

Työmäärien arviointi tietojärjestelmäprojekteissa; käytännön kokemuksia ja havaintoja nykytilasta

Outi Alarakkola

Opinnäytetyö

Tietojenkäsittelyn koulutusohjelma

2012



Tietojenkäsittelyn koulutusohjelma

<p>Tekijä Outi Alarakkola</p>	<p>Aloitusvuosi 2008</p>
<p>Opinnäytetyön nimi Työmäärien arviointi tietojärjestelmäprojekteissa; käytännön kokemuksia ja havaintoja nykytilasta.</p>	<p>Sivu- ja liitesivumäärä 44+4</p>
<p>Ohjaaja Juhani Välimäki.</p>	
<p>Työmäärien arviointi sivuaa jollain tavalla lähes jokaisen tietojärjestelmätyön ammattilaisen työtä. Tehtävistä on laadittava työmääräarviot ennen toteuttamista, ja valmiin projektin työmäärien toteutumista seurataan. Tietojärjestelmäprojektien työmäärät ylittyvät jostain syystä edelleen usein, vaikka erilaisia apuvälineitä työmäärien arviointiin on yritetty kehittää useiden vuosikymmenten ajan.</p> <p>Tutkimuksen tavoitteena oli selvittää, mitä menetelmiä ja työtapoja käyttäen tietojärjestelmäprojektien työmääräarviot tehdään tämän päivän Suomessa ja miten työmäärien arviointia voitaisiin kehittää. Tutkimus oli rajattu koskemaan suuria ja keskisuuria tietojärjestelmien toimittajayrityksiä.</p> <p>Tutkimuksen teoriaosuudessa kuvataan olemassa olevia työmäärien arviointimenetelmiä ja aiheeseen liittyviä aikaisempia tutkimuksia. Tutkimus toteutettiin haastattelemalla viittä asiantuntijaa, jotka joutuvat työssään tekemään työmääräarvioita. Haastattelun tulokset analysoitiin suhteessa teoriapohjaan ja aikaisempiin tutkimuksiin. Tutkimuksessa käytettiin kvalitatiivista tutkimusotetta, ja haastattelut olivat puolistrukturoituja teemahaastatteluja.</p> <p>Tutkimuksessa kävi ilmi, että työmäärät laaditaan enimmäkseen asiantuntijan omaan kokemukseen perustuen. Käytettävät apuvälineet ovat lähinnä yksinkertaisia taulukoita tai muistilistoja, joiden avulla arvioitava tehtävä jaetaan pienempiin osiin erilaisin kriteerein. Asiantuntijat itse toivoivat organisaatiolta enemmän tukea ja ohjeita työmääräarvioiden tekoon, sekä jotain helposti käytettävää apuvälinettä. Nyt työmääräarvioiden laatu on lähinnä yksittäisten asiantuntijoiden ammattitaidon ja huolellisuuden varassa.</p>	
<p>Asiasanat Työmäärien arviointi, työmääräarvio, työmäärä, tietojärjestelmät, projekti.</p>	

Degree Programme in Information Technology

<p>Author Outi Alarakkola</p>	<p>Year of entry 2008</p>
<p>The title of thesis Estimating Work Effort in Software Development Project; user experiences and observations of the current situation</p>	<p>Number of pages and appendices 44+4</p>
<p>Supervisor Juhani Välimäki</p>	
<p>Almost every professional in IT branch must participate in estimating work efforts. Work efforts for every assignment must be estimated beforehand. Project managers keep score how the estimations came true. For some reason software development projects seem to overrun their estimations even though there have been constant efforts to develop methods for estimating work efforts for decades.</p> <p>The goal for this thesis was to find out what are the methods used in estimating work efforts in software development projects in Finland today, and to discover means to improve estimating process. The research concentrated only on large and medium-sized IT companies.</p> <p>The thesis includes a theory section and an empirical section. The theory section consists of the base knowledge about methods used in estimating work efforts and of a summary of previous studies about the subject. The empirical part focused on interviews with a group of IT experts. The study was based on qualitative method and interviews where carried out as half structured theme interviews.</p> <p>The interviews showed that the estimation methods most frequent used are expert judgement-based. Excel-tables or checklists are used as tools when sorting software objects by their magnitude or complexity. The interviewed experts would like to have more support and guidelines from their organizations in estimating process and possibly some user-friendly tools. Now the quality of the estimates depends mainly on professional ability and competence of each individual.</p>	
<p>Key words Effort estimation, work effort, software development, project</p>	

Sisällys

1	Johdanto	1
1.1	Tutkimusongelma ja tutkimuskysymykset	2
1.2	Tutkimusmenetelmät ja perustelut.....	2
2	Työmäärien arvioinnissa käytettävät menetelmät	4
2.1	Asiantuntijoiden tekemät arviot	4
2.1.1	Analogiaan perustuvat arviot	5
2.1.2	Työn jako vaiheisiin, Bottom-up ja Top-Down.....	5
2.1.3	Työmäärien arviointi ketterillä menetelmillä toteutettavissa projekteissa	6
2.2	Malleihin perustuvat arviot	8
2.2.1	Algoritmin perusmalli ja parametrit	9
2.2.2	COCOMO-malli.....	10
2.2.3	FPA eli toimintopisteanalyysi.....	12
2.2.4	Käyttötapauksiin perustuva menetelmä.....	14
2.3	Muut menetelmät	18
2.3.1	Parkinsonin laki	18
2.3.2	Pricing to win eli taloudelliseen voittoon tähtäävä arviointi	18
2.4	Aiemmat tutkimukset.....	19
3	Haastattelututkimus tietojärjestelmäprojektien työmäärien arvioinnin nykytilasta suurissa ja keskisuurissa tietojärjestelmiä toteuttavissa suomalaisissa yrityksissä	22
3.1	Kohderyhmä	22
3.2	Haastattelut	23
3.2.1	Haastateltavien urallaan arvioimien tehtävien sisältö.....	23
3.2.2	Työmääräarvioinnin menetelmien tuntemus	25
3.2.3	Miten työmääräarviot tehdään	26
3.2.4	Pienten ja suurten kokonaisuuksien arviointi	28
3.2.5	Työmääräarvioinnin kompastuskivet.....	29
3.2.6	Työmäärien arviointiin varattu aika.....	30
3.2.7	Tiedon määrä ja laatu	30
3.2.8	Työmäärien jakaantuminen ohjelmistokehitysprojektissa	31
3.2.9	Työmääräarvioiden toteutuminen	31

3.2.10	Miten työmäärien arviointia voisi oppia?.....	32
3.2.11	Prosessiajattelu työmäärien arvioinnissa.....	33
3.2.12	Työmääräarvioiden parantaminen.....	33
3.2.13	Esimerkkejä haastateltavien aiemmin tekemistä arviointitehtävistä	34
4	Tutkimuksen tulokset	36
4.1	Minkälaisia menetelmiä ja työskentelytapoja työmäärien arvioinnissa käytetään?36	
4.2	Minkälaisia ongelmia työmäärien arvioinnissa kohdataan?.....	37
4.3	Ehdotuksia työmääräarvioinnin helpottamiseksi ja arvioiden laadun parantamiseksi	39
5	Tulosten ja oppimisprosessin arviointi.....	42
5.1	Tulosten pätevyys ja luotettavuus	42
5.2	Oppimisen ja opinnäytetyöprosessin arviointi	43
6	Yhteenveto	44
	Lähteet.....	45
	Liitteet.....	48

1 Johdanto

Projektipäällikön on osattava ennustaa mahdollisimman tarkasti etukäteen tietojärjestelmäprojektin työmäärät ja kustannukset. Ennustaminen on välttämätöntä sekä tietohallinnon päätösten tekoa varten että projektin suunnittelua ja hallintaa varten. Työmäärien arviointi ei ole pelkästään projektipäällikön tehtävä, vaan yhä useammin jokainen tietojärjestelmäprojektissa työskentelevä joutuu arvioimaan oman työnsä osuuden etukäteen. Niinpä aihe koskettaa kaikkia alan ammattilaisia jollakin tavalla. Tietojärjestelmäprojektien työmäärien arviointia pidetään hankalana, vaikka erilaisia apuvälineitä arvioiden tekoon on ollut käytettävissä useiden vuosikymmenten ajan.

Tämän tutkimuksen tavoitteena on selvittää, millä menetelmillä tietojärjestelmäprojektien työmääriä arvioidaan tänä päivänä Suomessa, ja miten käytössä olevia työtapoja voitaisiin parantaa. Aihe on tärkeä, koska tietojärjestelmien kehityksessä pyritään jatkuvasti yhä suurempaan kustannustehokkuuteen. Tutkimusten mukaan jopa 60-80% ohjelmistoprojekteista ylittää työmäärä- ja kustannusarvionsa, eli alalla on ollut miltei sääntönä, että projektit ylittävät budjettinsa (Chandrasekaran & Venkatesh Kumar 2012; Molokken & Jorgensen 2003.). Arvioiden tarkkuutta parantamalla voitaisiin välttää sovittujen aikataulujen venyminen ja kustannusarvioiden ylittyminen ja ohjata yrityksen resurssit mahdollisimman kannattavien tietojärjestelmäprojektien kehittämiseen.

Tutkimuksen aluksi luvussa 2 käydään läpi yleisimmät tunnetut menetelmät, joita voidaan käyttää työmääräarvioiden laatimisessa. Luvussa kerrotaan myös, mitä menetelmiä aikaisemmissa tutkimuksissa on todettu länsimaisissa yrityksissä käytettävän, ja mitkä ovat eri menetelmien edut ja haitat. Luvussa 3 kerrotaan haastattelututkimuksesta, jossa selvitettiin työmääräarvioinnin nykytilaa Suomessa. Tutkimusta varten on haastateltu järjestelmäasiantuntijoita, jotka työskentelevät suurissa tai keskisuurissa suomalaisissa yrityksissä, ja jotka osallistuvat työssään tietojärjestelmäprojektien työmäärien arviointiin.

Lopuksi luvussa 4 esitellään tutkimuksen tulokset ja luvussa 5 arvioidaan tutkimustuloksien pätevyyttä ja luotettavuutta sekä oppimista ja opinnäyteprosessia.

1.1 Tutkimusongelma ja tutkimuskysymykset

Tämän tutkimuksen tavoitteena oli selvittää, mikä on tietojärjestelmäprojektien työmäärien arvioinnin nykytila suurissa ja keskisuurissa tietojärjestelmiä toteuttavissa suomalaisissa yrityksissä. Tietojärjestelmäprojekteilla tarkoitetaan tässä yhteydessä ohjelmistojen kehitys- tai ylläpitoprojekteja, jotka sisältävät määrittelyä, suunnittelua, toteutusta ja testausta. Valmisohjelmistoprojektit rajattiin pois tutkimuksesta.

Tutkimuksessa pyritään vastaamaan kysymyksiin:

- Minkälaisia menetelmiä ja työskentelytapoja työmäärien arvioinnissa käytetään?
- Minkälaisia ongelmia työmäärien arvioinnissa kohdataan?
- Miten työmääräarvioiden tekemistä voitaisiin helpottaa ja arvioiden laatua parantaa?

1.2 Tutkimusmenetelmät ja perustelut

Tutkimuksessa käytettiin kvalitatiivista eli laadullista lähestymistapaa. Tutkimusmateriaalin tiedonkeruumenetelminä käytettiin kirjallisiin aineistoihin perehtymistä sekä puolistrukturoituja teemahaastatteluja. Analyysi tehtiin abduktiivisesti ja ymmärtämiseen pyrkivällä lähestymistavalla. Tutkimustulosten tulkinnassa ja synteisien laadinnassa käytettiin omaa pohdintaa.

Laadulliseen tutkimukseen päädyttiin tässä tutkimuksessa siksi, että tutkimuskohteena oli inhimillinen toiminta, jota pyrittiin kuvaamaan ja tulkitsemaan.

Tutkimuksen teoriapohjan ja aikaisempien tutkimusten kuvaamisen tutkimusmateriaalina käytettiin kirjallisia lähteitä. Haastattelututkimus valittiin nykytilan ja kokemusten tiedonkeruumenetelmäksi, koska aihe on sen tyyppinen, että siitä on vaikea antaa yksiselitteisiä vastauksia kirjallisiin kysymyksiin. Haastattelutyypiksi valittiin puolistrukturoitu teemahaastattelu, joka tarkoittaa Kanasen (2008, 73) mukaan sitä, että haastattelussa on avoimia kysymyksiä, joihin ei ole valmiita vastausvaihtoehtoja. Haastatteluja varten oli laadittu keskustelua ohjaavia kysymyksiä aiheista, jotka nousivat esiin kirjallisia lähteitä, mm. aikaisempia tutkimustuloksia läpikäydessä. Kysymykset löytyvät tämän raportin liitteestä 2. Haastateltavat asiantuntijat saivat kertoa myös vapaasti kokemuksiinsa ja ehdotuksiinsa, jotka liittyivät työmäärien arviointiin. Tarkoitus oli luoda vapautu-

nut keskustelun omainen tilanne, joka rohkaisisi haastateltavia tuomaan esiin omia näkemyksiään omalla tavallaan. Aiheesta tiedettiin ennalta, että se tuottaa monitahoisia vastauksia ja herättää kriittistä keskustelua, joten haluttiin kuulla mahdollisimman paljon asiantuntijoiden vapaita kommentteja. Oli myös ennakoitavissa, että haastateltavat voivat kokemuksestaan riippuen kertoa aiheesta laajemmin kuin mitä tutkija on pystynyt etukäteen arvioimaan. Haastattelutilanteessa oli mahdollista esittää lisäkysymyksiä ja pyytää tarkennuksia ja perusteluita vastauksiin.

Haastatteluista tehtiin muistiinpanoja, mutta niitä ei kirjoitettu puhtaaksi sanasanaisesti (litteroitu). Litterointi ei olisi ollut järkevää, koska aineistoa olisi kertynyt liikaa. Haastatteluja tehtiin ja analysoitiin osittain samanaikaisesti, eli Hirsjärvi, Remes & Sajavaaran (2008, 219) käyttämällä termillä kuvattuna tutkimus eteni spiraalimaisesti. Ensimmäisten kolmen haastattelun vastauksista pystyttiin jo löytämään paljon yhteneväisyyksiä ja tuloksia voitiin alkaa kuvata ja analysoida. Loput kaksi haastattelua pystyttiin suuntaamaan aiempien haastattelujen myötä tärkeimmiksi nousseisiin ja lisäselvitystä vaativiin kohtiin. Aineistosta tehtiin abduktiivinen analyysi. ”Analyysi lähtee aineistosta, mutta teoriaa käytetään apuna analyysissä. Teoriaa ei varsinaisesti testata.” (Kananen 2008, 92.) Analyysin lähestymistapa oli ymmärtämiseen pyrkivä, joka tarkoittaa Hirsjärven ym. (2008, 219) mukaan laadullista analyysia ja päätelmien tekoa.

Tutkimustulosten tulkinnassa, synteisien ja johtopäätösten tekemisessä käytettiin tutkijan omaa pohdintaa.

2 Työmäärien arvioinnissa käytettävät menetelmät

Työmäärien arviointimenetelmät voidaan jakaa esimerkiksi asiantuntija-arvioihin ja erilaisiin malleihin. Asiantuntija-arviot perustuvat tekijän kokemukseen vastaavantyyppisistä töistä tai järjestelmän tuntemukseen. Mallit ovat algoritmeja tai menetelmiä, joiden kehittämisessä on käytetty toteutuneista tietojärjestelmäprojekteista kerättyjä historiatietoja. Tässä luvussa kerrotaan tarkemmin erilaisista työmäärien arvioinnissa käytetyistä menetelmistä jaoteltuna asiantuntijoiden tekemiin, malleihin perustuviin ja muihin tapoihin. Asiantuntija-arvioinnissa käytetyistä menetelmistä kuvataan analogiaan perustuvat arviot, työn jako vaiheisiin sekä työmäärien arviointi ketterien menetelmien projekteissa. Malleihin perustuvista arvioista kuvataan algoritmien perusrakenteita, COCOMO-mallia, toimintopisteanalyysia ja käyttötapauksiin perustuvaa menetelmää. (Molokken & Jorgensen 2003.)

