Tero Tilus

MODEST: Menetelmä tietojärjestelmään kohdistuvien muutospaineiden arviointiin

Tietotekniikan (Ohjelmistotekniikka)
viidentoista (15) opintoviikon
laajuinen sivuaineen
laudatur-tutkielma

28. huhtikuuta 2006

Jyväskylän yliopisto

Tietotekniikan laitos

Jyväskylä

Tekijä: Tero Tilus
Yhteystiedot: tero.tilus@iki.fi
Työn nimi: MODEST: Menetelmä tietojärjestelmään kohdistuvien muutospaineiden arviointiin
Title in English: MODEST: A Method for Early System Modernization Pressure Estimation
Työ: Tietotekniikan (Ohjelmistotekniikka) viidentoista (15) opintoviikon laajuinen sivuaineen laudatur-tutkielma
Sivumäärä: 66
Tiivistelmä: Uudistamispäätökset ovat taloudellisesti merkittäviä. Siksi päätöksen valmisteluun ja tilanteen kartoittamiseen on varattava riittävästi aikaa. On osoittautunut, että nykyinen menetelmällinen tuki ei riittävän hyvin mahdollista uudistamistarpeen varhaista havainnointia. Tässä tutkielmassa esitellään MODEST (Modernization Pressure Estimation), menetelmä uudistamistarpeen varhaiseen havainnointiin. Se pohjautuu uudistamisen päätöskriteereistä ja päätöksentekoprosessista koottuun empiiriseen aineistoon. Aineisto on koottu ohjelmistoteollisuudesta. Menetelmä, sekä arviointi- että kalibrointiprosessi, kuvataan yksityiskohtaisesti. Kuvaus sisältää menetelmän empiirisen perustan, validoinnin, yhteismitallisuusongelmien käsittelyn ja käytön ohjeistuksen. Menetelmän käytön tukimateriaalia ei kuitenkaan oheisteta. MODEST on soveltuvuusvalidoitu asiantuntija-katselmoinnein ja pilotoinnein. Menetemän voidaan edellisten perusteella sanoa olevan tarkoitustaan varten riittävän kevyt, helppokäyttöinen ja toimiva.
English abstract: Decisions regarding software modernizations are economically important. Therefore a sufficent amount of time is needed to properly assess the decision and plan the action. It has appeared that currently available methods do not properly support early detection of modernization needs. In this thesis a method for early system modernization needs detection, MODEST (Modernization Pressure Estimation), is presented. It is based on empirical data of modernization decision factors and decision making process gathered from software industry. A detailed description of the method is given. Material supporting the method use is not included. Description includes empirical foundation and validation of the method and activities of legacy system evaluation process and self-calibration process. Commensurability aspects are also discussed. Applicability of MODEST has been validated by expert inspections and industrial use cases. Based on the experiences it can be concluded that it is low-effort, easy to use and potentially effective method for its purpose.
Avainsanat: ohjelmisto, uudistamistarve, arviointimenetelmä
Keywords: software, modernization need, estimation method

Esipuhe

Tässä opinnäytetyössäni esittelen menetelmän, jonka kehitystyö on tapahtunut Jyväskylän yliopiston Tietotekniikan tutkimusinstituutin (TITU) toteuttaman teollisuusyhteistyöprojektin puitteissa. Projektin mahdollistivat sen rahoittajat, TEKES, IBS Oy, TietoEnator Oyj ja Tietokarhu Oy, sekä hankkeen valmistelussa merkittävän työn tehneet Jarmo Ahonen (tällä hetkellä ohjelmistotekniikan professori Kuopion yliopistossa) ja Jussi Koskinen (tällä hetkellä ohjelmistotekniikan yliassistentti Jyväskylän yliopistossa).

Opinnäytetyötäni ohjasivat Jussi Koskinen ja Jarmo Ahonen. Menetelmän suunnittelun keskeisissä päätöksissä suurena apuna on ollut Jarmo Ahosen kokemus ja näkemykset aihealueelta. Lisäksi kehitystyössä ja erityisesti menetelmän testauksessa ovat auttaneet tutkijakollegat TITUlla, Henna Sivula, Heikki Lintinen, Irja Kankaanpää ja Päivi Juutilainen.

Sisältö

1 Johdanto

Tietojärjestelmän elinkaari voidaan jakaa kehitys- ja ylläpitovaiheeseen. Ylläpitovaiheeseen siirrytään kun tietojärjestelmä otetaan käyttöön [23]. Tässä opinnäytetyössä käytän rinnasteisesti termejä "ylläpito" ja "evoluutio" viittaamaan kaikkien tietojärjestelmään sen käyttöönoton jälkeen kohdistuvien toimenpiteiden muodostamaan kehitysprosessiin.

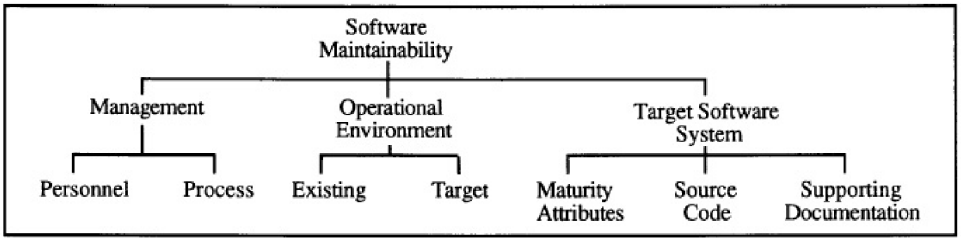
Ylläpitovaiheen suhteellinen osuus on menestyksekkäiden tietojärjestelmien tapauksessa yleisesti ollut 60–75 % elinkaarikustannuksista. Muutamia keskeisimpiä elinkaarikustannusten tutkimustuloksia on vedetty yhteen taulukossa 1.1. Suhteellinen osuus on paitsi suuri, myös kasvamaan päin [45]. Ylläpito oheistoimintoineen on sekä työ-, että kalenteriaikana mitaten elinkaaren vaiheista selkeästi merkittävin. Näin ollen ylläpitoprosessien kypsyys, siihen liittyvä organisaation kyvykkyys ja erityisesti ylläpidon päätöksenteko ovat tietojärjestelmän elinkaarikustannuksia tarkasteltaessa merkittäviä asioita.

