetty parhaana mittarina käyttäjätyytyväisyyden mittaamisessa [Moore ja Benbasat, 1991; Hartrum et al., 1989], on virhemarginaali aina olemassa. On myös muistettava, että kirjallisuuskatsaus käsittelee CASE-välineitä hyvin korkealla abstraktiotasolla, eivätkä yleiset tulokset ole aina suoraan erikoistettavissa tiettyyn yksittäiseen välineeseen.

Järjestetty kyselytutkimus ei myöskään välttämättä anna kuvaa mistään muusta kuin juuri kyseisessä yrityksessä vallitsevasta tilanteesta ja kyseisen yrityksen itse suunnittelemasta CASE-välineestä. Vaikka saadut tulokset suurelta osin tukevatkin muissa tutkimuksissa havaittuja seikkoja, on otettava kuitenkin huomioon kyselyn yksinkertainen rakenne sekä suhteellisen pieni ja valikoitunut otos. Myöskään määriteltyjä validointimenettelyjä ei suoritettu täydellisesti kysymysten puutteellisuuden vuoksi.

On myös syytä huomioida, että CASE-välineet kehittyvät muiden tietotekniikan osa-alueiden tapaan jatkuvasti. Osa käyttämästäni tutkimusmateriaalista saattaa sisältää jo vanhentunutta tietoa ohjelmistotuottajien tuodessa uusia tuotteita markkinoille. Kuitenkin ihmisten ominaisuudet ovat varmasti pysyneet samoina, joten pääpiirteissään tutkimukseni sisältöä voidaan käyttää hyväksi pitkälläkin aikavälillä.

Lähteet

[Aaen, 1993]

Aaen, Ivan, CASE user satisfaction-Impact evaluations in user organizations, Computer-Aided Software Engineering, 1993. CASE '93., Proceeding of the Sixth International Workshop on , 1993 Page(s): 66 - 74.

[Bailer et al., 1993]

B. Bailer, K. Bauknecht, C Schatzmann, Ways to CASE: An analysis of selected CASE experiences, Computer-Aided Software Engineering, 1993. CASE '93., Proceeding of the Sixth International Workshop on , 1993, Page(s): 36 - 43.

[Bailey and Pearson, 1983]

J. E. Bailey and S. W. Pearson, Development of a tool for measuring and analyzing computer user satisfaction, Management Science, 29(5):530-544, May 1983.

[Brown, 1994]

A.W. Brown, Why evaluating CASE environments is different from evaluating CASE tools, Assessment of Quality Software Development Tools, 1994, Proceedings, Third Symposium on , 1994, Page(s): 4 - 13.

[Budgen et al., 1993]

David Budgen, Mustafa Marashi, Andrew Reeves, CASE Tools: Master or Servants, Software Engineering Environments Conference, 1993, Proceedings, Sept. 1993, Page(s): 156 - 165.

[Coallier, 1990]

F. Coallier, A strategy for CASE tool standard, Computers and Communications, 1990, Conference Proceedings, Ninth Annual International Phoenix Conference on, 1990, Page(s): 380 - 384.

[Crozier et al., 1989]

M. Crozier, D Glass, J.G. Hughes, W. Johnston, I. McChesney, Information and Software Technology, Critical analysis of tools for computer-aided software engineering, Vol 31, No 9, 1989, Page(s): 486 - 496.

[Dabin, 1991]

Matthieu Dabin, Software Reuse and Case Tools, Computer Software and Applications Conference, 1991, COMPSAC '91, Proceedings of the Fifteenth Annual International, 1991, Page(s): 2 - 3.

[Finnigan et al., 2000]

D. Finnigan, E.A. Kemp, D. Mehandjiska, Towards an ideal CASE tool, Software Methods and Tools, 2000, SMT 2000, Proceedings. International Conference on, 2000, Page(s): 189 - 197.

[Goldstein, 1990]

R. Goldstein, Methodologies and CASE tools-the missing link, CompuEuro '90, Proceedings of the 1990 IEEE International Conference on Computer Systems and Software Engineering, 1990, Page(s):138 - 145.

[Griffiths, 1994]

G. Griffiths, CASE in the third generation, Software Engineering Journal, Volume: 9 Issue: 4 , July 1994, Page(s):159 - 166.

[Hartrum et al., 1989]

T.C. Hartrum, T.C. Mallery, J.W. Foley, Evaluating user satisfaction of an interactive computer program, Aerospace and Electronics Conference, 1989, NAECON 1989, Proceedings of the IEEE 1989 National , 1989, Vol.2., Page(s): 508 - 514.

[Ho, 1992]

Yaw-Chin Ho, To what extent will case tools assist users in the systems development?: a case study in academic environment, Proceedings of the 1992 ACM SIGCPR conference on Computer personnel research, 1992, Page(s) 93 - 96.