2.1 Asiantuntijoiden tekemät arviot

Tietojärjestelmäprojektin työmäärien arviointi annetaan yleensä tehtäväksi ammattilaiselle, joka tulee toteuttamaan tehtävän, tai jolla on kokemusta aiemmista vastaavista tehtävistä. Eri henkilöt voivat arvioida projektin eri osa-alueiden työmääriä, esimerkiksi määrittelijät arvioivat määrittelyvaiheen työmäärän, ohjelmoijat ohjelmakoodin toteutuksen työmäärät, testaajat järjestelmätestauksen työmäärät ja projektipäällikkö hallinnollisten tehtävien työmäärät. (Pelin 2011, 116; Sommerville 2004, 621.)

Asiantuntijoiden käyttämät menetelmät vaihtelevat henkilön kokemuksen ja yrityksen käytäntöjen mukaan. Arviot voivat yksinkertaisimmillaan perustua pelkkään intuitioon tai valistuneeseen arvaukseen. Yleisimmin taustalla on kuitenkin kokemuksen mukaan tuoma näppituntuma, jolla tuotettua arviota voi olla hankala perustella muuten kuin fraasilla ”minusta tuntuu, että...”. Asiantuntijat puhuvatkin mielellään ”hatusta vetämisestä”, ”hihasta ravistelusta”, ”Mutu- tai Stetson-menetelmästä”. Cohnin (2009; 54-55) mukaan asiantuntija-arviot ovat hyödyllisimpiä perinteisissä projekteissa. Asiantuntija ei yleensä tarvitse kauaa aikaa arvion tekemiseen, ja tällainen arvio pitää usein varsin hyvin paikkansa. (Jorgensen 2004; Jorgensen 2007; Pelin 2011.)

Kaikista asiantuntijoiden mahdollisesti käyttämistä menetelmistä ei ole tietoa, koska asiantuntijoiden käyttämät menetelmät on aiemmissa tutkimuksissa ryhmitelty vain yhden otsikon alle (Jorgensen 2004). Edellä kuvattujen lisäksi tiedetään asiantuntijoiden käyttävän apunaan erilaisia analogiaan perustuvia menetelmiä, mikäli heillä on kerättyä riittävästi luotettavaa ja analysoitavaa tietoa aiemmista projekteista. Isoissa projekteissa arviointia helpottaa työn jako pienempiin osiin tai vaiheisiin, ja niiden arviointi ja arviointujen työmäärien yhteenlasku. Ketterien menetelmien projekteja varten on kehitetty oma tapa arvioida työmääriä. Näitä tapoja kuvataan seuraavissa kappaleissa tarkemmin.

2.1.1 Analogiaan perustuvat arviot

Analoginen arvio tehdään vertaamalla tehtävää muihin aiemmin tehtyihin tai arvioituihin tehtäviin. Ihmisten on helpompi verrata tehtävien kokoa suhteessa toisiin tehtäviin kuin tehdä arviointi aivan puhtaalta pöydältä. Analogiaa voidaan käyttää, jos samantyyppisiä ja samaan tekniseen ympäristöön sijoittuvia järjestelmäprojekteja on tehty aiemminkin. Kullakin asiantuntijalla on omia muistikuvia vastaavien töiden aiemmasta kestosta, ja he voivat tukeutua niihin arvioissaan. He eivät ehkä kuitenkaan osaa perustella miten päätyivät arvioonsa. Järjestelmällisemmin analogiaa voidaan käyttää apuna tutkimalla projektinhallintavälineisiin talletettuja toteutumatietoja aiemmista projekteista ja vertaamalla aiempien tehtävien sisältöjä arvioitavana oleviin tehtäviin. Vertailua voidaan tehdä järjestelmän komponenttitasolla tai koko sovelluksen tasolla. Talletettujen toteumien avulla voidaan pitemmällä aikavälillä kehittää yrityksen tai tiimin käyttöön omia taulukoita ja kertoimia, joita sovelletaan samantyyppisten töiden arvioinnissa. Analogioihin perustuvat arviot ovat hyviä, jos ympäristö ja välineet pysyvät samantaisina, ja eri töistä osataan tunnistaa yhteiset piirteet. Myöhemmin tässä tutkimuksessa kuvatut arviointialgoritmit perustuvat projekteista kerättyihin historiatietoihin ja analogiaan. (Cohn 2009, 55; Meli & Santillo 1999; Pelin 2011, 117.)

2.1.2 Työn jako vaiheisiin, Bottom-up ja Top-Down

Isoja kokonaisuuksia on vaikea hahmottaa ja arvioida. Projektinhallinnan kannalta työ kannattaa jakaa pienempiin tehtäviin, jotka arvioidaan erikseen. Jako voidaan tehdä esimerkiksi toiminnallisin perustein tai toteutusvaiheittain. Osituksesta on hyötyä etenkin, jos arvioitava järjestelmä on vaativa, ja arviointi muilla tavooin toisi todennäköisesti

epävarmoja tuloksia, koska arvioijilla ei ole kokemusta samanlaisesta kokonaisuudesta. Tehtävien pilkkominen pienempiin osiin mahdollistaa analogian hyödyntämisen ja helpottaa asiantuntija-arvioiden tekoa. Vaihejako vähentää turhaa optimismia, mutta toisaalta liian tarkalle tasolle osittaminen saattaa kuitenkin kasvattaa kokonaistyömääräarvioita. Pienistä tehtävistä saadaan liikaa informaatiota, joka ei ole relevanttia arvioinnin kannalta, mutta kasvattaa arvion suuruutta. (Cohn 2009, 56; Jorgensen 2004)

Bottom-up lähestyminen tarkoittaa, että arviointi aloitetaan pienimmistä osista - komponenteista, ja arvioidaan kunkin yksittäisen komponentin toteuttamisen vaatima työmäärä. Kaikkien järjestelmän komponenttien työmäärät lasketaan lopuksi yhteen. Jotta arviointi on mahdollista, on järjestelmää täytynyt suunnitella jo sen verran, että komponentit ovat tunnistettavissa. Top-down arviointi puolestaan alkaa järjestelmän ylimältä tasolta arvioimalla järjestelmän toiminnallisuus ja kuinka se tuotetaan alemman tason komponenteilla. Järjestelmätason toiminnot, kuten integrointi, kokoonpanon hallinta ja dokumentointi otetaan huomioon. Eri komponenttien osuus voidaan arvioida esimerkiksi prosentteina kokonaisuudesta. (Meli & Santillo 1999; Sommerville 2004, 621-622.)

Jorgensen (2004) suosittaa käyttämään sekä bottom-up että top-down menetelmiä itsenäisinä ja vertaamaan molemmilla tavoilla saatuja arvioita. Bottom-up tuo hänen mukaansa lisähyötyä vain projekteissa, jotka ovat liian monimutkaisia arvioitaviksi kokonaisina, ja jotka ovat muuten epävarmoja. (Jorgensen 2004.)

2.1.3 Työmäärien arviointi ketterillä menetelmillä toteutettavissa projekteissa

Ketterillä menetelmillä (Agile development, Scrum) toteutettavilla tietojärjestelmäprojekteilla on käytössä perinteisistä poikkeavia tapoja työmäärien arviointiin. Tehtävät, joita kutsutaan termillä ”user story” luokitellaan suuruuden tai monimutkaisuuden mukaan käyttäen ”story points”-pisteystystä. Pisteasteikkona voidaan käyttää esimerkiksi Fibonaccin sarjaa: 1, 2, 3, 5 ja 8. Jos arvioidaan isompia kokonaisuuksia, voidaan laajentaa skaalaa lukuihin 13, 20, 40 ja 100. Pisteiden arvolla ei sinänsä ole merkitystä, kunhan tehtävien pisteytys kuvaa vaikeusastetta suhteessa muihin tehtäviin. Esimerkiksi kaksi pistettä saanut tehtävä on tuplasti niin vaikea kuin yhden pisteen saanut tehtävä.

Pisteytystä tehtäessä ei vielä oteta kantaa siihen, kuinka kauan työ kestää. Kuitenkin jos on kokemusta tiimin tuottavuudesta, eli tiedetään kuinka monen pisteen arvosta tehtäviä tiimi on aiemmin tehnyt tietyn ajanjakson aikana, voidaan arvioida pisteytettyjen tehtävien kesto suhteessa aiempaan tuottavuuteen. (Cohn 2009,35-41, 52, 54.)

Tiimin tuottavuutta arvioidessa on syytä ottaa huomioon, että jäsenet harvoin pystyvät käyttämään sataprosenttisesti aikaansa yhteen tehtävään. On otettava huomioon yllättävät häiriötekijät, kokoukset ja hallinnollisiin asioihin kuluva aika. Jos arvioitu työmäärä on viisi päivää, se ei tarkoita sitä, että tehtävä on valmis viiden kalenteripäivän kuluessa. Todellinen tuottava työaika riippuu organisaatiosta ja henkilön roolista. (Cohn 2009, 42-48.)

Vaikka tietyn tehtävän arvioiminen onnistuu todennäköisimmin parhaiten henkilöltä, jolla on kokemusta vastaavista tehtävistä, ketterissä menetelmissä suositellaan arviointia aina koko tiimin voimalla. Tehtävät on jaoteltu siten, että niiden toteuttaminen vaatii useamman kuin yhden asiantuntijan työpanoksen ja osaamista. Sellaista asiantuntijaa on vaikea löytää, joka osaisi yksin arvioida kaikki tehtävän vaatimat erikoisalueet. Siksi yhden asiantuntijan arvio ei riitä, vaan arvioita on hyvä käydä läpi koko tiimin kesken ja kyseenalaistaa myös siltä varalta, että tehtävä on ymmärretty väärin. Ohjelmoijat, testajat, tietokantasuunnittelijat, arkkitehdit, käyttöliittymäsuunnittelijat ja muut projektin ohjelmistokehitystyötä tekevät ammattilaiset osallistuvat työmäärien arviointiin. Asiaa parhaiten tuntevan henkilön arvio voi pitää paikkansa, jos hän itse tekee tehtävän. Tiimeissä työ voidaan kuitenkin jakaa niin, että joku toinen saa tehtävän hoitaakseen, ja tällöin työmäärä saattaa olla isompi. Cohn (2009, 54) suosittaa käyttämään työmäärien arvioinnissa yhdistettynä kolmea eri menetelmää, jotka ovat asiantuntija-arvio, analogia ja tehtävien pilkkominen pienempiin osiin. (Cohn 2009, 51-54.)

Ketterien menetelmien erikoisuus on Planning Poker. Tässä menetelmässä on yhdistetty asiantuntija-arviot, analogiset arviot sekä tehtävien pilkkominen. Menetelmä on nopea tapa saada luotettavia arvioita aikaan. Suunnittelupokeriin kokoontuu koko tiimi, joka ketterissä projekteissa on mielellään alle 10 hengen kokoinen. Osallistujille jaetaan numerokortit, joiden arvoina on esimerkiksi 0, 1, 2, 3, 5, 8, 13, 20, 40 ja 100. Jokainen tiimin jäsen näyttää kortilla oman arvionsa käsiteltävästä tehtävästä, user storysta. Hen-

kilöt, jotka ovat antaneet suurimman ja pienimmän arvion, esittävät perustelunsa, ja niistä keskustellaan. Tämän jälkeen sama tehtävä arvioidaan uudelleen. Tätä toistetaan, kunnes tiimi on yksimielinen arviosta. Yleensä kolme kierrosta riittää yhteisymmärryksen saavuttamiseen. Tarkoitus ei ole saavuttaa täydellistä tarkkuutta arvioinnissa, eikä pohtia asioita liian syvällisesti ja kauan. (Cohn 2009, 56-60.)

Suunnittelupokeri toimii hyvin, koska arviointiin saadaan kaikkien asiantuntijoiden näkemykset, ja lopullinen arvio on yksilöiden arvioiden keskiarvo, jonka on todettu olevan parempi kuin yhden asiantuntijan arvio. Lisäksi suuret ja pienet arviot on perusteltava järkevästi. Arviointi tällä menetelmällä on myös hauska ja sosiaalinen tapahtuma. (Cohn 2009, 56-60; Molokken & Jorgensen 2003.)

2.2 Malleihin perustuvat arviot

Algoritmi on matemaattinen malli, jolla yritetään laskea tietojärjestelmän tuottamiseen tarvittava työmäärä perustuen rakennettavan järjestelmän ja sitä tuottavan organisaation tietoihin. Useimmiten käytettävät perusparametrit liittyvät ohjelmiston kokoon ja monimutkaisuuteen, joita voidaan mitata eri tavoin, sekä organisaation arvioituun tuottavuuteen. Algoritmien kehittämisessä on käytetty erilaisista projekteista kerättyjä historiatietoja, joista on löydetty muuttujiin reagoivat vastaavuudet. Historiatietoihin perustuvat algoritmit eivät ole parhaita mahdollisia projekteissa, jotka perustuvat uudempaan tekniikkaan tai joissa käytetään ketteriä menetelmiä (Haikala & Märijärvi 2004 , 242; Sommerville 2004, 620.)

Ensimmäiset algoritmeihin perustuvat arviointimallit kehitettiin jo 1970-luvulla, ja niiden laatimista varten oli tutkittu laaja joukko toteutuneita ohjelmistoprojekteja. Ensimmäinen arviointialgoritmi oli SLIM eli Software Lifecycle Management, jonka kehitti Lawrence Putnam jo 1970-luvulla yli 70.000 koodirivin kokoisten järjestelmien arviointiin. COCOMO-mallin kehittäjä Barry Boehm oli ensimmäinen tutkija, jonka lähtökohta tietojärjestelmien arviointiin oli ensisijaisesti taloudellinen. Algoritmien ongelma on, että kun ohjelmiston koko ja monimutkaisuus kasvavat, arviointiin vaikuttavat tekijät muuttuvat yhä monimuotoisemmiksi ja parametreja on vaikea hallita. (Chandrasekaran & Venkatesh Kumar 2012.)

2.2.1 Algoritmin perusmalli ja parametrit

Perusmuodossaan arviointialgoritmi voidaan kuvata esimerkiksi näin:

$$\text{Työmäärä} = A * \text{Koko}^E * M$$

A on tuottavuuskerroin, joka riippuu paikallisista käytännöistä ja siitä, minkä tyyppistä ohjelmistoa ollaan kehittämässä.

Koko voi olla joko ohjelmakoodin rivimäärä tai ohjelman toiminnallisuuden määrä mitattuna esimerkiksi toimintopisteillä. Koon arvioinnissa voidaan käyttää vertailukoh- tana aikaisempia vastaaventyyppisiä järjestelmäprojekteja. Toimintopisteiden avulla voidaan myös arvioida tulevaa koodin määrää luokittelemalla mahdolliset tulevat oh- jelmakomponentit koon mukaan.

E on vakio, jonka arvo on yleensä 1 – 1,5. Tällainen kokoon liittyvä eksponentti on useimmissa arviointialgoritmeissa, koska kustannukset eivät yleensä nouse lineaarisesti ohjelmiston koon kasvaessa. Ylimääräisiä kustannuksia tulee, koska kommunikointi on vaikeampaa isommissa sovelluskehitystiimeissä ja ison järjestelmän kokoonpanon hal- linta ja integrointi on monimutkaisempaa. Mitä suurempi järjestelmä, sitä suurempi on eksponentti E:n arvo.

M on kustannuskerroin, joka muodostetaan arvioimalla esimerkiksi kehittämissympäris- töön ja –prosessiin liittyviä ominaisuuksia, ohjelmistolle asetettuja vaatimuksia ja oh- jelmistokehittäjien osaamista ja kokemusta vastaavista järjestelmistä.

(Jorgensen 2007; Sommerville 2004, 623-624.)