Vuosi	**Osuus**	**Ylläpidon osuus...**	**Lähde**
2000	>90 %	ohjelmistokustannuksista	Erlikh [9]
1993	75 %	tietojärjestelmäbudjetista	Eastwood [7]
1990	>90 %	ohjelmistobudjetista	Moad [33]
1990	60-70 %	tietohallintobudjetista	Huff [15]
1988	60-70 %	tietohallintobudjetista	Port [38]
1984	65-75 %	ohjelmistokehityksen työmäärästä	McKee [30]
1981	>50 %	ohjelmistotalon kokonaistyömäärästä	Lientz & Swanson [29]
1979	67 %	ohjelmistokustannuksista	Zelkowitz *ym.* [55]

Taulukko 1.1: Tutkimustuloksia ylläpidon osuudesta tietojärjestelmän elinkaaressa

Ylläpitovaiheen suuren suhteellisen osuuden johdosta ohjelmiston ylläpidettävyys on elinkaaren kokonaisuuden kannalta kriittinen tekijä. Ohjelmiston ylläpidettävyyttä on karakterisoitu useilla eri tavoilla. Oman ja Hagemeister [35] ovat esittäneet ylläpidettävyydelle hierarkkisen jaottelun, jossa ylläpidettävyys puretaan hierarkkisesti luetteloksi tekijöitä. Näitä tekijöitä heidän hierarkiassaan on 92. Hierarkian kolme ylintä tasoa on esitetty kuvassa 1.1. Sen lisäksi Oman ja Hagemeister ovat esittäneet myös kyseisen jaottelun mukaiset metriikat ohjelmiston ylläpidettävyyden mittaamiseen [34].

Ajan kuluessa ohjelmistolle asetetut vaatimukset lähes väistämättä muuttuvat. Se ai-



Kuva 1.1: Ylläpidettävyyden hierarkia Omanin ja Hagemeisterin mukaan (lähde [34])

heuttaa painetta myös ohjelmiston muuttamiseen, jotta muuttuneisiin vaatimuksiin voitaisiin vastata. Tämä jatkuvan muutoksen tarve tunnetaan Lehmanin ensimmäisenä lakina [28]. Järjestelmä, jonka ylläpidettävyys on hyvä, voi kestää hyvinkin laajoja muutoksia, mikäli ne hallitaan hyvin. Järjestelmän sisäinen rakenne yleensä ennen pitkää kärsii muutoksista, ellei laadun ylläpitämiseen erityisesti kiinnitetä huomiota. Tätä laadun heikkenemisen prosessia kutsutaan rämettymiseksi tai rapautumiseksi ja se tunnetaan myös Lehmanin seitsemäntenä lakina [27]. Lisäksi ylläpidettävällä ohjelmalla on taipumus muutosten myötä kasvaa ja monimutkaistua. Tämä kasvutaipumus tunnetaan Lehmanin toisena lakina.

Mutkikkuus vaikuttaa voimakkaasti ylläpitokustannuksiin [2]. Muutosten määrä ja niiden laajuus tulisi siksi säilyttää maltillisella tasolla. Toisaalta Lehmanin ensimmäinen laki huomioiden muutosten määrän tulee kuitenkin olla riittävä. Uudistamispäätöksistä vastaavat asiantuntijat ovat arvottaneet sekä mutkikkuuden, että kustannukset tärkeimpien päätöskriteerien joukkoon [24]. Tarvittavien muutosten laajuuteen voidaan puuttua tunnistamalla varhaisessa vaiheessa reagointia edellyttävät muutospaineet ja niiden kohdentuminen. Varhainen puuttuminen luonnollisesti edellyttää järjestelmällistä seurantaa. Kuinka tämä seuranta järjestetään siten, että sen toteuttaminen on kannattavaa, eli että aikaistuneesta muutospaineisiin reagoinnista on riittävästi etua? Reunaehtoja vastaukselle tähän kysymyksen hahmotellaan menetelmälle asetettavien vaatimusten muodossa luvussa 3.2 (s. 15).

Päämääränä ohjelmistokehityksessä tulisi, erityisesti asiakasorganisaation näkökulmasta katsottuna, Zvegintzovin [56] mukaan olla pitkäikäinen ohjelmistoinvestointi. Pitkäikäisellä investoinnilla on merkittävästi lyhyempää paremmat mahdollisuudet olla tuottava. Pitkän käyttöiän aikana saadaan suhteessa enemmän etua siitä ohjelmistojen erityislaatuisuudesta, että niitä voidaan muokata. Viimeisimpänä, mutta ei vähäisimpänä seikkana, ohjelmistot, joiden käyttöikä on pitkä, ehtivät kypsyä viimeistellyiksi ja virheet hioutuvat pois.

Pitkällä aikavälillä tarkastellen keskeisessä roolissa tietojärjestelmän kehittymisessä

ovat ne yleiset periaatteet, jotka kehittämispäätöksiä ja -valintoja ohjaavat. Näitä yleisiä periaatteita kutsutaan evoluutiostrategiaksi [54, 53]. Warrenin mukaan [53] evoluutiostrategiat voidaan tiivistää kolmeen perustyyppiin:

Ylläpidon jatkaminen (engl. *continued maintenance*) merkitsee sitä, että järjestelmää pidetään toimintakuntoisena mahdollisimman konservatiivisesti. Muutoksia järjestelmään pyritään välttämään.