[IBM, 2000]

IBM BlueEyes projekti, <http://www.almaden.ibm.com/cs/blueeyes/>. Luettu 08.04.2000

[IEEE, 1990]

Institute of Electrical and Electronics Engineers. *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*. New York, NY, 1990.

[Iivari, 1996]

J. Iivari, Why are CASE tools not used ?, *Communications of the ACM*, Vol. 39, No. 10, 1996, Page(s): 94 - 103.

[Iivari ja Maansaari, 1997]

J. Iivari ja J. Maansaari, The impact of CASE on IS professionals' work and the motivation to use CASE, International Conference on Information Systems, Atlanta, GA, 1997.

[ISO, 2001]

International Organization for Standardization, <http://www.iso.ch/>, luettu 29.4.2001.

[Jarzabek ja Huang, 1998]

Stan Jarzabek, Riri Huang, The Case for User-Centered CASE Tools, *Communications of the ACM*, August 1998/Vol. 41, No. 8.

[Järvinen ja Järvinen, 1996]

Pertti Järvinen ja Annikki Järvinen, Tutkimustyön metodeista, 1996, Opinpaja Oy, ISBN 951-97113-1-7.

[Jones, 1992]

Capers Jones, CASE's missing elements, *IEEE Spectrum* June 1992, pp. 38 - 41.

[Kemerer, 1992]

Chris F. Kemerer, How The Learning Curve Affects CASE Tool Adoption, IEEE Software, Volume 9 3, May 1992, Page(s) 23 - 28.

[Kusters ja Wijers, 1993]

R.J. Kusters, G.M. Wijers, On the practical use of CASE-tools: Results of a survey, Computer-Aided Software Engineering, 1993, CASE '93., Proceeding of the Sixth International Workshop on, 1993, Page(s): 2 - 10.

[Lee, 1993]

Hing Yan Lee, Software Engineering, Knowledge for Software Reuse, Computer-Aided Software Engineering, 1993, CASE '93, Proceeding of the Sixth International Workshop on, Page(s) 263 - 269.

[Lending ja Chervany, 1998]

Diane Lending and Norman L. Chervany, The use of CASE tools, Proceedings of the 1998 conference on Computer personnel research, 1998, Page(s) 49 - 58.

[Low ja Leenanurakra, 1999]

Graham Low ja Vichai Leen anuraksa, Software Quality and CASE Tools, Software Technology and Engineering Practice, 1999, STEP '99 Proceedings, Page(s) 142 - 150.

[MetaPHOR Group, 2001]

MetaPHOR Group, <http://metaphor.it.jyu.fi>, luettu 29.04.2001.

[Microsoft, 2001]

MSDN Online User Interface Center, <http://msdn.microsoft.com/ui/default.asp>, luettu 29.04.2001.

[Mili et al., 1995]

H. Mili, F. Mili ja A. Mili (1995), Reusing software: Issues and research directions, IEEE Transactions on Software Engineering 21, No. 6, Page(s) 528 - 562.

[Moore ja Benbasat, 1991]

G.C. Moore, I. Benbasat, Development of an instrument to measure the perceptions of adopting an information technology innovation, *Information Systems Research* 2, 3 (1991), Page(s) 192 - 222.

[Nielsen, 1993]

Nielsen, Estimating the relative usability of two interfaces: heuristic, formal, and empirical methods compared, *Proceedings of the conference on Human factors in computing systems*, 1993, Page(s) 214 - 221.

[Norman, 1998]

Ronald J. Norman, CASE Productivity Perceptions of Software Engineering Professionals, *Communications of the ACM*, September 1998/Vol. 32, No. 9.

[Nosek et al., 1992]

John Nosek, Gary Baram, Geoff Steinberg, Ease of learning and using a case software tool: an empirical evaluation, Proceedings of the 1992 ACM SIGCPR conference on Computer personnel research, 1992, Page(s) 75 - 80.

[Orlikowski, 1992]

Wanda J. Orlikowski, LEARNING FROM NOTES: Organisational Issues in Groupware Implementation, CSCW Proceedings, November 1992, Page(s) 362 - 369.

[Orlikowski, 1988]

Wanda J. Orlikowski, CASE Tools and the IS workplace: Some findings from empirical research, Proceedings of the ACM SIGCPR conference on Management of information systems personnel, 1988, Page(s) 88 - 97.

[Peterson, 1996]

M.G.E. Peterson, User satisfaction surveys, what the engineer should know, Computer-Based Medical Systems, 1996, Proceedings Ninth IEEE Symposium on, 1996, Page(s): 71 - 76.

[Prather, 1993]

B. Prather, Critical failure points of CASE tool evaluation and selection, Computer-Aided Software Engineering, 1993, CASE '93, Proceeding of the Sixth International Workshop on, 1993, Page(s): 60 - 63.

[Sorensen, 1995]

C. Sorensen, Why CASE tools do not support co-ordination, CSCW (Computer Supported Co-operative Working) and the Software Process (Digest No. 1995/036), IEE Colloquium on , 1995, Page(s): 4/1 - 4/3.