Ohjelmiston kokoa mitattiin vanhimmissa algoritmeissa ohjelmakoodin rivimäärällä; tästä käytetään termejä SLOC (Source lines of code) tai KLOC (1000 lines of code). Koodin suuri määrä ei välttämättä tarkoita, että ohjelma on monimutkainen, koska koodia voi generoitua joillakin ohjelmointikielillä tai sovelluskehittimillä paljonkin yk- sinkertaisesta ominaisuudesta. Toinen tapa mitata järjestelmän kokoa onkin toiminto- pisteiden mittaaminen. Voidaan käyttää esimerkiksi FPA-menetelmää (Function Point Analysis), jota kuvataan tämän tutkimuksen kappaleessa 2.2.3, tai laskea niin sanottuja objektipisteitä (Object Points). (Haikala & Merijärvi 2004, 244; Sommerville 2004, 618.)

Objektipisteiden (Object Points) laskemista varten tutkitaan, minkälaisia näyttöjä, raportteja ja ohjelmamoduuleja tuleva ohjelmisto sisältää. Jokainen ohjelman osa pisteutetaan niiden vaikeusasteen mukaan taulukossa 1 kuvatulla tavalla, ja pisteet lasketaan yhteen. Tuloksena saadaan järjestelmän objektipisteiden määrä. Objektipisteitä voidaan käyttää algoritmeissa koko-parametrin arvoina etenkin silloin, kun järjestelmä tuotetaan olio-ohjelmoinnilla.

Taulukko 1. Objektipisteiden määräytyminen

Objektien pisteytys vaikeusasteen mukaan	Yksinkertainen	Keskitaso	Monimutkainen
Näyttö	1	2	3
Raportti	2	5	8
Ohjelmamoduuli	10	10	10

(Sommerville 2004, 618.)

Algoritmien koko-parametrin arviointi on vaikeaa projektin alkuvaiheessa. Kaikki koodin määrään vaikuttavat tekniset ratkaisut eivät välttämättä ole tiedossa arviointivaiheessa. Eri ohjelmointikielillä generoituu myös erilainen määrä ohjelmakoodia samasta toiminnallisuudesta, ja vaikka itse ohjelmointi voi olla helpompaa, isomman koodimäärän validointi on kuitenkin työläämpää. Toimintopisteiden ja objektipisteiden määrittely on helpompaa, mutta siltikin usein epätäsmällistä. Ohjelmiston kokoon ja vaativuuteen vaikuttavien tekijöiden painoarvon määrittely on subjektiivista ja riippuu henkilön taustasta ja kokemuksesta vastaavien järjestelmien parissa. (Sommerville 2004, 623-624.)

2.2.2 COCOMO-malli

Constructive Cost Model eli COCOMO on malli, jonka ensimmäisen version julkaisi Barry Boehm 1981. Se perustui kokemuksiin vesiputousmallilla toteutetuista projekteista ja vanhoilla ohjelmointikielillä, kuten C tai Fortran, toteutetuista ohjelmista. Mallia on kehitetty edelleen, ja uusien versio COCOMO II on vuodelta 2000. Versio II ottaa huomioon esimerkiksi protoilun ohjelmistokehitystyössä, spiraalimaisen kehitysprosessin ja komponenttien uudelleenikäytettävyyden, mutta ei kuitenkaan sisällä kaikkia uu-

simprien tekniikoiden tuomia erityispiirteitä. (Haikala & Märijärvi 2004, 245; Sommerville 2004, 624-626.)

COCOMO-mallista on paljon julkista dokumentaatiota ja se on laajalti käytetty ja arvostettu eri yrityksissä (Sommerville 2004, 625). Mallista on olemassa neljä eri alimallia. Application-composition-malli on tarkoitettu uudelleenkäytettävien osien ja prototyyppien kehittämisen arviointiin. Early design-malli on suppea versio, jota voidaan käyttää projektin alkuvaiheessa. Reuse-malli on sellaisia projekteja varten, joissa integroidaan valmiita ohjelman osia tai tehdään paljon koodia generoimalla. Post-architecture on tarkimpia arvioita tuottava malli, jota voidaan soveltaa, kun käytettävissä on jo täsmällistä tietoa ohjelmiston koosta.

(Haikala & Märijärvi 2004, 245; Sommerville 2004, 626-627.)

COCOMO-malli on monimutkainen siinä käytettyjen kymmenien eri parametrien ja skaalaus- sekä kustannuskertoimien vuoksi. Laskennan lähtökohtana käytetään Reuse ja Post-architecture-malleissa ohjelman kokoa koodiriveinä (SLOC =Source lines of code), kun muissa malleissa lähtökohtana on sovelluksen toimintopisteet tai objektipisteet. Alustava työmääräarvio kuukausissa (Person Months, Nominal Schedule) lasketaan esimerkiksi Post architecture-mallissa kaavalla

$$PM_{NS} = A * Koko^E * EM_1 * EM_2 \dots * EM_n$$

A on tuottavuuskerroin, esimerkiksi käytetty henkilötyömäärä/ 1000 koodiriviä. Mallissa arvo on 2.94 kuukautta, mutta sitä voidaan räätälöidä tarpeen mukaan.

Koko on koodirivien määrä

E on koon skaalaustekijä, joka kuvaa työmäärän kasvamista suhteessa projektin koon. Luku muodostetaan omalla kaavallaan, jonka vakiot saadaan COCOMO-mallin taulukosta. Perustilanteessa E:n arvoksi tulee noin 1,1. Skaalaustekijät on kuvattu liitteessä 1 olevassa taulukossa 2.

EM₁...EM_n ovat kustannuskertoimia, jotka liittyvät ohjelmiston vaatimuksiin, henkilöiden kyvykkyyteen ja aikatauluun. Normaalitilanteessa kertoimet ovat arvoltaan ykkösiä, mutta kasvavat projektin vaativuuden mukaan. Yksi kustannuskertoimista on SCED eli aikakerroin, jonka arvo on alustavaa työmäärää laskettaessa 1. Kustannusker-

toimet on kuvattu liitteessä 1 olevassa taulukossa 3. (Haikala & Märijärvi 2004, 245-248; Chandrasekaran & Venkatesh Kumar 2012.)

Jos projektin aikataulua halutaan kiristää, lasketaan vaikutus työmäärään kertomalla edellä mainitulla kaavalla saatu alustava työmäärä PM_{NS} aikataulukertoimella SCED.

Jos alustava työmäärä on esim. 50 henkilötyökuukautta, ja halutaan lyhentää aikataulua 15%:lla, käytetään kerrointa 1,14 ja saadaan lopulliseksi työmääräksi 70 htkk.

(Haikala & Märijärvi 2004, 245-248; Sommerville 2004, 631-634.)

COCOMOn avulla tuotetut arviot ovat tarkkoja, jos on käytettävissä riittävästi historia-tietoa tai ulkopuolisen asiantuntijan apua juuri omalle organisaatiolle sopivien paramet-risarvojen määrittämistä varten. Mallin vakioarvojen sopivuutta omiin projekteihin voi muuten joutua arvailemaan, eivätkä tulokset välttämättä osu kohdalleen. Hankalien laskukaavojen ja parametrien syvällinen ymmärtäminen voi osoittautua liian työlääksi perusprojektipäällikölle. (Sommerville 2004, 634.)

2.2.3 FPA eli toimintopisteanalyysi

FPA eli Function Point Analysis on IBM:n Allan Albrechtin vuonna 1979 julkaisema menetelmä tietojärjestelmän toiminnallisen koon mittaamiseksi. Menetelmä perustuu järjestelmän toiminnallisuuden arvioimiseen käyttäjän näkökulmasta. Arviointia varten tarkastellaan syötteiden, tiedostojen, tulosteiden jne. lukumäärää ja laatua ja saadaan tuloksena ohjelman toimintopisteiden lukumäärä. Menetelmä on riippumaton järjes-telmän teknisestä toteutustavasta ja perustuu enemmänkin järjestelmän tarvitsemaan ja tuottamaan tietoon sekä liiketoiminnan siihen kohdistamiin vaatimuksiin. Työmäärän arvioimiseen toimintopisteiden avulla tarvitaan historiatietoja vastaavien toimintojen aiemmista toteutuksista, jotta voidaan määrittää kerroin, jolla toimintopisteet muunne-taan työmääräksi. (Fetcke 1999; Haikala & Märijärvi 2004, 244; Thompson 2003.)

Nykyään FPA-menetelmästä on useita eri suuntauksia. Alkuperäistä menetelmää tuke-maan perustettiin 1984 International Function Point Users Group (IPFUG), joka jul-kaisee laskentasäännöt sisältävää ohjekirjaa Counting Practices Manual (CPM) . FPA-menetelmästä edelleen kehitettyjä muunnelmia ovat mm. Charles R. Symonsin 1988

julkaisema ja United Kingdom Software Metrics Associationin (UKSMA) ylläpitämä Mark II FPA ja Denis St-Pierren 1997 kehittämä Full Function Points. Menetelmissä on pieniä eroja, mutta kaikissa kolmessa analysoidaan toimintopisteitä kaksivaiheisesti. Ensin tutkitaan ohjelman dokumentteja, käsittelysääntöjä ja transaktioita käyttäjän näkökulmasta. Toisessa vaiheessa muunnetaan löydetty yksiköt toimintopisteiksi. Ensimmäinen vaihe on puhtaasti ihmisten tekemää työtä, mutta toinen vaihe voidaan automatisoida. Markkinoilta löytyy sovelluksia toimintopisteiden laskentaa varten, esimerkiksi Charismatek Software Metrics:in Function Point Workbench™ (Lalcrest). (Fetcke 1999.)

Toimintopisteiden laskenta aloitetaan etsimällä järjestelmästä viiden eri loogisen komponenttityypin esiintymiä: järjestelmän sisäisiä tiedostoja, liittymätiedostoja, syötteitä, tulosteita ja kyselyitä. Löydetty komponentit pisteytetään niiden monimutkaisuuden mukaan taulukossa 4 kuvatulla tavalla. Kun nämä pisteet lasketaan yhteen, saadaan 'sovitamattomat' toimintopisteet (Unadjusted FP), jotka pitää kertoa sovitustekijällä (Value Adjusted Factor VAF), jotta saadaan lopulliset toimintopisteet. (Adjusted Function Points AFP). Sovitustekijä määräytyy 14 erilaisen järjestelmältä vaadittavan teknisen ominaisuuden mukaan, joille annetaan arvot nolasta viiteen. Tällaisia tekijöitä ovat esimerkiksi tietoliikenne, hajautettu käsittely, suorituskyky, monimutkaiset käsittelysäännöt, asennuksen helppous ja komponenttien uudelleen käytettävyys. Tekniset ominaisuudet on kuvattu liitteessä 1 olevassa taulukossa 5. Järjestelmän teknisten ominaisuuksien yhteenlaskettu kerroin (TDI) voi olla arvoltaan 0-70. Lopullisten toimintopisteiden laskennassa käytetty kaava on erilainen riippuen siitä, onko kyseessä esim. kehitysprojekti tai ylläpitoprojekti. (Savolahti 2002.)

Taulukko 4. Function Count Weighting Factors

Factor	Low	Average	High
Internal Logical File (ILF)	7	10	15
External Interface File (EIF)	5	7	10
External Input (EI)	3	4	6
External Output (EO)	4	5	7
External Query (EQ)	3	4	6

(Chandrasekaran & Venkatesh Kumar 2012; Meli & Santillo 1999.)

Jotta lopulliset toimintopisteet voidaan muuttaa työmääräksi, on tiedettävä organisaation tuottavuus tämäntyyppisissä projekteissa ja arvioitava montako henkilötyöpäivää tarvitaan yhden toimintopisteen tuottamiseen. Työmäärä riippuu mm. ohjelmointikielstä, koodin määrästä ja tekijöiden kokemuksesta. (Chandrasekaran & Venkatesh Kumar 2012.)

Koska toimintopisteiden laskentaa ei voida tehdä ilman riittävää dokumentaatiota, toimintopisteanalyysi ei sovellu käytettäväksi aivan projektin alkuvaiheessa, ennen kuin järjestelmästä on olemassa riittävät vaatimusmäärittelyt. Vanhan järjestelmän ylläpito- projektien arviointi ei myöskään onnistu, jos dokumentit eivät ole kunnossa. (Chandrasekaran & Venkatesh Kumar 2012; Meli & Santillo 1999.)

2.2.4 Käyttötapauksiin perustuva menetelmä

Työmäärien arviointi käyttötapausten avulla on eräänlainen muunnelmä FPA-menetelmästä, ja se sopii etenkin oliopohjaisten järjestelmien arviointiin. Sen kehitti Gustav Karner (Objectory AB) vuonna 1993. Käyttötapauspisteiden eli UCP-pisteiden (Use Case Points) määrittely tehdään analysoimalla järjestelmän ulkopuolisia toimijoita (actors), käyttötapausten monimutkaisuutta ja järjestelmän teknisiä ominaisuuksia. (Fetcke, Abran & Nguyen 1997; Schneider & Winters 2000, 140.)

Järjestelmän käyttötapauksiin määriteltyjä erilaisia toimijoita ovat esimerkiksi käyttäjä tekstipohjaisen käyttöliittymän välityksellä, käyttäjä graafisen käyttöliittymän välityksellä tai toinen järjestelmä tietoliikenneyhteyden välityksellä. Järjestelmän kaikki erilaiset käyttäjäroolit etsitään ja pisteytetään taulukossa 6 kuvatulla tavalla ja pisteet lasketaan lopuksi yhteen. (Schneider & Winters 2000, 140-141.)

Taulukko 6: Actor Weighting Factors

Actor type	Description	Factor
Simple	Program interface	1
Average	Interactive, or protocol-driven interface	2
Complex	Graphical interface	3

(Schneider & Winters 2000, 141.)

Käyttötapaukset luokitellaan yksinkertaisiin, keskinkertaisiin ja monimutkaisiin. Luokittelu voidaan tehdä kahdella eri tavalla. Voidaan tutkia, paljonko erilaisia tapahtumia kussakin käyttötapauksessa on. Kaikki eri skenaariot lasketaan omiksi tapahtumikseen. Jos käyttötapauksessa on 3 tapahtumaa tai vähemmän, se luokitellaan yksinkertaiseksi. Keskinkertaisessa käyttötapauksessa on 4-7 tapahtumaa ja monimutkaisessa on enemmän kuin 7 tapahtumaa. Toinen tapa luokitella käyttötapaukset on laskea montako luokkaa on yhdessä käyttötapauksessa – mikäli järjestelmän suunnittelu on jo niin pitkällä, että luokat on jo tunnistettu. Jos käyttötapauksessa on vähemmän kuin 5 luokkaa, se luokitellaan yksinkertaiseksi. Keskinkertaisessa käyttötapauksessa on 5-10 luokkaa ja monimutkaisessa enemmän kuin 10 luokkaa. Kunkin yksinkertaisen käyttötapauksen pistearvo on 5, keskinkertaisen 10 ja monimutkaisien 15. Jos järjestelmässä on vaikkapa 3 yksinkertaista käyttötapauksia, 9 keskinkertaista ja 2 monimutkaista, saadaan yhteenlasketuksi käyttötapauksien pistearvoksi $3 * 5 + 9 * 10 + 2 * 15 = 147$. Käyttäjäroolien ja käyttötapauksien yhteenlaskettua pistearvoa kutsutaan UUCP:ksi eli sovittamattomiksi käyttötapauspisteiksi. (Schneider & Winters 2000, 141-142.)

Aivan kuten FPA-menetelmässä, seuraavaksi tutkitaan järjestelmän teknisiä ominaisuuksia tai vaatimuksia ja pisteytetään ne. Kun FPA-laskennassa oli 14 eri ominaisuutta, käyttötapauspisteityksessä niitä on 13. Pisteytettävät ominaisuudet ovat molemmissa samankaltaisia, esimerkiksi järjestelmän suorituskyky, uudelleenkäytettävyys, helppokäyttöisyys, muutosten teon helppous ja järjestelmän asentamisen yksinkertaisuus. Kunkin ominaisuuden kriittisyys kehitettävälle järjestelmälle arvioidaan pisteillä nolasta viiteen. Ominaisuuksien pisteet painotetaan lisäksi UCP-menetelmässä vakioina annetuilla painokertoimilla 0,5 – 2 ja lasketaan yhteen. Tuloksena saadaan teknisten ominaisuuksien painotetut pisteet eli TFactor. Teknisistä ominaisuuksista laskettua arvoa kutsutaan TCF:ksi (Technical Complexity Factor) ja se saadaan edellä mainituista järjestelmän teknisistä painotetuista pisteistä kaavalla:

$$TCF = 0.6 * (0.01 * TFactor)$$

(Schneider & Winters 2000, 143-144.)