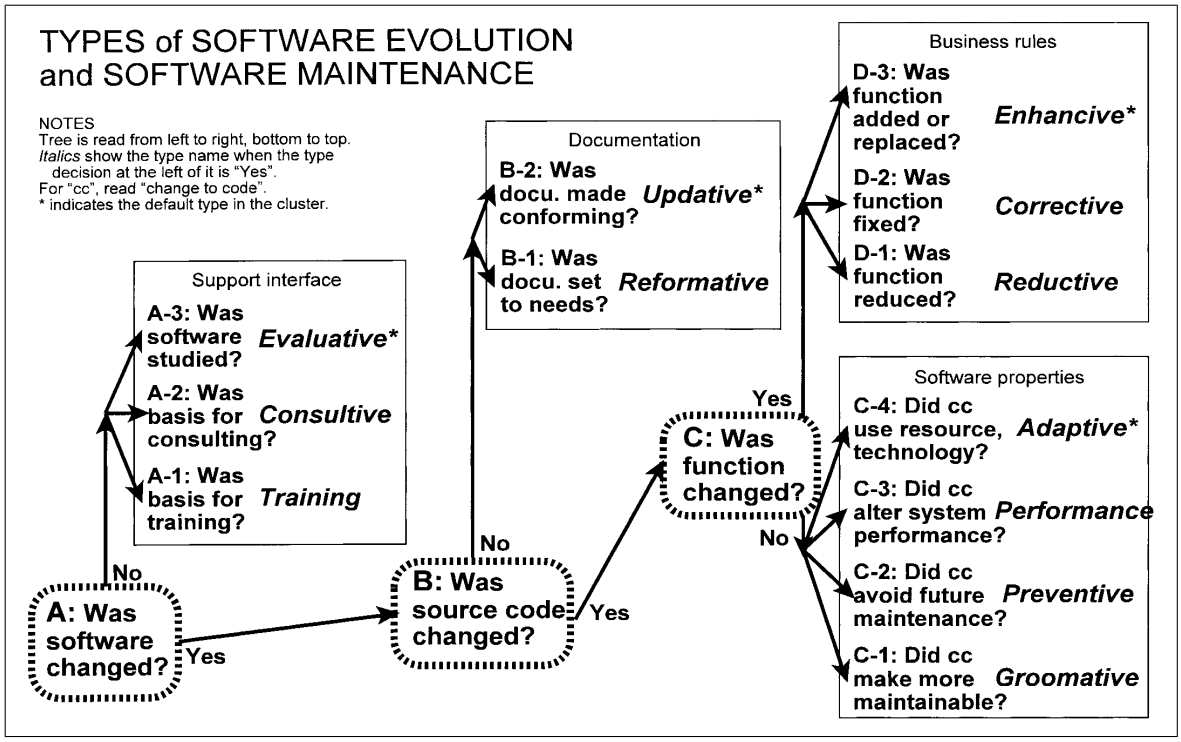
Uudelleenkonstruointi (engl. *reengineering*) merkitsee sitä, että järjestelmää analysoidaan ja muokataan tavoitteena muuttaa se uuteen muotoon [1, 32, 53]. Uudelleenkonstruointi ei ensisijaisesti kohdistu toiminnallisuuden muuttamiseen, vaikka sivutuotteena näin tavallisesti käykin.

Korvaaminen (engl. *replacement*) merkitsee sitä, että käytössä oleva järjestelmä hylätään ja sen tukemia (liike)toimintoja jatketaan käyttäen tukena toista tietojärjestelmää.

Chapin ym. mukaan [5] evoluutioaktiviteetit voidaan luokitella neljään pääluokkaan ja kahteentoista alaluokkaan. Pääluokkajaottelussa kriteereinä on aktiviteetin kohdentuminen. Muuttaako aktiviteetti ohjelmistoa, sen lähdekoodia tai toiminnallisuutta. Luokitus on kuvattu päätöspuun muodossa kuvassa 1.2.

Pitkäikäisten ohjelmistoinvestointien kohdalla on aina ratkaistavana eri evoluutiostrategioiden välillä tasapainoilun ongelma. Mikä etenemistapa missäkin tilanteessa on kokonaisuuden kannalta paras? Tämä ongelma on aidosti vaikea, myös investointilaskennan mielessä. Investointitilanteessa on usein piirteitä, jotka sulkevat yksinkertaiset ja suoraviivaiset laskentamallit pois käytöstä [20]. Toinen tapa lähestyä tätä samaa ongelmaa on kääntää se muotoon: "Kuinka radikaaleihin toimenpiteisiin ja milloin meidän on ryhdyttävä, ja mihin toimenpiteet on kohdennettava, että kykenemme pitämään asiakastyytyväisyyden [44, 43] riittävällä tasolla?" Asiantuntijuus on näissä päätöksissä ja niihin johtavien tapahtumaketjujen aloitteissa tärkeässä roolissa [25]. Asiantuntijuuden hyödyntämisen kehittäminen on siten kriittistä [24]. Asiantuntijuuden järjestelmällinen hyödyntäminen edellyttää riittävän yhtenäistä käsittely- ja esitystapaa, jotta esitetyt arviot olisivat nopeasti omaksuttavissa ja vertailukelpoisia, jolloin niitä voidaan tehokkaasti hyödyntää näkemysten kommunikointiin.

Tässä opinnäytetyössä tarkastelen uudistamistarpeen (tai yleisemmin evoluutiostrategian uudelleenarvioinnin tarpeen) varhaista havainnointia ja sen nykyistä menetelmällistä tukea. Konstruktiivisena osana muotoilen uuden varhaiseen havainnointiin sovel-



Kuva 1.2: Evoluution tyypit Chapin ym. mukaan (lähde [5])

tuvan arviointimenetelmän, MODESTin (Modernization Pressure Estimation) ja esittelen menetelmän käytöstä saatua empiiristä aineistoa.