[Spurr ja Layzell, 1992]

Kathy Spurr ja Paul Layzell, Case, Current Practise, Future Prospects. Wiley, 1992, ISBN 0 471 93304 X.

[Stiemerling et al., 1997]

Oliver Stiemerling, Helge, Volker Wulf, How to make software softer--designing tailorable applications, Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques, 1997, Page(s) 365 - 376.

[Stobart et al., 1993]

S.C. Stobart, A.J. van Reeken, G.C. Low, J.J.M. Trienekens, J.O. Jenkins, J.B. Thompson, D.R. Jeffery, An Empirical Evaluation of the Use of CASE Tools, Computer-Aided Software Engineering, 1993, CASE '93, Proceeding of the Sixth International Workshop on, Page(s) 81 - 87.

[Sumner, 1995]

Mary Sumner, Factors influencing the success of computer-assisted software engineering, Proceedings of the 1995 ACM SIGCPR conference on Supporting teams, groups, and learning inside and outside the IS function reinventing IS, 1995, Page(s) 148 - 155.

[Vessey et al., 1992]

Iris Vessey, Sirkka L. Jarvenpaa, Noam Tractinsky, Evaluation of vendor products CASE tools as methodology companions, *Commun. ACM* 35, 4 (Apr. 1992), Page(s) 90 - 105.

Vastatessanne kysymyksiin järjestelmän toimivuudesta, olkaa hyvä ja tarkastelkaa, miten järjestelmä on toiminut normaalioloissa viimeisen puolen vuoden aikana, ja jättäkää huomiotta kohtaamanne yksittäiset ongelmalliset tapahtumat.

1. Sukupuoli

Mies

Nainen

2. Ikä

3. Mitä versiota käytät MetaEdit+ -ohjelmistosta ?

4. Kuinka kauan olet käyttänyt MetaEdit+ -ohjelmistoa ?

5. Kuinka paljon käytät tätä ohjelmistoa ?

Päivittäin

Ainakin kerran viikossa

Ainakin kerran kuukaudessa

En lainkaan

6. Kuinka kauan olet ollut yrityksen palveluksessa ?

—

7. Oletko muita käyttänyt MetaEdit+:n kaltaisia CASE-välineitä ?

En

Kyllä, mitä ?

8. Kuinka olet oppinut MetaEdit+ -ohjelmiston käytön ?

Itseopiskelulla

Koulutuksessa

Työtovereiden opastuksella

Muulla tavalla, miten ?

9. Jos olet saanut koulutusta välineen käytössä, kuinka paljon olet sitä saanut ?

 tuntia

10. Olen saanut koulutusta välineen käytössä

Liikaa

Sopivasti

en osaa sanoa

Liian vähän

en ollenkaan

11. Kuinka paljon sinulta kului aikaa välineen käytön oppimiseen ?

12. Välineen dokumentaatio on mielestäni

Hyvä

Riittävä

Välttävä

Huono

13. Onko sinulla käytettävissäsi muita vaihtoehtoisia välineitä työsi suorittamiseen ?

14. Käyttäisitkö mieluummin edellä mainittuja vaihtoehtoisia välineitä ?

Kyllä

En

ei vaihtoehtoisia välineitä

15. Koen, että pystyn MetaEdit+ avulla suoriutumaan työstäni paremmin kuin perinteisin menetelmin (esim käsin koodaamalla)

Kyllä

Ei

16. MetaEdit+ :n käyttö on mielestäni

Hyvin mieluisaa

Mieluisaa

en osaa sanoa

Vastenmielistä

Hyvin vastenmielistä

17. Välineen ominaisuudet vaikuttavat työhalukkuuteeni

Hyvin negatiivisesti

Negatiivisesti

en osaa sanoa

Positiivisesti

Hyvin positiivisesti

18. Välineen käyttöliittymä on mielestäni

Selkeä

Melko selkeä

en osaa sanoa

Melko vaikeaselkoinen

Vaikeaselkoinen

19. Välineen toimintaperiaatteet ovat mielestäni

Vaikeita ymmärtää

Melko vaikeita ymmärtää

en osaa sanoa

Melko helppoja ymmärtää

Helppoja ymmärtää

20. Väline tukee mielestäni käytössä olevia prosesseja ja käytäntöjä

Hyvin

Melko hyvin

en osaa sanoa

Melko huonosti

Huonosti

21. Väline tukee mielestäni ohjelmakoodin uudelleenkäyttöä

Hyvin

Melko hyvin

en osaa sanoa

Melko huonosti

Huonosti

22. Yhteensopivuus MetaEdit+:n ja muiden käyttämäni työkalujen kanssa on mielestäni

Huono

Melko huono

en osaa sanoa

Melko hyvä

Hyvä

23. Mainitse kolme välineen parasta ominaisuutta

24. Mainitse kolme välineen ärsyttävintä ominaisuutta

25. Vapaat kommentit välineestä ja sen käytöstä tai tästä kyselystä