Vielä on pisteytettävä projektin henkilöstöön tai ympäristöön liittyvät tekijät EF-luvuksi (Environmental Factor). Arvioitavat ominaisuudet ovat kokemus RUP-menetelmästä, sovellusosaaminen, kokemus olio-ohjelmoinnista, järjestelmävastaavan tai –arkkitehdin osaaminen, tiimin motivaatio, odotettavissa olevat muutokset projektissa, osa-aikaisten työntekijöiden määrä sekä ohjelmointikielen vaikeusaste. Ominaisuudet arvioidaan pisteillä nolasta viiteen. Esimerkiksi viisi pistettä kohdassa motivaatio tarkoittaa, että ryhmä on erittäin motivoitunut ja yksi piste kohdassa ohjelmointikielen vaikeusaste tarkoittaa hyvin yksinkertaista ohjelmointikieltä. Kullekin tekijälle on lisäksi määrätty vakiona painokertoimet -1 – 2. Tekijät ja painokertoimet on kuvattu taulukossa 7. Arvioidut pisteet kerrottuna painokertoimella lasketaan yhteen tekijäksi EFactor, ja niistä muodostetaan EF-pisteet kaavalla:

$$EF = 1.4 * (-0.03 * EFactor).$$

(Schneider & Winters 2000, 144-145.)

Taulukko 7: Environmental Factor for Team and Weights

Factor number	Description	Weight
F1	Familiar with Rational Unified Process	1.5
F2	Application experience	0.5
F3	Object-oriented experience	1
F4	Lead analyst capacity	0.5
F5	Motivation	1
F6	Stable requirements	2
F7	Part-time workers	-1
F8	Difficult programming language	-1

(Schneider & Winters 2000,145.)

Kun on analysoitu ja laskettu pisteet käyttäjärooleista, käyttötapausten toiminnallisuudesta ja ympäristön ja henkilöstön vaikutuksesta, lasketaan UCP-pisteet kaavalla:

$$UCP = UUCP * TCF * EF.$$

(Schneider & Winters 2000, 146.)

Käyttötapauspisteiden muuntaminen työmääräksi voidaan tehdä yksinkertaisimmin kertomalla UCP-pisteet 20 henkilötyötunnilla. Tämä on Karnerin alkuperäinen sääntö. Schneider ja Winters (2000, 146) suosittelevat kuitenkin EF-tekijöiden arvioimista tarkemmin, koska juuri näillä tekijöillä on suuri vaikutus suoraan työmääriin. Taulukon 7 tekijöistä F1 – F6 lasketaan yli kolme pistettä saaneiden tekijöiden ja tekijöistä F7 ja F8 alle kolme pistettä saaneiden tekijöiden lukumäärä yhteen. Jos riskiarvot saavuttaneiden tekijöiden lukumääräksi saadaan kaksi tai vähemmän, voidaan käyttää Karnerin ehdottamaa 20 henkilötyötuntia työmäärien laskentaan. Jos taas lukumäärä on kolmesta neljään, käytetään laskennassa 28 henkilötyötuntia UCP-pistettä kohden. Jos lukumäärä on viisi tai enemmän, Schneider ja Winters (2000, 146) suosittelevat tekemään muutoksia projektiin, niin että riskialttiiden ympäristötekijöiden määrä vähenee, ja tekemään sitten uudet laskelmat. Esimerkiksi tekijää F7 voidaan parantaa muuttamalla osa-aikaisten

työntekijöiden sopimukset kokopäiväisiksi, ja tekijöitä F1, F2 ja F8 voidaan parantaa koulutuksella. (Schneider & Winters 2000, 146.)

2.3 Muut menetelmät

Työmäärien arvioinnissa käytetään edellä kuvattujen asiantuntijoiden tekemien ja malleihin perustuvien menetelmien lisäksi lukuisia muita tapoja esimerkiksi sen vuoksi, että käytössä ei ole riittävästi tietoa, aikaa tai osaamista perusteelliseen arvion tekoon. Joskus työ on vain yritettävä tehdä tiettyyn päivään mennessä, vaikka se ei realistisesti arvioiden ole mitenkään mahdollista. Seuraavissa kappaleissa on kuvattu Parkinsonin laki ja Pricing to win esimerkkeinä tavoista, joilla ei edes pyritä mahdollisimman suureen tarkkuuteen työmäärien ennustamisessa.

2.3.1 Parkinsonin laki

Parkinsonin lain mukaan työ vie sille varatun ajan ja resurssit, eli arviot ovat itseään toteuttavia. Työmäärä kasvaa, kun aikataulu on väljä, tai työtä tehdään tehokkaammin, kun ylimääräistä aikaa ei ole käytettävissä. Kun projektin valmistumispäivä on ylempältä taholta määrätty esimerkiksi lakimuutosten takia, otetaan työmääräarvioinnissa lähtökohdaksi käytössä oleva aika ja resurssit. Työmäärä lasketaan kertomalla käytettävissä olevat resurssit käytettävissä olevalla ajalla. Jos projektin on oltava valmis esim. puolen vuoden kuluttua ja käytettävissä on kolmen ihmisen työpanos, saadaan työmääräksi $6 * 5 = 30$ henkilötyökuukautta. Mitään varmuutta tällä laskelmalla saadun ajan riittäväyydestä ei kuitenkaan ole, vaan tällä tavalla saadaan ainoastaan työmäärän alaraja. Menetelmä tuottaa tiukkoja ja epärealistisia arvioita.

(Pelin 2011, 115; Sommerville 2004, 621.)

2.3.2 Pricing to win eli taloudelliseen voittoon tähtäävä arviointi

Jos tiedetään, mitä asiakas on enimmillään valmis maksamaan projektista, saatetaan työmääräarviota kasvattaa tarkoituksellisesti niin, että saavutetaan myyjälle mahdollisimman paljon voittoa tuottava hinta. Mahdollisesti jo lähtökohtaisesti työmäärät arvioidaan eri tavalla riippuen siitä, tehdäänkö järjestelmää omalle yritykselle vai asiakkaalle (Molokken 2003). Työmääriä ei aina kerrota asiakkaalle - eivätkä ne asiakasta välttämät-

tä edes kiinnosta - ainoastaan valmiin järjestelmän hinta. Toisaalta tiukassa kilpailutilanteessa on mahdollista myydä projekti tappiollakin, jos halutaan säilyttää asiakas. Pricing to win ei oikeastaan ole mikään arviointimenetelmä, eikä sillä ole mitään tekemistä järjestelmän toiminnallisuuden kanssa, vaan asiakkaan budjetti tai tavoiteltava hyöty ratkaisee arvion tuloksen. Mahdollisesti järjestelmän tarkasta sisällöstä neuvotellaan asiakkaan kanssa vasta sen jälkeen, kun on lyöty lukkoon sen hinta. Varsinainen työmääräarviointi aloitetaan vasta, kun sisällöstäkin on sovittu.

(Molokken 2003; Sommerville 2004, 621-622.)

Toisaalta asiantuntija voi tutkimusten mukaan alitajuisesti ajautua soveltamaan Pricing to win menetelmää, jos eri tahojen tavoitteet ovat tiedossa. Jos ennen työmääräarvion tekoa tiedetään, että asiakas olettaa työn kestävän 1000 tuntia, tiedon on todettu vaikuttavan arvion suuruuteen siten, että arvio lähestyy asiakkaan oletusta (Jorgensen & Grimstad 2006). Arviointiin voi myös tuoda ylioptimistisuutta se, että halutaan tehdä mahdollisimman edullinen tarjous asiakkaalle kilpailun voittamiseksi (Jorgensen 2004).

2.4 Aiemmat tutkimukset

Molokken ja Jorgensen (2003) ovat tehneet yhteenvedon muiden tutkijoiden tekemistä tutkimuksista liittyen tietojärjestelmäprojektien työmääräarviointimenetelmiin sekä arvioitujen työmäärien ja kustannusten toteutumiseen. Tutkimukset on tehty vuosina 1984 – 2002 ja ne osoittavat, että 60-80% projekteista ylittää budjettinsa ja aikataulunsa. Ylityksen määrä on tutkimusten mukaan keskimäärin 30-40%. Yhden tutkimuksen tulokset poikkeavat kuitenkin merkittävästi muista: Standish Groupin ”kaaosraportin” 1994 mukaan projektit ylittivät arvioidut kustannukset keskimäärin 89%:lla. Standishin tutkimus sai aikaan paniikkia ja vääriä uskomuksia, ja lisäsi taipumusta liioitella työmääräarvioita. Molokken ja Jorgensen (2003) eivät kuitenkaan pidä Standishin raporttia tieteellisesti luotettavana, eikä yritys ole suostunut julkaisemaan tutkimusaineistoaan muille tutkijoille liikesalaisuuksiin vedoten. Työmääräarvioiden ylitysten syistä ei ole tehty kattavaa tutkimusta, mutta Chandrasekaran ja Venkatesh Kumar (2012) esittävät, että tärkein syy on arvioinnissa käytettävien parametrien epätarkkuus ja tiedon puute, kun projekteja arvioidaan hyvin aikaisessa vaiheessa. (Molokken & Jorgensen 2003.)

Molokkenin ja Jorgensenin (2003) tekemän yhteenvedon mukaan ainakin länsimaisissa yrityksissä käytetään eniten erilaisia asiantuntijoiden tekemiin arvioihin perustuvia menetelmiä työmäärien arvioinnissa. Muita menetelmiä, kuten malleihin perustuvia arvioita, käytetään vähemmän, ja usein yhdistettyinä asiantuntija-arvioihin. Tutkijoiden mukaan tämä johtuu mahdollisesti siitä, että malleihin perustuvien arvioiden hyödyistä ei ole tieteellistä näyttöä. (Molokken & Jorgensen 2003.)

Magne Jorgensen on tehnyt vastaavan yhteenvedon työmääräarvioinnin menetelmien käyttöä tutkivista tutkimuksista vuonna 2007. Kaksi kolmasosaa Jorgensenin analysoimista tutkimuksista osoitti, että asiantuntija-arvioon perustuva työmääräarvio on keskimäärin parempi kuin malleihin perustuva. Mallien käytön huonompi tarkkuus johtuu Jorgensenin (2007) mukaan siitä, että mallien parametreja ei ole sovitettu yritykseen ja ympäristöön, jossa niitä käytetään. Lisäksi asiantuntijoilla on sellaista järjestelmään liittyvää hiljaista tietoa, joita mallit eivät sisällä. Neljäsosa tutkimuksista osoitti, että yhdistelmä asiantuntija-arviosta ja jonkun mallin käytöstä tuottaa tarkkuudeltaan parhaita arvioita. Myös Sommerville (2004) suosittelee kirjassaan asiantuntija-arvioiden ja mallien yhteiskäyttöä. Arvioinnit tulisi tehdä useita eri tekniikoita käyttäen, ja vertailla syntyneitä lukuja. Jos eri tavoin tehdyissä arvioissa on suuria eroja, se voi tarkoittaa, että arvioiden perusteena ei ole riittävästi tietoa. Mitä tarkemmin tunnetaan projektin vaatimukset ja sen myötä eri arviointitavoissa käytettävät tekijät, sen suurempiin tarkkuuksiin päästään. (Jorgensen 2007; Sommerville 2004, 621, 638.)

Asiantuntijoiden tekemien arvioiden luotettavuus vaikuttaa kyseenalaiselta Simula Research Laboratoryn julkaisemassa tutkimuksessa, jossa Jorgensen ja Grimstad (2006) tutkivat, miten arvioijan saama informaatio tulevasta järjestelmästä vaikutti arvion suuruuteen. Kokeessa ryhmä A sai lyhyen vaatimusluettelon, jossa oli vain järjestelmän kannalta tarpeellisia tietoja. Ryhmä B sai saman vaatimusluettelon, mutta sen seassa oli turhaa tietoa esim. muista sovelluksista, jotka olisivat käytössä samalla työasemalla. Ryhmä B:n työmääräarvio oli noin kaksinkertainen ryhmän A arvioon verrattuna. Samankaltainen lopputulos saatiin antamalla molemmille ryhmille täsmälleen samat tiedot, mutta ryhmän A teksti oli tiivistetty yhdelle A4-sivulle, ja ryhmän B teksti oli venytetty seitsemän sivun pituiseksi käyttämällä isompaa fonttia ja suurempaa riviväliä. Pitempi vaatimusluettelo tuotti suuremman työmääräarvion. Ihmisaivot eivät kykene ko-

vin hyvin suodattamaan harhaanjohtavaa ja epärelevanttia tietoa, vaan turhatkin tiedot vaikuttavat arvioijan käsitykseen järjestelmän monimutkaisuudesta. Jotta tällainen ei-rationaalinen vaikutus vältettäisiin, Jorgensen ja Grimstad (2006) suosittelivat, että arviointiin osallistuisi aina useampi kuin yksi henkilö. Ensimmäinen henkilö kokoaa vaatimusluettelon niin, että kaikki nimenomaan arvioitavaa tehtävää koskeva tieto jätetään pois. Kun arviointi on tehty, voidaan arvioijalle näyttää poistettu tieto. Todennäköisesti turha tieto on tärkeää vain jossain toisessa tehtävässä tai taustojen tuntemisessa, mutta jos sillä nyt todetaan olevan merkitystä, voidaan erittäin perustetulla syyllä muuttaa alkuperäistä arviota. (Jorgensen & Grimstad 2006.)

Arvioijalla ei saisi olla Jorgensenin ja Grimstadin (2006) mukaan etukäteen mitään tietoa siitä, minkälaiseen haarukkaan työmäärän ja kustannusten toivotaan osuvan. Etukäteistiedon on todettu tutkimuksessa vaikuttavan arvion suuruuteen. Asiantuntijat itse eivät kuitenkaan myöntäneet, että tiedolla olisi ollut mitään merkitystä. Tutkijat eivät onnistuneet selvittämään, mistä vaikutus johtui, mutta todennäköisesti kyse on ihmisen alitajunnan toiminnasta, jota ei täysin tunneta. (Jorgensen & Grimstad 2006.)

3 Haastattelututkimus tietojärjestelmäprojektien työmäärien arvioinnin nykytilasta suurissa ja keskisuurissa tietojärjestelmiä toteuttavissa suomalaisissa yrityksissä

3.1 Kohderyhmä

Tutkimushaastattelun kohderyhmänä oli joukko asiantuntijoita, jotka edustavat tietojärjestelmien toimittajaorganisaatioita pääkaupunkiseudulla. Haastateltavia henkilöitä oli viisi, ja he toimivat haastatteluhetkellä järjestelmäasiantuntijoina, projekti-, sovellus- tai kehityspäälliköinä. Tutkimuksen kohteena olevilla asiantuntijoilla oli eniten kokemusta tyypillisistä tietojärjestelmien ylläpito- ja kehitysprojekteista, jotka sisältävät määrittelyä, suunnittelua, toteutusta ja testausta. Valmisohjelmistoprojektit rajattiin pois tutkimuksesta. Osa projekteista oli useamman toimittajan yhteistyönä tehtyjä ohjelmistoprojekteja. Tutkituissa projekteissa käytettiin sekä perinteistä vaihejakomallia että ketteriä menetelmiä. Asiantuntijoiden edustamat yritykset olivat keskisuuria tai suuria. Edustettuna oli sekä ulkopuoliselle asiakkaalle tietojärjestelmiä toimittavia yrityksiä, että oman yrityksen käyttöön tietojärjestelmiä tekeviä tietotekniikkayksiköitä.

Tässä lyhyt kuvaus haastateltavista asiantuntijoista ja heidän taustoistaan. Haastateltaville luvattiin, että heidän nimiään ja yrityksiään ei julkaista, joten heistä käytetään keksittyjä nimikirjaimia.

AM on työskennellyt IT-alalla 27 vuotta ja toiminut pankki-, vakuutus- ja työeläkevakuutussektoreilla. Työtehtävinä on ollut ohjelmoija, systeemisuunnittelija, tietovarasto- ja tietosuunnittelija, tietohallinnon tehtävät ja testaustehtävät. AM on toiminut sekä oman yrityksen IT:ssä että alihankkijana / konsulttina asiakasyrityksissä.