Keskityn erityisesti evolutiivisiin tietojärjestelmiin, koska niiden kohdalla muutospaineet ja uudistamispäätösten problematiikka ovat tarkastellussa mielessä relevantteja. Evolutiivisilla järjestelmillä viittaan Lehmanin SPE-jaottelun tyypin E ohjelmia sisältäviin tietojärjestelmiin [26]. Lehmanin omaa muotoilua lainaten kyseessä ovat "järjestelmät, jotka ohjelmistoa hyväksikäyttäen ratkaisevat jonkin tosielämän ongelman ollen samalla itse osa mallintamaansa todellisuutta". Tyypin S ohjelmat ovat staattisiin määrittelyihin perustuvia ja tyypin P ohjelmat ongelmien mallinnuksia. Evoluutiostrategiaa koskeviin päätöksiin liittyvät ongelmat koskettavat tietysti kaikkia tietojärjestelmiä, joita tavalla tai toisella ylläpidetään. Tässä mielessä rajaus, jos sellaisesta halutaan puhua, on varsin lavea.

Tässä opinnäytetyössä kuvataan MODEST-mallin rakentaminen ja sen edellyttämä pohjatyö. Luvussa 2 jäsennetään kirjallisuuskatsauksen kautta aihealue ja käydään läpi aiempi tutkimus, erityisesti aiempi menetelmällinen tuki. Luku 3 kuvaa menetelmän empiirisen perustan ja suunnitteluperiaatteet. Luvussa 4 esitellään yksityiskohtaisesti MODEST-menetelmän rakenne. Luku 5 kuvaa menetelmän käytön ja luku 6 pilotoinnin ja kootun asiantuntijapalautteen. Lopuksi luvussa 7 vedetään johtopäätökset.

Menetelmäkehitys ja sen edellyttämä tutkimus on toteutettu osana ELTIS-projektia (Extending the LifeTime of Information Systems). Tutkimustyö on raportoitu yksityiskohtaisesti erikseen [24, 25]. Tämän tutkielman kirjoittaja osallistui ELTIS-projektiryhmän jäsenenä tutkimuksen suunnitteluun, toteutukseen (aineiston keruu, prosessointi ja analysointi) ja raportointiin. Tutkielman kirjoittaja vastasi menetelmän suunnittelusta, dokumentoinnista, tukimateriaalin valmistamisesta ja validoinnista. Muut projektiryhmän jäsenet ovat osallistuneet menetelmäkehitykseen katselmoimalla tutkielman kirjoittajan tuotoksia ja avustamalla pilotoinnissa.

ELTIS-projektissa on tutkittu muiden aiheiden ohella tässä opinnäyteessä käsiteltyä ongelmaa. Viittaukset projektiin liittyviin julkaisuihin löytyvät ELTIS-projektin julkaisuyhteenvedosta [48]. Tämän opinnäytetyön aiheesta on kirjoitettu myös englanninkielinen tekninen raportti [49], joka on myöhemmin tarkoitus julkaista vertaisarvioidulla foorumilla.

2 Kirjallisuuskatsaus

Tässä luvussa käydään läpi taustaa tietojärjestelmien ylläpidolle. Hieman syvemmin tutustutaan evoluutiostrategioihin sekä muutospaineiden ja -tarpeiden tunnistamisen problematiikkaan. Lopuksi tarkastellaan nykyisiä ohjelmistojen evoluution arviointimalleja uudistamistarpeen havaitsemisen näkökulmasta.

2.1 Ylläpito

Ohjelmiston elinkaarella (engl. *life cycle*) tarkoitetaan aikajaksoa ohjelmistokehityksen aloittamisesta siihen kun ohjelmisto poistuu käytöstä (mm. [10, 56]).

Ohjelmistojen elinkaarta voidaan jaotella eri vaiheisiin monilla tavoin [47, 39]. Luokittelut sisältävät ainakin suunnittelu- ja toteutusvaiheet. Näitä voi jaottelutavasta riippuen edeltää esitutkimus-, tarvekartoitus- tai määrittelyvaihe, joko kaikki tai joitakin, kuitenkin mainitussa järjestyksessä. Toteutuksen perään sijoittuvat testaus, käyttöönotto ja viimeisinä vaiheina ylläpito ja alasajo.

Haikalan [10] mukaan ylläpito on ”asiakkaan ongelmien ratkomista, virheiden korjaamista, ohjelman muuttamista vaatimusten muuttuessa sekä uusien piirteiden lisäämistä.” Ylläpitoa koskevissa IEEE standardeissa [16, 17] ohjelmiston ylläpito määritellään ohjelman käyttöönoton jälkeiseksi muuttamiseksi. Sen päämääränä voi olla virheiden korjaaminen, ohjelman laadun parantaminen tai ohjelman mukauttaminen erilaiseen ympäristöön. Tuorein ylläpidon standardi [18] on hyväksytty juuri äskettäin (30.3.2006), eikä minulla tätä kirjoittaessa ole tietoa sen sisällöstä. Muutokset aiempiin versioihin ovat kuitenkin varsin merkittäviä. Esimerkiksi 1219-1998-standardin prosessimalli korvautuu 14764-2006:n prosessimallilla.

Elinkaaren vaiheista ylläpito on tiettyjä erikoistapauksia lukuun ottamatta sekä työ-, että kalenteriaikana mitaten merkittävin. Poikkeustapauksia ovat lähinnä tavalla tai toisella ”kertakäyttöiset” ohjelmat, sellaiset sulautetut järjestelmät, joiden ohjelmistoja ei laitteen käyttöönoton jälkeen voi muuttaa ja ohjelmistotuotteet joissa ylläpito piilotetaan seuraavan version kehitysprojektiin. Tutkimusten valossa [9, 7, 33, 15, 38, 30, 55, 29, 45, 4, 47] suhteellisesti merkittävä osuus työstä ohjelmistoyrityksissä kuuluu tähän elinkaaren vaiheeseen (ks. taulukko 1.1, s. 1). Ylläpitokustannusten suhteellinen

osuus kokonaiskustannuksista on lisäksi kasvanut huolimatta uusien suunnittelumenetelmien käyttöönotosta [8]. Kehityssuunta on varsin luonnollinen, koska ylläpidettävä ohjelmistomassa kasvaa ja monimutkaistuu koko ajan.