RA on työskennellyt IT-alalla 29 vuotta. Hän on toiminut mm. pankki-, tele-, myynti-, varasto-, osto- tukkukauppa- ja kirjanpitojärjestelmien parissa tehtävänimikkeillä ohjelmoija, suunnittelija, projektipäällikkö, määrittelijä ja sovelluspäällikkö. Hän on myös toiminut sekä yrityksen omassa tietohallinnossa että konsulttina asiakasyrityksissä.

AT :lla on peräti 42 vuoden kokemus tukkukaupan, vakuutusalan, raha-automaattiyhdistyksen, pankin ja julkishallinnon järjestelmistä. Hän on toiminut aluksi nauha-arkiston hoitajana, mutta edennyt siitä ohjelmoijan, suunnittelijan, määrittelijän, projektipäällikön ja osastopäällikön tehtäviin. Tällä hetkellä hän toimii alihankkijana ja järjestelmäasiantuntijana julkishallinnon järjestelmiä toimittavassa yrityksessä.

EN on toiminut 23 vuotta tietojärjestelmien kehitystehtävissä. Hän on ollut vakuutus- ja telesektorin palveluksessa ja enimmäkseen toteuttanut järjestelmiä oman organisaation käyttöön. Hän on toiminut ohjelmoijana, suunnittelijana, projektipäällikkönä ja testausvastaavana. Tällä hetkellä hän toimii projektipäällikkönä vakuutusyhtiön järjestelmäyksikössä.

AP on ollut alalla 32 vuotta suunnittelijana, järjestelmäasiantuntijana, järjestelmäpäällikkönä ja sovellusvastaavana. Hän on käynyt projektipäällikkökurseja, mutta ei ole varsinaisesti toiminut projektipäällikkönä. Hän on toiminut pankkisektorilla, graafisen alan taloushallinnossa sekä julkishallinnossa ja tehnyt järjestelmiä sekä omalle yritykselle että ulkopuolisille asiakkaille. Tällä hetkellä hän toimii sovellusvastaavana julkishallinnolle tietojärjestelmiä toimittavassa yrityksessä.

3.2 Haastattelut

Vaikka kaikilla haastateltavilla oli usean vuosikymmenen kokemus tietojärjestelmätyöstä ja työmäärien arvioinnista, he väittivät heti haastattelun aihealueen kuullessaan, että eivät tiedä aiheesta mitään. Kuitenkin haastattelun kuluessa kävi ilmi, että arviointia oli tehty paljon, ja aiheesta löytyi paljon kokemuksia, tietoa ja kehitysehdotuksia.

Haastatteluissa käytettiin apuna ennalta laadittuja kysymyksiä, jotka löytyvät liitteestä 2. Seuraavissa kappaleissa on kuvattu haastattelussa läpikäytyjä aiheita.

3.2.1 Haastateltavien urallaan arvioimien tehtävien sisältö

Haastateltavia pyydettiin kertomaan, minkälaisista tehtävistä he olivat uransa aikana tehneet työmääräarvioita.

Eniten oli arvioitu yksittäisten erikokoisten ohjelmointitehtävien työmääriä. Tällainen tehtävä sisältää useimmiten suunnittelun, ohjelmoinnin toteutuksen ja ohjelmoijan tekemän oman testauksen eli yksikkötestauksen. Joissakin tapauksessa arvioissa on ollut mukana myös määrittelyn, järjestelmätestauksen ja käyttöönoton osuudet. Tehtävän sisältö on ollut esim. jonkun uuden toiminnallisuuden lisääminen tai jonkin ulkopuolisen muutoksen vaikutuksen arviointi useaan ohjelmaan. Näistä yksittäisistä työmääräarvioista on voitu koota projekti, joka sisältää useita tehtäviä, tai sitten muutokset on toteutettu sellaisinaan ylläpitotehtävinä.

Kolme haastateltavista oli arvioinut kokonaisia projekteja tai osaprojekteja, jotka ovat sisältäneet kaikki ohjelmistokehityksen vaiheet määrittelystä käyttöönottoon. Harvemmin oli arvioitu täysin uusia järjestelmiä. Sen sijaan erilaisia konversio- ja muutosprojekteja oli arvioitu paljon ja niiden arvioinnista oli hyviä kokemuksia.

Pelkkää määrittelyä sisältäviä tehtäviä tai projekteja oli arvioinut kaksi haastateltavaa. Vain testausta sisältäviä tehtäviä tai projekteja oli tässä kohderyhmässä arvioitu vain vähän. Testaus koettiin hankalaksi alueeksi arvioida, koska kuten eräs haastateltavista totesi: ”testausta voidaan tehdä miten paljon tahansa, ja silti se ei ole tarpeeksi”.

Tässä yhteydessä useampi haastateltava totesi, että heidän ei tarvinnut juurikaan tehdä työmääräarvioita ennen vuosituhannen vaihteen suuria Y2K- ja Euro-projekteja. Toteuttajilta ei pyydetty työmääräarvioita, vaikka päälliköt saattoivat tehdä omia laskelmiaan. Pari vuosikymmentä sitten ainakin omalle organisaatiolle toteutettavat järjestelmät vain tehtiin, eikä työmääriä tai kustannuksia juurikaan mietitty silloisissa ATK-yksiköissä. Aikataulu oli kuitenkin usein ylhäältä päin annettu, joten työt vain yritettiin tehdä niin, että ohjelmat valmistuisivat ajoissa. Samantyyppistä käytäntöä on edelleen esimerkiksi pakollisissa fuusioprojekteissa, jotka ovat enemmän aikataulu- kuin kustannuskriittisiä. Niissä ei työmääräarvioiden ylityksillä ole merkitystä, ja resursseja tarvittaessa lisätään, jos aikataulut eivät muuten pidä.

3.2.2 Työmääräarvioinnin menetelmien tuntemus

Haastateltavilta kysyttiin, minkälaisia työmääräarvioinnissa käytettäviä menetelmiä he tuntevat. Kysymys herätti hilpeyttä vastaajissa, ja useimmiten he totesivat ensin, että eivät tunne muita menetelmiä kuin arvaus, muttu tai hihasta ravistelu. Muitakin menetelmiä kuitenkin löytyi, kun asiaa mietittiin pitempään.

Kolme vastaajista kertoi joskus käyttäneensä arvioinnin apuna erilaisia excel-taulukoita. AT:n yrityksessä on käytössä taulukko, joka on alun perin tehty yhden sovelluksen tarpeisiin. Sitä oli yritetty käyttää myös muissa sovelluksissa, mutta taulukon laskukaavojen antamat työmäärät ovat olleet hirvittävän suuria. Taulukkoon tarvittavien tietojen sisältöä on hankala ymmärtää muilla kuin alkuperäisellä sovellusalueella. Taulukon käyttöönotto omassa sovelluksessa olisi vaatinut paljon perehtymistä, eikä sen toimintaan ole uskottu niin paljoa, että siihen olisi haluttu käyttää aikaa.

EN muisti aikaisemmassa työpaikassa käytetyn apuna järjestelmäkohtaista taulukkoa, johon oli jaoteltu järjestelmän eri osat (näytöt, metodit, tietokantakäsittelyt jne) monimutkaisuuden mukaan suuruusluokittain ja näitä luokitteluja apuna käyttäen saatiin muodostettua järjestelmän ylläpitotehtäviin tarvittavat työmäärät. Taulukon tietoja ylläpidettiin järjestelmän eläessä, ja tietoja voitiin hyödyntää myöhemmissä työmäärien arvioinneissa.

AP oli käyttänyt aiemmassa yrityksessä taulukkoa jokaisen arvioinnin yhteydessä aivan pieniä töitä lukuun ottamatta. Taulukko oli tehty yrityksen omiin tarpeisiin, ja se oli hyvä ja toimiva. Taulukkoon piti kerätä pohjatiedoiksi lukumäärät järjestelmän osista luokiteltuna online- tai eräohjelmiin ja vaikeisiin, keskivaikeisiin tai helppoihin moduuleihin. Myös tietokantakäsittelyjen ja muiden I/O käsittelyjen lukumäärät kerättiin taulukkoon. Taulukossa oli taustalla laskukaavat, jotka antoivat lopputuloksena arvion työmäärästä. Taulukon laskemaa työmäärää käytettiin asiantuntijan tekemän arvion tukena. Vaikka arviot eivät aina osuneet kohdalleen, taulukko varmisti ainakin sen, että helposti muuten unohtuvat hallinnolliset osuudet ja dokumentointi tulivat huomioitua.

Laskentataulukoiden lisäksi oli käytetty erilaisia muistilistoja ja dokumenttipohjia ja näistä todettiin olevan apua. Esimerkiksi määrämuotoisissa projekti- tai tehtäväsuunnitelmapohjissa on pakko ottaa kantaa eri ohjelmistokehitysvaiheiden työmääriin, ja esimerkiksi laatukatselmoiteihin.

Kaksi haastateltavista mainitsi toimintopistelaskennan, mutta ei ollut itse käyttänyt sitä.

Kaikki haastateltavat mainitsivat kokemuksen yhtenä menetelmänä, tarkoittaen asiantuntijan aikaisempaa kokemusta samasta tai samantyyppisestä järjestelmästä ja tehtävästä.

3.2.3 Miten työmääräarviot tehdään

Haastateltavia pyydettiin kuvailemaan konkreettisesti, miten he itse tekevät työmääräarvion.

Kolme haastateltavista kertoi tekevänsä työmääräarvion yleensä jakamalla tehtävän osiin. Lasketaan esimerkiksi montako aliohjelmia/metodia, montako näyttöä tms. järjestelmän osaa täytyy tehdä tai ylläpitää. Eri osat luokitellaan eri vaikeuskertoimien alle monimutkaisuuden mukaan, eli tutkitaan minkälaisia tietokannan päivitys-, haku- tai selailutoimintoja tai muuten vaativaa logiikkaa niissä on. Tehtävän työmäärä saadaan kertomalla osien lukumäärät vaikeuskertoimilla.

Arviointitaulukkoa apuna käyttävä haastateltava AP tekee pohjatyönä edellä mainitut osuudet, syöttää tiedot taulukon sarakkeisiin, ja taulukko laskee ehdotuksen työmääräarvioksi. Lisäksi hän tekee arvion omalla kokemusperäisellä menetelmällä ja vertaa sitä taulukon antamaan arvioon. Lopullinen arvio on jotakin näiden kahden menetelmän väliltä.

Kaikki haastateltavat pitivät tärkeänä, että työmäärää arvioitaessa tiedetään, tuleeko työn tekemään kokenut henkilö, joka tuntee järjestelmän ja käytettävät ohjelmistokehitysvälineet. Kokemattomalle henkilölle täytyy varata enemmän työaikaa normaalilla

menetelmällä tehdyn arvion lisäksi. Paras tilanne olisi, jos tekijä itse voisi arvioida tulevan työnsä työmäärän, mutta aina tähän ei ole mahdollisuutta.

Kokenein asiantuntija AT ei perusta arvioitaan moduulien kappalemääriin tai vaikeuskertoimiin. Hänen mielestään paras lähtökohta työmääräarvion tekemiselle on, että arvioija tuntee tarkasti arvioitavan työn sisällön ja arvioi tapauskohtaisesti tehtävän vaikutukset järjestelmään. Hän jakaa kuitenkin tehtävät mahdollisuuksien mukaan pienempiin osiin, ja laskee lopuksi osien työmäärät yhteen.

Kaikki haastateltavat käyttäisivät mielellään arvioinnissa apuna aikaisempien samantyyppisten tehtävien toteumia. Niitä on kuitenkin hankala saada selville projektinhallintajärjestelmistä, koska järjestelmät vaihtuvat tiuhaan, eikä niihin ole kaikilla käyttöoikeuksia. Yksi haastateltavista kerää jopa omaa ”kokemustietokantaa” ruutuvihkoon. AP:n entisessä työpaikassa oli kaikki arviot sekä toteumat talletettu niin, että ne löytyivät helposti ja niitä voitiin oikeasti käyttää apuna seuraavissa arvioinneissa. Arviointitaulukkoa käyttävällä yrityksellä on taulukkoon liittyvä tallennuspaikka, jonne voidaan viedä tehdyt arviot, toteumat ja ylitysten tai alitusten syyt. Tietojen avulla on tarkoitus kehittää taulukon makroja.

Haastatellut asiantuntijat tekevät työmääräarviot yleensä ensin yksin, mutta ainakin isompien tehtävien arviot käydään läpi toisen asiantuntijan, koko toteuttavan ryhmän, projektipäällikön tai myyntihenkilön kanssa. Tässä vaiheessa arvioita suurennetaan tai pienennetään, jos uusia näkökohtia tulee esiin. Erään haastateltavan kokemuksen mukaan myyntihenkilöillä on isossa ohjelmistotalossa tapana leikata annettuja arvioita jopa 20%:lla, jotta projekti saadaan myytyä asiakkaalle.

Työmäärien läpikäynti ja tarkastaminen on tarkempaa yrityksissä, jotka tekevät asiakasyrityksille tietojärjestelmiä. Oman organisaation käyttöön tulevan järjestelmän työmääräarvion saattaa tarkastaa ja hyväksyä vain tilaaja, joka on järjestelmän vastuuhenkilö liiketoimintayksikössä.

3.2.4 Pienten ja suurten kokonaisuuksien arviointi

Haastateltavien kanssa keskusteltiin pienten ja isojen kokonaisuuksien työmäärien arvioinnista – onko menetelmissä jotain eroja. Suuri työ tarkoittaa esimerkiksi satojen henkilötöypäivien suuruista kokonaisuutta tai useita liittymäjärjestelmiä koskettavaa muutosta.

Projektipäällikön roolissa toiminut haastateltava kertoi pitävänsä isojen töiden vaatimusmäärittelyistä läpikäyntipalavereja asiakkaan tai liiketoimintayksikön edustajien kanssa. Näin varmistetaan, että asiasisältö on ymmärretty oikein ja saadaan tarvittavia tarkennuksia. Tarvittaessa palaveroidaan myös teknisten asiantuntijoiden ja järjestelmäarkkitehtien kanssa ja mietitään eri toteutusvaihtoehtoja, tarvitaanko tietokantamuutoksia jne. Kukin asiantuntija arvioi oman osa-alueensa työmäärän.

Periaatteessa isojenkin töiden työmäärät arvioidaan samalla tavalla kuin pienempien – yleensä jakamalla tehtävät pieniin kokonaisuuksiin, ja laskemalla summat yhteen. Kaikki haastateltavat olivat kuitenkin yksimielisiä siitä, että tehtävän suuruus itsessään lisää työmäärää, ainakin jos toteuttamassa on useita henkilöitä. Kommunikointiin ja töiden koordinointiin on varattava aikaa. Kaikenlainen byrokratia lisääntyy, kun tehtävän koko kasvaa. Ohjelmien integrointi, liittymäyhteistyö ja järjestelmätestaus on työtyölästä, eikä kaikkia ongelmia voida ennakoita etukäteen. Yllätyksiä tulee väistämättä. Haastateltavilla onkin tapana lisätä hiukan ylimääräistä isompien töiden työmääriin. Eräs haastateltava lisäsi 20% 'suuruuslisää' arvioituun työmäärään. Muut eivät maininneet tarkkoja lukuja, vaan arvioivat tilanteen tapauskohtaisesti. Suurien töiden työmääräarvioita käydään mahdollisuuksien mukaan läpi koko toteuttavan ryhmän kanssa, jotta saadaan mukaan kaikkien osapuolien näkökannat.

Pieniltä vaikuttavien töiden arviointiin ei pidetty tarkoituksenmukaisena käyttää juuriakaan aikaa, ja ne ovatkin yleensä helppoja arvioitavia kohdejärjestelmän asiantuntijoille. Omalle organisaatiolle ohjelmia tekevä haastateltava totesi, että alle viikon tehtäviltä vaikuttavat työt vain tehdään, niistä ei tarvitse laatia arvioita. Asiakasyrityksille töitä tekevät joutuvat kuitenkin arvioimaan etukäteen pienetkin tehtävät.

3.2.5 Työmääräarvioinnin kompastuskivet

Haastateltavia pyydettiin kertomaan, mikä työmäärien arvioinnista tekee hankalaa tai minkälaisissa tapauksissa se on erityisen helppoa.

Haastateltava AT kiteyttää: ”Vaikeaa on se, että osuu kohdalleen.”

Arviointi on vaikeaa, jos ei tunne kunnolla järjestelmää, tehtävän lopullista sisältöä tai tulevan tekijän osaamista. Ohjelmistoyrityksissä asiakkaan yhteistyökyky tai vieraat menetelmät tuovat lisähaasteita arviointiin. Joissakin tapauksessa on vaikea erottaa, mikä on muutos ja mitä olisi pitänyt osata ottaa huomioon alkuperäisessä arvioinnissa. Liittymäyhteistyön vaatiman ajan arviointi on vaikeaa, väärinymmärryksiä voi tulla erilisten kulttuurien ja ammattikielen vuoksi jo tehtävän sisällön ymmärtämisessä.

AM:n mukaan työmäärien arviointi on kuitenkin aina arvaamista ja haasteena on löytää etukäteen mahdolliset ongelmat. Esimerkiksi jos tehdään tarjouksia vieraalle organisaatiolle, jossa ei ole aiemmin ollut vastaavia projekteja, eikä ole ehditty perehtyä asiakkaan vaatimuksiin, tulee vaikeuksia. Asiakkaan menetelmistöön, katselmointi- ja laatuvaatimuksiin sekä työtapoihin pitäisi tutustua etukäteen.

AP:n mukaan erityisen vaikeaa ovat tilanteet, joissa pyydetään ”alustavaa arviota” jostain tehtävästä. Kaikki osapuolet tietävät, että arviointiin tarvittavat tiedot ovat puutteelliset, mutta silti alustavana ja ilman sitoumuksia annetulle arviolle annetaan suuri painoarvo ja sen perusteella tehdään päätöksiä. Alustavaa arviota on jälkikäteen vaikea muuttaa, koska se on voitu jo kirjata moneen paikkaan ja sitä pidetään jo totena alkuperäisistä lupauksista huolimatta. Pitäisi olla rohkeutta olla antamatta arvioita, jos sen tekoon ei kerta kaikkiaan ole riittävästi tietoa, tai ainakin vaatia lisää tietoa arvion tekoa varten.

Arviointi koettiin hankalaksi, jos tehtävä on iso, mukana on vieraita liittymäsovelluksia ja toteuttajien osaamista ei tunneta etukäteen. Kaikkien haastateltavien mielestä työmäärän arviointi on melko helppoa, jos tehtävän määrittelyt ovat täsmälliset, järjestelmä on hyvin dokumentoitu ja arvioija tuntee järjestelmän ja teknisen ympäristön sekä väli-

neet hyvin. Työmäärien arviointi onnistuu, jos pohjatyö eli dokumentteihin tutustuminen tehdään huolella.

3.2.6 Työmäärien arviointiin varattu aika

Haastateltavien mukaan arviointiin on yleensä käytettävissä riittävästi aikaa, jos tehtävä ei ole kovin suuri. Toisaalta arviointiin käytettävä aika ei ole kaikissa yrityksissä laskutettavissa asiakkaalta, joten arviointiin ei ole suotavaa käyttää paljon aikaa. Jos arviointia tehdään tarjouspyyntöä varten, aikataulu voi olla kiireellinen, koska toimeksiantaja on sanellut aikataulun. Joskus saatetaan pyytää pikaisia arvioita ennen jotain tiettyä kokousta, jossa tehtävän toteuttamisesta tehdään päätös.

Arvioinnissa tarvitaan usein järjestelmäarkkitehdin tai liittymien vastuuhenkilöiden apua. Jos näille avainhenkilöille ei ole varattu aikaa arviointiin osallistumiseen, voi olla vaikeaa tuottaa luotettava työmääräarvio pyydettyssä ajassa.

3.2.7 Tiedon määrä ja laatu

Haastateltavilta kysyttiin, voiko arvioitavasta tehtävästä saada liikaa tai turhaa informaatiota.

Kaikki haastateltavat olivat sitä mieltä, että kuuluu ammattitaitoon erottaa tarpeeton tieto oleellisesta. Toimeksiannon laatija ei välttämättä tiedä, mitä tietoa järjestelmäasiantuntija tarvitsee tehtävän toteuttamiseen ja arviointiin. Asiakkaan käyttämä kieli ja ajattelutapa ovat yleensä erilaisia kuin IT-ammattilaisella. Asiantuntijan täytyy osata suodattaa turhat tiedot, ja kysyä tarvittaessa tarkennuksia. Tiedon käsittelyyn kuluu tietysti aikaa, mutta on kuitenkin parempi, että tietoa on liikaa kuin että sitä on liian vähän. Täysin väärä ja harhaanjohtava tieto on kuitenkin vaarallista, joten kysymysten teko on aiheellista, jos joku asia vähänkin epäilyttää. Yleensä tarpeeton tieto on jotain ulkopuoliseen ympäristöön tai liiketoimintaan liittyvää ”nice to know”-tietoa, jota voidaan tarvita jossain muussa yhteydessä, tai se pitää osata välittää eteenpäin sille, joka tietoa tarvitsee.

3.2.8 Työmäärien jakaantuminen ohjelmistokehitysprojektissa

Käytetäänkö työmäärien arvioinnissa jotain kaavaa projektin eri osa-alueiden työmäärille? Miten jakautuvat työmäärät projektinhallintaan, määrittelyyn, suunnitteluun, toteutukseen, testaukseen ja käyttöönottoon?

Kolme haastateltavaa, jotka olivat arvioineet kokonaisia projekteja, käytti tiettyjä prosentteja ainakin projektinhallintaan, testaukseen ja käyttöönottoon. Projektinhallintaan käytetty prosentti oli yleensä 10%, testaukseen 20-25% ja käyttöönottoon 5%. Erään haastateltavan yrityksessä oli käytössä projektien vaihemalli, jossa esitutkimus, määrittelyt, toteutus ja käyttöönotto tehdään omina projekteinaan, joten osuuksien työmäärät arvioidaan toisistaan riippumattomia.

Testauksen työmääriä arvioidaan nykyään usein aivan itsenäisinä, ja testauksen työmäärän osuus projekteissa on kahden haastateltava mukaan hyvin suuri. He kertoivat testauksen työmäärän olevan joissakin tehtävissä jopa kaksinkertainen toteutukseen verrattuna. Yksi haastateltavista kertoi, että testauksen arvio oli tehty laskemalla kaikkien eri tapahtumavariaatioiden lukumäärä ja kerrottu tietyllä työmäärällä - siis aivan samaan tapaan kuin järjestelmien toteutuksen työmäärät lasketaan.

3.2.9 Työmääräarvioiden toteutuminen

Työmääräarvioinnin onnistuminen mitataan sillä, miten hyvin ne toteutuvat. Haastateltavia pyydettiin kertomaan, tietävätkö he miten heidän tekemänsä työmääräarviot ovat toteutuneet ja mistä poikkeamat mahdollisesti johtuvat.

Suurin osa haastateltavien tekemistä arvioista oli pitänyt hyvin paikkansa. Kaikista tehtävistä ei ole saatu palautetta. Työmäärien ylityksiä oli ollut lähinnä isoissa ja kalenteriajassa pitkään kestäneissä töissä. Parhaiten olivat toteutuneet sellaiset arviot, jotka oli tehty tutusta järjestelmästä ja töistä, jotka oli saanut itse toteuttaa.

Eräs haastateltavista kyseenalaisti työmäärien ajantasaisen seurannan toteuttajien tekemiä tuntikirjauksia vasten. Kirjaukset voivat vääristyä, jos toteuttajille kerrotaan, että työmäärä on ylittymässä. Tällöin toteuttaja kirjaakin osan tunteista muille töille tai hal-

lintokoodille. Näin toteumat vääristyvät, ja kun arvioidaan seuraavaa työtä käyttäen toteumia pohjana, virhe kertautuu.

Toinen haastateltava ihmetteli, miksi suurien töiden työmääräarviot ylittyvät, vaikka jokaiselta toteuttajalta itseltään kysytään arvioita ja heidän arvioidensa päälle vielä lisätään tunteja. Arvioivatko kokeneetkin tekijät jatkuvasti työt optimistisesti alakanttiin, vai ylittyykö työmäärä vain siksi, että tekijöitä on monta? Tämä on ratkaisematon mysteeri.

Asiakasprojekteissa työmääriä seurataan tarkemmin ja palautetta annetaan herkemmin. Yhden haastateltavan mukaan seurannan tarkkuus on suoraan suhteessa asiakkaan kiinnostukseen ja palautteeseen. Oman organisaation projekteissa tieto työmäärien paikkansapitävyydestä ei välttämättä kulje toteuttajalle asti. Vain yksi haastateltavista oli ollut yrityksessä, jossa oli käyty toteuttajien kanssa säännöllisesti läpi tehtävien työmäärien toteumat. Kaikki haastateltavat ottaisivat mielellään vastaan enemmän palautetta työmääräarvioidensa toteutumisesta, jotta voisivat parantaa arvioitaan.

3.2.10 Miten työmäärien arviointia voisi oppia?

Kukaan haastateltavista ei muista saaneensa opiskellessaan mitään opetusta työmäärien arvioinnissa. Yksi haastateltavista muistaa saaneensa aiheeseen liittyviä kurssikutsuja työpaikallaan, mutta hän ei ollut osallistunut niihin.

AT oli sitä mieltä, että työmäärien arviointia ei voi opettaa. Se on opittava omien kokemusten kautta. Hankalimpia arviointitehtäviä ei voi antaa aloittelijoille. Jos tekee monta vuotta samanlaisia asioita samaan järjestelmään, jotkut asiat oppii arvioimaan paremmin.

Muut haastateltavat olivat samaa mieltä siitä, että kokemuksen kautta opitaan parhaiten. Mutta heidän mielestään kokeneemmat voisivat kuitenkin opastaa aloittelijoita. Jos jollain on hyväksi koettuja menetelmiä, ne on järkevää opettaa kaikille. RA ehdotti, että esimerkiksi projektipäällikkö voisi olla aluksi toisen päällikön apulaisena oppimassa. Hänen mielestään jonkinlainen harjoitusprojekti voisi olla hyvä idea oppilaitoksiinkin,

jos esimerkit vastaavat todellista elämää ja opettajalla on työkokemusta oikeista tietojärjestelmistä.

AP esitti, että ensimmäinen arviointitehtävä voisi olla helpohko ylläpitotehtävä, jossa apuna olisi kokenut asiantuntija. Hänen kanssaan yhdessä käydään läpi vaatimusmäärittelyt ja tutustutaan järjestelmään tuleviin muutoksiin aivan ohjelmatasolla, eli tehdään puoliksi jo suunnittelua. Näinhän ei yleensä pitäisi työmääräarvioinnissa tehdä, mutta se konkretisoisi kokemattomalle asiantuntijoille tulevan työmäärän. Kun muutokset on löydetty, tehdään niiden luokittelua ja käytetään mahdollisesti apuna arviointitaulukkoa, jos sellainen on käytössä. Kokemattomille arvioijille voisi myös tähdentää, että ei ole tarkoitus tehdä ”kieli vyön alla” arvioita, vaan arvioon täytyy sisällyttää myös ajatus-työhön käytettävä aika – dokumentointia, versionhallintaa yms. rutiineja unohtamatta.

3.2.11 Prosessiajattelu työmäärien arvioinnissa

Haastateltavilta kysyttiin, onko heidän organisaatiossaan käytössä joku prosessi työmäärien arviointia varten, ja jos on, niin noudatetaanko sitä.

Neljä haastateltavista vastasi, että työmäärien arviointia varten ei ole omaa prosessia, tai he ainakaan eivät tunne tai noudata sitä. Työmäärien arviointi voi olla mainittu ohjelmistokehitysprosessissa siten, että sitä edellytetään ennen tehtävän toteutus päätöstä, ja arvioiden hyväksymiselle ja niiden asiakkaalle esittämiselle on määritelty vastuuhenkilö. Arvioinnin tekemisestä ei kuitenkaan ole tarkempaa ohjeistusta.

Yksi haastateltavista oli ollut yrityksessä, jossa otettiin käyttöön toimeksiantomenettely, joka sisälsi työmääräarvioiden teon arviointitaulukkoa apuna käyttäen, arvion läpikäynnin esimiehen kanssa, esimiehen hyväksynnän ja arvion tallettamisen tiettyyn paikkaan ja tietojen täydentämisen myöhemmin toteutuneella työmäärällä. Prosessia noudatettiin hyvin.

3.2.12 Työmääräarvioiden parantaminen

Haastateltavia pyydettiin miettimään, mitä he haluaisivat parantaa organisaationsa työmäärien arviointimenetelmissä.

Useimmilla haastateltavista ei ollut yrityksissään ohjeistusta siitä, miten työmääräarviot pitäisi tehdä. RA toivoi, että ohjeita laadittaisiin, ja että ne olisivat helposti löydettävissä ja noudatettavissa. Projektipäälliköille voi olla olemassa jotain ohjeita, mutta ei tavallisille toteuttajille.

Kaikki haastattelivat toivoivat jotakin helposti käytettävää apuvälinettä. AT, AP ja EN mainitsivat esimerkkinä arviointitaulukon, joka olisi oikeasti toimiva ja hyvin ohjeistettu. Sen käytön opetteluun ja laskentakaavojen parantamiseen pitäisi varata aikaa.

Kolmen haastateltavan mielestä aiempien arvioiden ja toteumien tulisi olla tallessa jossain hyvin dokumentoituina, ja niiden pitäisi olla kaikkien niitä tarvitsevien nähtävillä ja hyödynnettävissä. Työmäärien seurannan pitäisi olla avoimempaa, ja kaikkien toteuttajien pitäisi päästä näkemään koko tehtävän toteutuneet tunnit.

AM:n mielestä asiantuntijan tekemää arviota ei pitäisi pienentää vain siksi, että projekti olisi helpompi myydä asiakkaalle.

Usea haastateltava totesi, että on eniten omasta ammattitaidosta kiinni, että osaa huomioida kaikki työmäärien arvioinnissa huomioon otettavat asiat. Sudenkuoppia voitaisiin kuitenkin mahdollisesti vähentää ohjeistuksella ja välineillä.

3.2.13 Esimerkkejä haastateltavien aiemmin tekemistä arviointitehtävistä

Haastateltavia pyydettiin lopuksi kertomaan esimerkkejä erityisen hankalista tai mukavista arviointitehtävistä.

Mukava arviointitehtävä oli esimerkiksi tietokantakonversioprojekti keskikokoiseen järjestelmään. Tekniikka oli tuttua, ja työn toteuttajat olivat päteviä ja ammattitaitoisia. Tehtävälle ei ollut saneltu valmistumispäivää ylhäältä päin.

EN oli joutunut arvioimaan pieneen, vanhaan järjestelmään tarvittavaa helpolta kuulostavaa parametrimuutosta. Järjestelmä oli ollut käytössä lähes 20 vuotta, eikä siihen oltu

tehty ylläpitoa useisiin vuosiin. Ohjelmat oli toteutettu tuotteella, jota valmistaja ei enää tukenut. Ohjelmadokumenttejakaan ei ollut riittävästi, jotta olisi voinut helposti selvittää, mihin ohjelmiston kohtiin muutos vaikuttaisi. Tiedettiin etukäteen, että muutos jouduttaisiin tekemään yrityksen ja erehdyksen kautta. Toimeksiantaja yllättyi työmäärän suuruudesta, mutta ei voinut muuta kuin hyväksyä tilanteen.

Yksi esimerkki hankalasta arviointitehtävästä oli projekti, jonka alkuperäistä arviota asiakas ei hyväksynyt. Työmäärät oli arvioitava uudelleen pienemmiksi, vaikka merkitsevä toiminnallisuus ei muuttunut millään tavalla. Toteutuksesta piti etsiä vähäpätöisiä osuuksia, jotka karsittiin pois, jotta työmäärää saatettiin pudottaa. Asiakkaalle piti kuitenkin oikeastaan tehdä samat asiat pienemmällä työmäärällä.