Manny M. Lehman on muotoillut ohjelmistojen evoluution lait [27]. Olen suomentanut ne taulukkoon 2.1.

I	*Jatkuva muutos* (1974) E-tyypin järjestelmiä on lakkaamatta mukautettava. Muutoin niistä tulee tarkoitustaan yhä vähemmän palvelevia.
II	*Kasvava mutkikkuus* (1974) E-tyypin järjestelmän kehittyessä sen mutkikkuus kasvaa, ellei sen vähentämiseen erityisesti kiinnitetä huomiota.
III	*Itsesäätely* (1974) E-tyypin järjestelmän kehittymisprosessi on itsesäätelevä ja prosessi- ja tuotemetriikoiden jakaumat ovat lähellä normaalijakaumaa.
IV	*Organisaation vakauden säilyminen* (1980) Keskimääräinen tehokas aktiivisuuden taso kehittyvässä E-tyypin järjestelmässä on vakio koko elinkaaren ajan.
V	*Perehtyneisyyden säilyminen* (1980) E-tyypin järjestelmän kehittyessä on sidosryhmien säilytettävä perehtyneisyytensä järjestelmän rakenteeseen ja toimintaan, jotta järjestelmä kehittyisi tyydyttävällä tavalla. Siten järjestelmän keskimääräinen vaiheittainen kasvu on vakio.
VI	*Jatkuva kasvu* (1980) E-tyypin järjestelmän toiminnallisuuden on jatkuvasti lisäännyttävä, jotta käyttäjätyytyväisyys säilyy.
VII	*Heikkenevä laatu* (1996) E-tyypin järjestelmien laatu heikkenee iän myötä, ellei ylläpitoa tehdä hallitusti ja järjestelmää soviteta toimintaympäristön muutoksiin.
VIII	*Takaisinkytkentä* (1974, muotoiltu laiksi 1996) E-tyypin järjestelmien evoluutio on monitasoinen, monikehäinen ja monitoimijainen takaisinkytkentäprosessi ja sitä tulee myös pitää sellaisena, mikäli prosessia halutaan parantaa.

Taulukko 2.1: Ohjelmistojen evoluution lait

Lehmanin mukaan SPE-jaottelun E-tyypin tietojärjestelmiin liittyy erottamattomasti jatkuva muutospaine ("The pressure for change is built in" [26]). Tämä on seurausta siitä, että E-tyypin järjestelmä on osa toimintaympäristöään ja vaikuttaa siihen. Kaikki muutokset ohjelmaan vaikuttavat toimintaympäristöön ja toimintaympäristön muuttuminen vastaavasti luo paineita ohjelman muuttamiseen.

2.2 Evoluutiostrategiat

Tukeudun Warrenin [53] esittämään evoluutiostrategioiden kolmijakoon. Käytän kuitenkin Warrenista poiketen sijaan Harsun [13] ja Seacordin [45] mukaista terminologiaa.

Tässä yhteydessä tämä ilmenee siten, että Warrenin mukaisen uudelleenkonstruoinnin (engl. *reengineering*) sijaan vastaavan asian nimenä on uudistaminen (engl. *modernization*).

Menetelmän teoreettisen eheyden säilyttämiseksi viittaan jatkossa usein "nykyiseen evoluutiostrategiaan" ja "evoluutiostrategian muuttamiseen". Nykyinen strategia käytännössä aina tarkoittaa Warrenin strategioiden kolmijaon mukaista tyyppiä "ylläpidon jatkaminen". Strategian muuttaminen käytännössä aina tarkoittaa strategioiden kolmijaon mukaisesti "uudistaminen" tai "korvaaminen" -tyyppisiä strategioita, jotka käytännössä tarkoittavat rajallista ja siten projektoitavaa evoluutioaktiviteettia.

2.2.1 Uudistaminen

Ohjelman uudistaminen (engl. *renovation, modernization*) on yleinen käsite, joka kattaa ohjelman rakenteen selvittämisen, nykytilan dokumentoinnin, ohjelman ominaisuuksien (esim. rakenteen ja luettavuuden) parantamiseksi tehdyt suunnitelmat ja suunnitelmien toteutuksen. Taustalla on ajatus haluttujen ominaisuuksien parantamisesta suhteellisen konservatiivisin toimenpitein ja uudelleenkäyttöä painottaen.

Standardoidussa [17] terminologiassa uudistaminen on ylläpidon alakäsite. Tämä tulkinta on jossain määrin venyvä, koska uudistamisen tekniikat vaihtelevat radikaaliudeltaan hienoisesta ehostamisesta täydelliseen uudelleentoteutukseen. Tulkinta sille, missä tapauksessa uudelleentoteutus tuottaa uuden ohjelmiston ja vanhan elinkaari päättyy, taas on veteen piirretty viiva.

Aihealueen (erityisesti suomenkielinen) käsitteistö on vasta vakiintumassa ja elää voimakkaasti. Tässä opinnäytetyössä on uudistamisen käsitteistön osalta seurattu Harsun [13] tekemiä valintoja. Poikkeuksen tekee modernisointi-sanan (engl. *modernization*, Harsulla ajanmukaistaminen) käyttäminen uudistamisen kanssa vaihdannaisena. Käytäntö on yhtenevä Seacordin [45, s. 9] määritelmän kanssa, jonka mukaan modernisointi on (linja)ylläpitoa laajempi toimenpide, joka kuitenkin säilyttää merkittävän osan olemassa olevaa järjestelmää.

Uudistamiseen liittyy läheisesti monia termejä: parantaminen (engl. *improvement*), uusiminen (engl. *renewal*), uudistaminen (engl. *renovation*), kohentaminen (engl. *refurbishing*), uudelleenkehittämisen suunnittelu (engl. *redevelopment engineering*), ajanmukaistaminen (engl. *modernization*), pelastaminen (engl. *reclamation*), uudelleenkäytön suunnittelu (engl. *reuse engineering*). Näistä viisi ensimmäistä viittaa ohjelmiston laadun parantamiseen. Ajanmukaistaminen viittaa Harsun mukaan edellisten lisäksi ohjelmistokehityksen ja ylläpitotoimintojen parantamiseen. Kaksi viimeistä taas viit-

taavat lähdekoodin uudelleenkäytön helpottamiseen. Uusiminen tarkoittaa ohjelmiston laadun parantamista tulevaa kehittämistä ja jatkokäyttöä varten.

Uudistamista voidaan karakterisoida myös toiminnan tavoitteiden näkökulmasta. Tavoitteet ovat Rosenbergin mukaan [42] seuraavat.

Pohjatyö toiminnallisuuden lisäämiselle Etenkin laajempi toiminnallisuuden kehittäminen ja lisääminen vaatii pohjustavaa työtä. Niissäkin tapauksissa, missä pohjustus ei sinänsä ole välttämätöntä, on se silti hyödyllistä. Ohjelman rakennetta täytyy ehkä selkiyttää, dokumentaatiota päivittää ja tuottaa lisää, osien yleiskäyttöisyyttä ja tehokkuutta hioa. Hyvin tehdyn valmistelun jälkeen lisätoiminnallisuuksien suunnittelu ja toteutus sujuvat helpommin.

Ylläpidon helpottaminen Ohjelmiston rapautuminen heikentää ylläpidettävyyttä. Uudistamisella pyritään korjaamaan ja osittain ennalta ehkäisemään (kasautuvien vaikutusten osalta) rapautumisen kautta tapahtuvaa ylläpidettävyyden heikentymistä.

Uuteen ympäristöön siirtyminen Ohjelmisto tai sen osia saatetaan jossain vaiheessa (esim. teknologian tuen loppumisen johdosta) siirtää uuteen ympäristöön. Tällöin suurella todennäköisyydellä tarvitaan mahdollisesti laajojakin uudistustoimenpiteitä ohjelmiston sovittamiseksi valittuun uuteen toimintaympäristöön.

Luotettavuuden parantaminen Ohjelmistoon tehtävien muutosten myötä tapahtuva rapautuminen heikentää myös luotettavuutta. Uudistamisella pyritään korjaamaan ja ennalta ehkäisemään myös tätä rapautumisen vaikutusta.

2.2.2 Korvaaminen

Korvaaminen viittaa käytössä olevan järjestelmän hylkäämiseen ja sen tukemien (liike)toimintojen jatkamiseen käyttäen tukena toista tietojärjestelmää. Tähän evoluutiostrategiaan ei sisälly uudelleenkäyttöä millään tasolla. Monissa tapauksissa uudelleenkäyttöä koskevien reunaehtojen rajaamana (esim. sopimukset, teknologian tuki ja yhtiöjärjestelyt) ajaudutaan korvaamaan käytössä oleva ja toimiva järjestelmä.

Tietojärjestelmän korvaaminen voi konkreettisesti tapahtua montaa eri tietä. Vastaavien toimintojen hoitamiseksi voidaan joko itse tai ulkopuolisella avulla kehittää kokonaan uusi järjestelmä. Toiminnot saatetaan siirtää toisen, jo käytössä olevan, tietojärjestelmän hoidettavaksi. Useissa organisaatioissa käytössä olevilla tietojärjestelmillä on toiminnallista päällekkäisyyttä, joskus merkittävästikin. Toimintojen hoitamiseen

voidaan hankkia markkinoilta olemassa oleva ns. “hyllymetri” pakettiratkaisu tai räätälöitävä pakettiratkaisu.

2.2.3 Ylläpidon jatkaminen

Ylläpidon jatkaminen merkitsee järjestelmän pitämistä toimintakuntoisena mahdollisimman konservatiivisesti. Muutoksia järjestelmään pyritään välttämään. Lyhyen tähtäimen taloudellisia vaikutuksia tarkastellen, tämä strategia on käytännössä aina tavoiteltavin vaihtoehto.

2.3 Muutospaineiden ja -tarpeiden tunnistaminen

Nykytilan analysointi ja vaikuttavien tekijöiden tunnistaminen on tietojärjestelmän evoluution suunnittelun kannalta keskeistä [39, 47, 10].

2.3.1 Organisaation kyvykkyys

Ohjelmistoprosessien kypsyydellä ja organisaation kyvykkyydellä on keskeinen vaikutus laatuun [14]. Kyvykäs organisaatio kykenee järjestelmällisesti tunnistamaan merkitykselliset asiat, tässä tapauksessa tietojärjestelmiin kohdistuvat muutospaineet, ja reagoimaan niihin ennakoivasti. Havainnot eivät ole yksittäisten sattumien tai sankaritekojen varassa, vaan osa organisaation tavanomaista toimintaa, ne on institutionaalistettu. Jos luotetaan siihen, että uudistamisen tarve ”jotenkin vain huomataan” joudutaan tyytymään siihen, tarpeiden ilmeneminen on yllätyksellistä ja epävarmaa. Järjestelmällinen tunnistaminen edellyttää järjestelmällistä havainnointia.

Muutospaineiden tunnistamisessa ei ole kyse yksin kehittäjä- tai käyttäjäorganisaation kyvykkyydestä, vaan näiden kahden osapuolen yhteisestä kyvykkyydestä. Tieto abstraktiosta on kehittäjällä ja tieto käytön todellisuudesta on käyttäjällä. Molempia tarvitaan havainnointiin.