RA oli arvioinut tehtävän, jossa vanhan ohjelman toiminnot haluttiin siirtää modernimmalle välineelle. Vaatimusmäärittelyjä ei tehty, vaan asiakas halusi ohjelman, joka toimi samoin kuin vanha ohjelma. Ongelmallisen tästä teki se, että vanhasta ohjelmasta ei ollut dokumentteja ja tiedettiin sen toiminnassa olevan joitakin virheitä. Tehtävään täytyi sisällyttää selvitysvaihe, jossa dokumentoitiin ensin vanhan ohjelman toiminta ohjelmakoodin ja testien perusteella, ja selvitettiin mikä osa toiminnasta oli virheellistä. Selvitysvaiheen työmäärä oli hankala arvioida etukäteen.

Kaikkien haastateltavien mielestä hankala arvioitava on sellainen tehtävä, jossa käytetään vierasta tekniikkaa, johon ei välttämättä saa tukea, tai jossa asia on vaikea. Esimerkiksi jos ei tunne kirjanpitoa ennestään, on EN:n mielestä hankala arvioida kirjanpito-sovelluksen työmääriä, vaikka käytettävä tekniikka olisi tuttua.

4 Tutkimuksen tulokset

Tutkimuksessa pyrittiin selvittämään työmääräarvioinnin nykytilaa ja käytäntöjä suomalaisissa suurissa ja keskisuurissa yrityksissä toteutettavissa tietojärjestelmäprojekteissa. Haluttiin myös löytää ratkaisuja työmääräarvioiden tekemisen helpottamiseen ja niiden laadun parantamiseen. Tutkimusongelmaan etsittiin vastauksia työmäärien arviointimenetelmistä kerätyn teorian, aiempien tutkimusten ja haastattelututkimuksen vastausten perusteella. Haastattelun tulokset olivat linjassa Molokkenin ja Jorgensenin (2003) tekemän tutkimusyhteenvedon kanssa: suomalaisissa yrityksissä käytetään eniten erilaisia asiantuntijoiden tekemiin arvioihin perustuvia menetelmiä työmäärien arvioinnissa, kuten muissakin länsimaissa. Muita menetelmiä, kuten erilaisia algoritmeja, ei juurikaan tunneta eikä käytetä.

4.1 Minkälaisia menetelmiä ja työskentelytapoja työmäärien arvioinnissa käytetään?

Haastattelututkimuksen perusteella tietojärjestelmäprojektien työmääräarviot tehdään asiantuntijan aiempiin kokemuksiin perustuen eli analogiaa käyttäen. Pienistä ja tuttuihin järjestelmiin kohdistuvista töistä asiantuntija osaa arvioida työmäärän suoraan määrittely- ja ohjelmadokumentteja lukemalla. Jotta analogiaa pystytään soveltamaan isosakin tehtävissä, asiantuntijat käyttävät apuna tehtävien jakoa osiin joko omin kriteerein tai erilaisten taulukoiden ja lomakepohjien avulla. Mahdollisuuksien mukaan käytetään apuna myös toteumatietoja aiempien vastaavien tehtävien työmääristä.

Useimmin käytetty analogian apumenetelmä on tehtävien jako pienempiin osiin niiden toiminnallisuuden tai vaikeusasteen perusteella asiantuntijan itse merkittävänä pitämien kriteerien mukaan. Samankaltaisten osien määrä lasketaan yhteen ja kerrotaan yhdelle osalle arvioidulla työmäärällä. Lopuksi lasketaan osasummat yhteen. Vaikka haastateltujen asiantuntijoiden luokittelukriteerit olivat kunkin itse määrittelemiä, niistä löytyi samoja asioita kuin esimerkiksi Cocomo-mallin kustannuskertoimista ja FPA-menetelmän skaalaustekijöistä (Liite 1 taulukot 3. ja 5).

Valmiita luokitteluja ja laskukaavoja sisältävien taulukoiden käyttö arvioinnin apuna oli harvinaisempaa. Kolme haastateltavista oli kokeillut niitä, mutta vain yksi pyrki käyttämään taulukkoa jatkuvasti, vaikka nykyisen yrityksen käytössä oleva taulukko oli osoittautunut puutteelliseksi. Syy taulukon suosimiseen tämän haastateltavan kohdalla oli se, että edellisessä yrityksessä oli ollut toimiva aputaulukko käytössä ja siitä oli hyviä kokemuksia. Hän oli motivoitunut kokeilemaan ja kehittämään uudessakin yrityksessä vastaavantyyppistä arvioinnin aputaulukkoa. Hyvät käytännöt siirtyvät yrityksestä toiseen työntekijöiden mukana.

Lomakepohjia käytetään apuna riippuen arvioijan asemasta projektissa. Haastatteluissa mainitut lomakepohjat ovat lähinnä projektipäälliköiden tai sovellusvastaavan käytössä olevia määrämuotoisia dokumentteja, joilla tehtävä kuvataan ja työmäärä ilmoitetaan tilaajalle; esimerkiksi tehtäväsuunnitelma tai projektisuunnitelma. Lomaketta täyttäessä on pakko ottaa kantaa sellaisten osuuksien työmääriin, joita ei välttämättä muisteta ottaa huomioon arviota tehdessä. Tällaisia osuuksia ovat esimerkiksi työpalaverit, liittymättestaus ja katselmoinnit. Pelkän toteutustyön arvioinnissa edellä mainituista lomakepohjista ei ole hyötyä.

Tarkkoja toteumatietoja aikaisemmista vastaavista tehtävistä ei juurikaan käytetä apuna arvioinnissa. Asiantuntijat haluaisivat käyttää niitä, mutta vain yhdessä yrityksessä ne olivat olleet helposti arvioinnin yhteydessä saatavilla, ja niitä myös kerättiin säännönmukaisesti talteen. Yleensä asiantuntijoilla on käytössä vain omia muistikuvia tai muistutpanoja aikaisempien tehtävien toteutuneista työmääristä.

Asiantuntijat tekevät työmääräarviot yleensä ensin yksin, mutta valmiit arviot käydään läpi vähintään työparin kanssa. Isot tehtävät arvioidaan yleensä niin, että kunkin osa-alueen paras asiantuntija arvioi oman osuutensa, ja arviot käydään läpi koko toteuttavan ryhmän kanssa.

4.2 Minkälaisia ongelmia työmäärien arvioinnissa kohdataan?

Haastattelututkimuksen perusteella asiantuntijoiden mielestä pahin mahdollinen arviointitehtävä olisi vieraaseen ympäristöön, vieraalle asiakkaalle tehtävä, vaikeaa tai täysin

vierasta liiketoimintalogiikkaa sisältävä tietojärjestelmä, jonka määrittelyt olisivat puutteelliset. Näin vaativia tehtäviä ei kuitenkaan käytännössä tule usein.. Sen sijaan pienempiä konkreettisia ongelmia kohdataan arvioita tehtäessä jatkuvasti.

Arviointimalleissa kuten Cocomossa ja UCP-mallissa käytetään yhtenä arvion kokoon vaikuttavana parametrina henkilöstön kokemukseen tai pätevyyteen liittyvää tekijää. Myös kaikki haastateltavat asiantuntijat pitivät tärkeänä, että he tietävät arviota tehdessään, kuka työn tulee tekemään. Tämä tieto ei kuitenkaan aina ole käytettävissä, ja tällöin arvion tekeminen on vaikeaa.

Isojen kokonaisuuksien arviointi oli haastateltavien mielestä aina vaikeaa. Arviot ylittyvät usein, eikä selkeää syytä ylityksiin aina löydetä. Vaikka isojen töiden arvioihin lisätään ”koordinointilisää” jopa 20%, silti yllätyksiä tulee. Isoissa kokonaisuuksissa on useimmiten mukana liittymäsovelluksia, joiden kanssa esimerkiksi rajapinnoista ja testaamisesta sopiminen tuo lisähaasteita projektiin.

Arvioinnissa tarvitaan usein muiden asiantuntijoiden apua. Esimerkiksi tietokantaan tai arkkitehtuuriin liittyvistä ratkaisuksista ei voida tehdä työmääräarvioita ja päätöksiä ilman alan asiantuntijan apua. Nämä henkilöt ovat kuitenkin usein kiireisiä, eikä heillä aina ole varahenkilöitä esimerkiksi lomien aikana. Arvioiden teko voi viivästyä toisen asiantuntijan lausuntoa odotellessa, tai arvio täytyy antaa annetun aikataulun puitteissa omien puutteellisten tietojen perusteella.

Asiantuntijan on helppo tehdä työmääräarvio, jos hänellä on kokemusta aikaisemmista samantyyppisistä tehtävistä ja niihin käytetyistä työmääristä. Haastateltavat asiantuntijat eivät kuitenkaan saaneet säännöllisesti palautetta tekemiensä työmääräarvioiden toteutumisesta. Tietoja työmäärien toteutumisesta ei myöskään talletettu yhdenkään haastateltavan nykyisessä organisaatiossa sellaiseen paikkaan, josta ne olisivat helposti löydettävissä seuraavia arvioita tehtäessä.

Haastattelututkimuksessa selvisi, että yrityksissä ei ole selkeitä toimintatapoja ja ohjeita siitä, miten työmääräarviot tulisi tehdä. Ohjelmistokehitysprosessissa voi olla maininta siitä, että työmäärät on arvioitava, ja että esimiehen on tarkistettava tai hyväksyttävä

arvio. Työmäärän tekotapa on kuitenkin täysin asiantuntijan oman ammattitaidon ja viitsimisen varassa. Puhdas arvaus tuntuu olevan yrityksissä yhtä hyväksyttävä tapa arvioida työmääriä kuin huolellinen määritysten ja ohjelmakuvausten läpikäynti ja tehtävien osittaminen.

4.3 Ehdotuksia työmääräarvioinnin helpottamiseksi ja arvioiden laadun parantamiseksi

Tutkimuksessa haluttiin selvittää, mitä voitaisiin tehdä, jotta työmääräarvioinnin teko olisi asiantuntijoiden mielestä helpompaa ja arviot pitäisivät paremmin paikkansa.

Aiemmista tutkimuksista tehdyssä yhteenvedossa Jorgensen (2007) on todennut, että asiantuntija-arvioon perustuva työmääräarvio on keskimäärin parempi kuin malleihin perustuva. Tämä johtuu siitä, että mallien parametrit eivät sovellu ympäristöön, jossa niitä käytetään, tai siitä ettei malleja osata käyttää oikein. Toisaalta Jorgensen ja Grimstad (2006) ovat havainneet tutkimuksessaan, että asiantuntijoiden arviot kärsivät epätarkasta ja turhasta tiedosta, ja että etukäteen tiedossa olevat odotukset projektin valmistumispäivästä voivat vaikuttaa asiantuntijan tekemään arvioon.

Haastatteluista nousi esiin kehitysehdotuksia, jotka olisivat helposti toteutettavissa. Haastateltavat toivoivat esimerkiksi selkeitä ohjeita ja menetelmiä työmääräarvioiden tekoon.

Työmäärien arviointiin pitäisi luoda prosessi, jonka lähtökohtana ovat tietojärjestelmän määritykset ja välituloksena työmääräarvio – prosessi päättyy vasta siihen, kun projektin toteutuneet työmäärät talletetaan kokemustietokantaan ja arvioija saa palautteen arvion toteutumisesta. Prosessissa kuvataan vaiheet ja menetelmät, jotka läpikäymällä syntyy tarvittava dokumentaatio ja päästään lopputulokseen. Menetelmät valitaan kullekin yritykselle sopiviksi. Asiantuntijat itse toivoivat jotain helposti opittavaa ja käytettävää apuvälinettä, joka tukisi kokemukseen perustuvien arvioiden tekemisessä. Jonkinlainen tietojärjestelmäkohtaisesti räätälöitävä taulukko voisi sopia moneen projektiin.

Hyvin tärkeää olisi luoda kokemustietokanta, jonne toteutuneet työmäärät talletetaan. Tietokannasta pitäisi olla helposti löydettävissä tehtävän kuvaus, siihen arvioitu työ-

määrä, sekä lopullinen toteutunut työmäärä. Näitä lukuja voitaisiin käyttää hyväksi asiantuntijan itse tekemässä arviossa, sekä apuvälineenä käytettävän taulukon laskukaavojen muokkaamisessa. Tällä hetkellä haastateltujen asiantuntijoiden tekemät arviot perustuivat vain hatarasti aiempiin toteumiin. Suoraa palautetta arvioiden toteutumisesta saatiin vain harvoin. Toteumatiedot ovat yleensä tallessa projektinhallintajärjestelmissä, mutta siellä tiedot olivat käytännön arviointityötä tekevien henkilöiden saavuttamattomissa, tai niitä ei enää löydetä tai osata tulkita esimerkiksi ohjelmistoversion tai projektipäällikön vaihduttua.

Työmääräarviointi opitaan haastateltavien mukaan pikkuhiljaa kokemuksen kasvaessa. Kuitenkin arviointia olisi syytä opettaa aloitteleville järjestelmäsuunnittelijoille siten, että arviot tehdään aluksi kokeneemman henkilön kanssa. Opettamisessa auttaisi, jos yrityksessä olisi käytössä selkeä prosessi ja menetelmät työmäärien arviointia varten. Työmäärien arviointia voitaisiin myös harjoitella todellisten toteutuneiden projektien tiedoilla, kuten Jorgensen (2004) ehdottaa. Arvioita verrattaisiin toteumiin, ja arvioinnissa tehdyt virheet analysoidaisiin.

Haastateltavien joukossa oli henkilö, joka oli edellisessä yrityksessä tottunut käyttämään arvioinnin aputaulukkoa. Samasta yrityksestä oli siirtynyt nykyiseen yritykseen useita työntekijöitä, ja heidän aloitteestaan samankaltaista aputaulukkoa yritettiin ottaa käyttöön myös uudessa ympäristössä. Vaikka taulukon räätälöinti uuteen organisaatioon ja erilaisiin järjestelmiin vaatii paljon työtä ja uusien käytäntöjen opettelua vanhoille työntekijöille, tämä on hyvä esimerkki siitä, miten hyviksi todettuja käytäntöjä siirtyy työntekijöiden mukana organisaatiosta toiseen. Asiantuntijoiden kannattaisikin olla avoimia uusille ideoille ja yrittää ottaa selvää, onko muualla hankalat asiat hoidettu paremmin – kuitenkin liikesalaisuuksia paljastamatta.

Projektin johto voi helpottaa työmääräarvioiden tekoa varaamalla työmäärien arviointiin riittävästi aikaa kaikille tarvittaville asiantuntijoille. Haastatteluista kävi ilmi, että joskus teknisten asiantuntijoiden, kuten järjestelmäarkkitehtien tai liittymäsovellusten asiantuntijoiden lausuntoja ei ehditä saada ajoissa arvioiden tekoa varten.

Projektin johdon rooli on tärkeä myös siinä, missä vaiheessa asiantuntijoilta pyydetään työmääräarvioita. Tarkan arvion teko puutteellisilla tiedoilla on mahdotonta. Chandra-sekaran ja Venkatesh Kumar (2012) esittävät, että merkittävin syy työmääräarvioiden ylityksiin on arvioinnissa käytettävien parametrien epätarkkuus ja tiedon puute. Kuten yksi haastateltavista totesi, pitäisi olla rohkeutta olla tekemättä arviota, jos tiedot ovat selkeästi puutteelliset. Pitäisi arvioida vain ne osuudet projektista, jotka varmasti tunnetaan. Sommervillen (2004, 624) mukaan työmääräarvioiden tarkkuus riippuu käytettävissä olevasta informaatiosta ja mitä myöhemmin arvion voi tehdä, sitä tarkempi siitä tulee. Arvioita pitäisi voida myös olla mahdollista tarkentaa, kun ollaan pitemmällä projektissa. Ketterien menetelmien projekteissa tämä onkin jo otettu paremmin huomioon.