Yksi virhearvioinnin mahdollisuus piilee siinä, että olemassa olevan järjestelmän monimutkaisuus ja ongelmat ovat, elleivät jo hyvin tiedossa, niin ainakin olemassa ja tutkittavissa. Uuden järjestelmän tai osan monimutkaisuus ja ongelmat taas eivät sitä ole. Näin mukauttamisen ja uudistamisen kokonaiskustannukset helposti oletetaan todellisuutta suuremmiksi verrattuna uudelleentoteuttamisen tai kokonaan uuden järjestelmän kehittämisen kustannuksiin [37]. Näin siltikin vaikka absoluuttisesti kustannukset usein aliarvioidaan, erityisesti uudiskehityshankkeissa.

2.3.2 Arviointimallit

Tarkastelen tässä nykyisten ohjelmistojen evoluution arviointimallien suhdetta uudistamistarpeiden havainnointiin.

Sneedin engl. *Reengineering Planning Process (RPP)* [46] on menetelmä uudistamisen hyödyllisyyden arviointiin. RPP on viisivaiheinen:

1. Portfolio-oikeutus (engl. *portfolio justification*), määritetään tuottoaste ohjelmiston liiketoiminta-arvon, laadun ja ylläpitoprosessin parantumisen kautta.
2. Portfolioanalyysi (engl. *portfolio analysis*), sovellusten uudistustarpeet priorisoidaan niiden teknisen laadun ja liiketoiminnallisen arvon mukaan.
3. Kustannusarvio (engl. *cost estimation*), arvioidaan kustannukset komponenteittain.
4. Kustannus-hyöty -analyysi engl. *cost-benefit analysis*), verrataan kustannuksia ja hyötyjä kolmen strategian välillä: a) uudistaminen b) uudelleentoteutus c) ylläpidon jatkaminen.
5. Sopimusneuvottelu (engl. *contracting*), sopimus voi perustua mm. aikaan, muihin resursseihin tai tuloksiin.

Visaggio esittelee artikkelissaan [50] uusimisen (engl. *renewal*) päätösmallin engl. *Value-based Decision Model* (VDM). Se hyödyntää samaa teknisen laadun ja liiketoiminnallisen arvon nelikenttätarkastelua kuin Sneedin RPP:n portfolioanalyysi.

Warren ja Ransom esittelevät artikkelissaan [54] evoluution kokonaisvaltaisen prosessikehyksen nimeltä *Renaissance*. Siinä pyritään ensiksi vakaaseen lähtötilanteeseen käyttäen apuna uudistamisen menetelmiä. Lähtötilanteesta sitten aloitetaan jatkuva inkrementaalinen muutosprosessi. Renaissance hahmottaa järjestelmän nykytilan liiketoiminnallisen arvon ja teknisen laadun määrittämisen kautta samaan tapaan kuin RPP:n portfolioanalyysi ja VDM.

Renaissancen kaltainen korkean tason kehys on myös SABA [3]. Se on skenaariopohjainen kehys tietojärjestelmän evoluution ja siirroksen (engl. *migration*) suunnitteluun. SABA huomioi teknisten asioiden lisäksi myös organisatoriset seikat.

Wallace *ym.* esittelevät artikkelissaan [52] uudistamisen päätöskehikon, joka käsittelee päätöstilannetta syötteiden ja tulosten ominaispiirteiden kautta. Päätöksen tulos hahmotetaan tässä mallissa koostuvaksi teknologisesta tavoitetilasta, järjestelmäarkkitehtuurin tavoitetilasta, integraatiostrategiasta ja siirtymästrategiasta. Näiden valintaa

ohjaavat syötteet, eli uudistamisen tavoitteet, teknologiset mahdollisuudet ja nykyinen järjestelmäarkkitehtuuri.

Uudistamisen päätöksentekoon liittyvä menetelmällinen tuki voidaan hahmottaa kolmeksi kokonaisuudeksi: strategisen tason arviointimallit (SABA, Renaissance ja Sahin & Zahedi [44]), kustannusarviointimallit (FPA, COCOMO, SOFTCALC), yksityiskohtaiset arviointimallit (VDM, RPP).

Edellä kuvatut menetelmät pureutuvat uudistamisen oikeutukseen ja läpivientiin, mutta eivät itsessään anna eväitä tarpeen havaitsemiseen. Lisäksi mainitut analyysi- ja arviointimenetelmät ovat luonteestaan ja tarkoituksestaan johtuen varsin raskaita [22]. Niiden jatkuva kattava käyttö uudistamistarpeen seuraamiseen ei siten ole mielekästä.

3 Menetelmän perusta

Menetelmän suunnitellussa on tarpeellista edetä jäsennellysti vaiheittain. Kirjallisuuskatsaus antaa kuvan uudistamisen menetelmällisen tuen nykytilasta. Lisäksi on tarpeen hahmottaa myös päätösprosessia, jotta menetelmä voidaan suunnitella ottaen huomioon ohjelmistoteollisuuden päätösprosessien haasteet.

3.1 Uudistamispäätösten päätöksentekoprosessien tutkimus

Uudistamistarpeen havainnoinnista alkava ketju kulminoituu päätöstilanteeseen, jossa valitaan reagointitapa. Näin ollen tehtyjen havaintojen tulisi suoraan tukea päätöksentekijää tässä päätöstilanteessa. Uudistamispäätösten syntyprosesseja ei aiemmin ole selvitetty, joten päätöksentekijän näkökulmasta tehtävä selvitystyö uudistamispäätösten päätöksentekoprosesseista on välttämätöntä päätöksenteon tukimenetelmän suunnittelun pohjaksi. Kun lisäksi huomioidaan Jørgensenin havainto, että parhaidenkaan mallien selitysvoima ei toistaiseksi yllä asiantuntijoiden tasolle [19], on asiantuntijuuden laadun ja luonteen selvittäminen uudistamispäätösten kontekstissa perusteltua.