5 Tulosten ja oppimisprosessin arviointi

Tutkimus toi mielestäni hyviä vastauksia määriteltyihin tutkimuskysymyksiin. Valitut tutkimusmenetelmät tukivat työtä ja auttoivat määrätietoisessa etenemisessä kohti tutkimuksen lopputuloksia.

5.1 Tulosten pätevyys ja luotettavuus

Tutkimuksen tuloksia voidaan soveltaa koskemaan keskisuuria ja suuria tietojärjestelmiä toimittavia suomalaisissa yrityksissä ja niissä työskenteleviä asiantuntijoita.

Tutkijan asuinpaikasta ja omista verkostoista johtuen haastateltavien asiantuntijoiden yritykset sijaitsevat pääkaupunkiseudulla. Sattumalta myös haastateltavat ovat aikoinaan opiskelleet pääkaupunkiseudun oppilaitoksissa. On mahdollista, että muilla paikkakunnilla sijaitsevissa yrityksissä työskenteleviä tai muissa oppilaitoksissa opiskelleita asiantuntijoita haastatteleamalla olisi saatu erilaisia tutkimustuloksia. Yritysten toimialat olivat perinteisiä esimerkiksi pankki-, vakuutus-, julkishallinnon aloja. Mukana ei ollut uusmedia- tai pelialan yrityksiä, jotka usein ovat pieniä organisaatioita. Tutkimustulokset eivät päde tällaisiin yrityksiin, eikä tutkimuksen kehittämisehdotuksia voida soveltaa niihin.

Haastateltavia asiantuntijoita oli viisi, ja heillä kaikilla oli kokemusta työskentelystä useissa eri yrityksissä. Näin ollen haastattelut antoivat tietoa useammassa kuin viidessä yrityksessä käytettävistä arviointimenetelmistä. Kanasen (2008, 34) mukaan tutkimusaineistoa on riittävästi, kun uudet tapaukset eivät enää muuta tulkintaa. Asiantuntijoiden vastaukset useimpiin kysymyksiin olivat hyvin samanlaisia. Haastateltavien määrä oli siis riittävä laadullisen tutkimuksen kannalta. On kuitenkin mahdollista, että haastateltaviksi valikoituneet asiantuntijat poikkeavat edustamansa ryhmän valtavirrasta, eivätkä haastattelututkimuksen tulokset näin ollen antaisi oikeaa kuvaa tietojärjestelmien työmääräarvioinnin nykytilasta Suomessa.

5.2 Oppimisen ja opinnäytetyöprosessin arviointi

Opinnäytetyöprojekti oli suunniteltu 17 viikon pituiseksi. Aikataulu oli sopiva, eikä ongelmia ollut. Oli hyvä, että tarkistuspisteitä oli sovittu muutaman viikon välein, joten tuloksia oli pakko saada aikaan ja työ eteni hyvin. Projektimuotoinen työskentely ja töiden aikataulut ja raportointi oli minulle tuttua entuudestaan, joten siltä osin en oppinut uutta. Sen sijaan tieteellisen tutkimuksen teoriat ja käytännöt olivat vieraita, ja tutkimusraportin kirjoitustyyliä täytyi opetella lähdeoteoksista ja aiemmin julkaistuista opinnäytetöistä.

Valitsin tutkimuksen aiheeksi työmäärien arvioinnin, koska se on tärkeä ja kiinnostava osa omaa työtäni, ja halusin oppia siitä lisää. Opinkin paljon uutta tutkiessani erilaisia arviointimenetelmiä ja niiden käytöstä tehtyjä tutkimuksia sekä haastatellessani asiantuntijoita. Näitä tietoja voin suoraan soveltaa työssäni.

6 Yhteenveto

Työmäärien arviointi on vaikeaa, mutta välttämätöntä. Tietojärjestelmien asiantuntijat osaavat arvioida hyvin tuttuihin järjestelmiin tulevia muutoksia tai tutulla tekniikalla tehtäviä uusia järjestelmiä. Tehtävän suuruus, liiketoiminnan vaikeus tai tuntematon tekniikka tekevät arvioinnista vaikeaa. Työmääräarvioiden tekemisessä tärkeintä on asiantuntijan kokemus ja pätevyys. Arvioinnin avuksi kehitettyjä malleja ja algoritmeja tunnetaan huonosti, eikä niitä käytetä. Asiantuntijat kuitenkin soveltavat samantapaisia menetelmiä, jotka he itse ovat kehittäneet, tai joista he ovat kuulleet kollegoiltaan. Muistilistat, dokumenttipohjat, tehtävien ositus ja lajittelu, painokertoimet ja aputaulukot ovat vaihtelevasti käytössä eri yrityksissä, ja ne koetaan hyödyllisiksi. Asiantuntijoita vaivaa kuitenkin epävarmuus omien menetelmien riittävydestä, koska he eivät saa säännöllistä palautetta siitä, kuinka arviot ovat toteutuneet.

Työmääräarvioiden tekoa yrityksissä voitaisiin helpottaa suunnittelemalla selkeät käytännöt siihen, miten arviot kannattaa tehdä; eli luotaisiin työmääräarvioinnille prosessi. Prosessissa kuvattaisiin työmääräarvioinnin vaiheet alkaen siitä, mitä pohjatietoja asiantuntijan on saatava arviointia varten ja päättyen siihen, että asiantuntija saa palautetta alkuperäisen arvion toteutumisesta. Asiantuntijoiden tällä hetkellä käyttämät parhaat apuvälineet - esimerkiksi arviointitaulukot - opetettaisiin kaikille, ja välineitä yritettäisiin parantaa kokemuksen karttuessa. Tehdyt alkuperäiset arviot ja niiden toteumat talletettaisiin kokemustietokantaan, josta tiedot löytyisivät helposti, kun pitäisi arvioida uudelleen samankaltaisia tehtäviä. Uusille työntekijöille olisi helppo opettaa prosessin mukainen toiminta, ja he voisivat harjoitella työmäärien arviointia kokemustietokannasta löytyvien projektitietojen avulla.

Sataprosenttiseen luotettavuuteen tietojärjestelmäprojektien työmääräarvioinnissa tuskin koskaan päästään, koska tietojärjestelmien vaatimusmäärittelyt ovat parhaimmillaankin puutteellisia, ja projektin eri vaiheissa voi tulla yllätyksiä, joita kukaan ei ole osannut ennakoida. Riittävä tarkkuus voidaan kuitenkin saavuttaa, kun arviointityö tehdään huolellisesti hyviksi havaittuja menetelmiä käyttäen ja opitaan aiemmista kokemuksista.

Lähteet

Chandrasekaran, R. & Venkatesh Kumar R. 2012. On the Estimation of the Software Effort and Schedule using Constructive Cost Model – II and Functional Point Analysis. International Journal of Computer Applications (0975-8887). Volume 44 No 9. April 2012. Luettavissa: <http://research.ijcaonline.org/volume44/number9/pxc3878492.pdf>
Luettu: 26.5.2012.

Cohn, M. 2009. Agile Estimating and Planning. Robert C. Martin Series. 9th printing. Pearson Education. Upper Saddle River, NJ.

Fetcke, T., Abran, A. & Nguyen, T. 1997. Mapping the OO-Jacobson Approach into Function Point Analysis. IEEE. Piscataway, NJ.
Luettavissa: <http://www.cs.unibo.it/~cianca/wwwpages/ids/lettere/Fetcke.pdf>
Luettu: 24.5.2012.

Fetcke, T. 1999. A Generalized Structure for Function Point Analysis. International Workshop on Software Measurement (IWSM'99) – September 8-10, 1999 Montreal, Canada. Luettavissa:
<http://s3.amazonaws.com/publicationslist.org/data/gelog/ref-421/450.pdf>. Luettu: 9.6.2012.

Haikala, I. & Märijärvi, J. 2004. Ohjelmistotuotanto. Valikko-sarja. 10. painos. Talentum. Helsinki.

Jorgensen, M. 2004. A Review of Studies on Expert Estimation of Software Development Effort. Journal of Systems and Software, 2004 Luettavissa:
http://simula.no/research/se/publications/SE.4.Joergensen.2004.c/simula_pdf_file
Luettu: 26.5.2012.

Jorgensen, M. 2007. Forecasting of Software Development Work Effort: Evidence on Expert Judgment and Formal Models. International Journal of Forecasting 23(3):449-

462. Luettavissa: <http://simula.no/research/se/publications/Jorgensen.2007.2>. Luettu: 26.5.2012.

Jorgensen, M. & Grimstad, S. 2006. How to avoid impact from irrelevant and misleading information on your cost estimates. Simula Research Laboratory. Lysaker, Norway. Luettavissa: <http://www.simula.no/research/prepare/projects/bestproject/seminars/estimation06/jorgensen/view> . Luettu: 31.5.2012.

Hirsjärvi, S., Remes, P. & Sajavaara, P. 2008. Tutki ja kirjoita. Tammi. Helsinki.

Lalcrest limited. Function Point Workbench. Verkkosivusto. Luettavissa: <http://www.lalcrest.co.uk/workbench.php>. Luettu: 9.6.2012.

Kananen, J. 2008. Kvali – Kvalitatiivisen tutkimuksen teoria ja käytänteet. Jyväskylä: Jyväskylän Ammattikorkeakoulun julkaisuja 93.

Meli, R. & Santillo, L. 1999. Function point estimation methods: A comparative overview. Data Proseccing Organization. Luettavissa: <http://www.dpo.it/resources/papers/1999-fesma-fpestmet-en.pdf>. Luettu: 9.6.2012.

Molokken, K & Jorgensen, M. 2003. A review of surveys on software effort estimation. International Symposium on Empirical Software Engineering (ISESE 2003). Rome, Italy. IEEE Computer Society. Luettavissa: http://folk.uio.no/isu/INCO/Papers/Review_final8.pdf. Luettu: 26.5.2012.

Pelin, R. 2011. Projektihallinnan käsikirja. 7. uudistettu painos. Projektijohtaminen Oy Risto Pelin, Helsinki.

Savolahti, M. 2002. Ohjelmiston toiminnallisen koon laskenta. Pro gradu –tutkielma Joensuun yliopisto. Joensuu. Luettavissa: ftp://cs.joensuu.fi/pub/Theses/2002_MSc_Savolahti_Markku.pdf Luettu: 9.6.2012.

Schneider, G & Winters, J. 2000. Applying use cases, a practical guide. 6th printing. Addison Wesley. Reading, Massachusetts.

Sommerville, I. 2004. Software Engineering 7. Pearson Education Limited. Harlow, UK.

Thompson, K. 2003. Using Function Point Analysis for an Object Oriented Methodology. Tehcnology Transfer. A monthly article from our Speakers. October 2003. Roma, Italy. Luettavissa:

http://www.technologytransfer.eu/article/29/2003/12/Using_Function_Point_Analysis_for_an_Object_Oriented_Methodology.html. Luettu: 29.5.2012.

Liitteet

Liite 1 Taulukot 2, 3 ja 5.

Liite 2 Haastattelukysymykset

Liite 1.

Taulukko 2. COCOMO-mallin skaalaustekijät

Tekijä	Kuvaus	VL	L	N	H	VH	EH
PREC	Samantlaisia on tehty aikaisemminkin	6,20	4,96	3,72	2,48	1,24	0,00
FLEX	Vaatimuksia voidaan muuttaa	5,07	4,05	3,04	2,03	1,01	0,00
RESL	Riskit tunnetaan ja hallitaan	7,07	5,65	4,24	2,83	1,41	0,00
TEAM	Projektihenkilöstön yhteistyökyky	5,48	4,38	3,29	2,19	1,10	0,00
PMAT	Tuotantoprosessin kypsyys	7,80	6,24	4,68	3,12	1,56	0,00

Arviointiasteikko: VL=hyvin alhainen, L= alhainen, N=normaali, H=korkea, VH=hyvin korkea ja EH=erittäin korkea.

(Haikala & Märijärvi 2004, 247)

Taulukko 3. COCOMO-mallin kustannuskertoimet

Tekijä	Kuvaus	VL	L	N	H	VH	EH
Tuote							
RELY	Luotettavuusvaatimukset	0,82	0,92	1,00	1,10	1,26	
DATA	Tietokannan koko		0,90	1,00	1,14	1,28	
DOCU	Dokumentointivaatimukset	0,81	0,91	1,00	1,11	1,23	
CPLX	Tuotteen monimutkaisuus	0,73	0,87	1,00	1,17	1,34	1,74
RUSE	Uudelleenkäytettävyysvaatimukset		0,95	1,00	1,07	1,15	1,24
Toteutusalue							
TIME	Suoritusaikavaatimukset			1,00	1,11	1,29	1,63
STOR	Muistitarajoitukset			1,00	1,05	1,17	1,46
PVOL	Kehitysjärjestelmän epäkypsyys		0,87	1,00	1,15	1,30	
Henkilöstö							
ACAP	Suunnittelijoiden kyvykkyys	1,42	1,19	1,00	0,85	0,71	
APEX	Sovellusalueen tuntemus	1,22	1,10	1,00	0,88	0,81	
PCAP	Ohjelmoijien kyvykkyys	1,34	1,15	1,00	0,88	0,76	
PLEX	Kehitysjärjestelmän tuntemus	1,19	1,09	1,00	0,91	0,85	
LTEX	Ohjelmointikielen tuntemus	1,20	1,09	1,00	0,90	0,84	
PCON	Henkilökunnan pysyvyys	1,29	1,12	1,00	0,90	0,81	
Projekti							
TOOL	Työkalujen käyttö	1,17	1,09	1,00	0,90	0,78	
SITE	Tuotekehityksen hajautus	1,22	1,09	1,00	0,93	0,86	0,80
SCED	Aikataulu:VL=25% kiristys , L=15% kiristys	1,43	1,14	1,00	1,00	1,00	

Arviointiasteikko: VL=hyvin alhainen, L= alhainen, N=normaali, H=korkea, VH=hyvin korkea ja EH=erittäin korkea.

(Haikala & Märijärvi 2004, 247)

Taulukko 5. Factor of Technical Complexity

Factor	
F1	Data communications
F2	Distributed data processing
F3	Performance
F4	Heavily used configuration
F5	Transaction rate
F6	On-line data entry
F7	End-user efficiency
F8	On-line update
F9	Reusability
F10	Complex processing
F11	Installation ease
F12	Operational ease
F13	Multiple sites
F14	Facilitate change

(Chandrasekaran & Venkatesh Kumar 2012)

Liite 2 Haastattelukysymykset

- Tausta, kokemus, nykyinen tehtävä.
- Minkälaisia tehtäviä olet arvioinut? (Määrittely, ohjelmointi, testaus, käyttöönotto, ylläpito, projektin johto, isompi kokonaisuus?)
- Mitä työmäärien arviointimenetelmiä tunnet?
- Miten itse teet työmääräarvion?
- Teetkö arviot yksin vai ryhmässä?
- Onko organisaatiossi käytössä joku prosessi työmäärien arviointia varten? Jos on, niin noudatetaanko sitä?
- Miten arvioit isoja kokonaisuuksia?
- Entä pienempiä?
- Miten pienten ja suurten kokonaisuuksien arviointi eroaa toisistaan?
- Mikä arvioinnissa on vaikeaa / helppoa?
- Mikä auttaa tekemään työmääräarvion?
- Mikä hankaloittaa työmääräarvion tekemistä?
- Onko työmäärien arviointiin varattu riittävästi aikaa?
- Voiko arvioitavasta tehtävästä saada liikaa/turhaa informaatiota?
- Miten työmäärät jakaantuvat ohjelmistokehitysprojektissa? (Määrittely, suunnittelu, toteutus, yks & integrointitestaus, järjestelmättestaus, hyväksymistestaus, käyttöönotto, projektinhallinta....) Käytätkö tähän jotain kaavaa?
- Miten työmääräarviosi ovat toteutuneet? Saatko palautetta?
- Miten työmäärien arviointia voisi opettaa / oppia?
- Kuka tarkastaa / hyväksyy arviot?
- Miten arvioit projektin- tai muutostenhallintaan tarvittavan ajan?
- Mitä haluaisit parantaa organisaatiosi työmääräarviointiprosessissa? Entä omissa menetelmissäsi?
- Kerro joku esimerkki hankalasta / mukavasta työmäärien arviointitehtävästä.