3.1.1 Taustaa

Tässä luvussa kuvattava tutkimus on toteutettu osana ELTIS-projektia ja raportoitu yksityiskohtaisesti erikseen [24, 25]. Osallistuin ELTIS-projektiryhmän jäsenenä tutkimuksen suunnitteluun, toteutukseen (aineiston keruu, prosessointi ja analysointi) ja raportointiin.

Tutkimuksessa selvitimme uudistamisen päätöksentekoa. Halusimme tietää millaisia tarpeita ja arvottamisia evoluutiopäätöksiin liittyy ja mitä päätöksentekijä toimintansa tueksi tarvitsee. Yhtäältä tavoitteena oli päätöksenteon optimaaliseen tukeen liittyvä selvitystyö ja toisaalta päätöksentekijöiden arvotukset päätöskriteereille.

Onnistunut ohjelmistotekniikan empiirinen käsittely edellyttää kattavaa aineistoa, kaupallisen yhteistyökumppanin halua sitoutua tutkimukseen ja tutkimuksen järjestelmällistä toteuttamistapaa [21].

Jäsentymätön aihealue ja oletus, että esiin mitä todennäköisimmin nousee ennakoimattomia asioita puolsivat laadullisten menetelmien käyttöä [6]. Toisaalta tavoite selvittää päätöksentekijöiden arvotuksia eri päätöskriteerien suhteen edellytti määrällistä tai vähintäänkin määrällistettävissä olevaa keruutapaa.

3.1.2 Suunnittelu ja toteutus

Päätöksentekoprosessiin ja tietotarpeisiin liittyvä tiedonkeruu tehtiin puolistrukturoidun haastattelun avulla. Kriteerien arvottamisia selvitettiin haastateltujen henkilöiden haastattelun jälkeen täyttämien lomakkeiden avulla.

Haastattelu ja kyselylomake suunniteltiin ELTIS-projektiryhmän toimesta iteratiivisen prosessin kautta. Lomakkeen ja haastattelun tavoitteita tarkennettiin, kerättiin tietoa kirjallisuudesta ja kokemuksia testihaastatteluista ja lomakkeen täytöistä (koekaniineina muita yksikön tutkijoita) ja muokattiin haastattelua ja lomaketta näiden pohjalta.

Näytteistykseen käytettiin lumipallo-otantaa, joka on tavoitettavuudeltaan haastavien populaatioiden yhteydessä vakiintunut menetelmä [51]. Haastateltavia oli kaikkiaan 29. He edustivat 8:aa organisaatiota, joista 3 oli toimittajaorganisaatiota, 5 käyttäjäorganisaatiota, 2 julkisen sektorin ja 6 yksityisen sektorin organisaatiota. Yhteenveto haastateltujen taustatiedoista on taulukossa 3.1.

Tiedonkeruu aloitettiin elokuussa 2003 ja saatiin päätökseen helmikuussa 2004. Valittu tiedonkeruutapa osoittautui toimivaksi tavaksi paitsi kerätä tietoa, myös sitouttaa osallistujat tutkimukseen. Lomakkeita lähetettiin kaikille haastatelluille, eli 29, joista 26 (90 %) palautui. Tätä voidaan pitää varsin hyvänä ottaen huomioon kohderyhmä.

3.1.3 Tulokset

Tutkimus tuotti uudistamisen päätöksenteon tuen kannalta relevantteja tuloksia. Lisäksi MODESTin parametriston laadinnassa hyödynnettiin vaiheessa myös asiantuntijoiden lomakekyselyn kautta antamia päätöskriteerien painotuksia.

Päätöksentekijöiden toiveena oli argumentoinnin ja valmistelun yleinen systematisointi. Tukimenetelmän näkökulmasta tämä merkitsee yksinkertaista toimintatapaa ja jäljitettävyyttä, joiden kautta päätöksentekijän on helpompi jäsentää päätöstä edeltävien toimien systemaattisuus.

Päätöksentekijät kokivat yleisesti epävarmuutta siitä, että ovatko he tulleet ajatelleeksi kaikkia päätöksen kannalta välttämättömiä ja olennaisia seikkoja. Useinkaan päätök-

Piirre	**Arvo**
Ikä (keskiarvo)	48 v
Kokemus (keskiarvo)	
IT-alalta	19 v
ohjelmistojen ylläpidosta	13 v
modernisointipäätöksistä	8 v
On vastuussa	
päätöksestä	3
argumentoinnista	6
molemmista	10
Työskentelee	
yksityisellä sektorill	18
julkisella sektorilla	11
Työskentelee	
käyttäjäorganisaatiossa	17
toimittajan organisaatiossa	12

Taulukko 3.1: Haastateltujen asiantuntijoiden taustatiedot

set eivät synny toistettavan prosessin kautta, joten hyvin suurella todennäköisyydellä olennaisten seikkojen ohittaminen ei jää pelkäksi mahdollisuudeksi. Tähän asiaan liittyi useammalla haastateltavalla myös toive jonkinlaisesta avusta "näkökulman avartimesta."

Intuition rooli päätöksenteossa on keskeinen ja merkittävä, muttei läpinäkyvä. Tällä on potentiaalisesti useita haitallisia vaikutuksia. Erityisesti argumenttien sepittäminen [36].

Päätöksiin ei yleisesti ottaen niiden tekemisen jälkeen enää palata. Lisäksi dokumentaatio päätöksistä on yleensä tallennettu käytettävyydeltään heikkoon muotoon. Epäselvää on, onko näiden asioiden välillä kausaalista yhteyttä, ja jos on niin kumpi niistä on syy ja kumpi seuraus. Nämä yhdessä vaikuttavat kuitenkin siihen, että Lehmanin laeissa mainittu ohjelmistojen evoluutioprosessin välttämätön takaisinkytkentä näyttäisi tältä osin puuttuvan.

3.2 Menetelmälle asetettavat vaatimukset

Arviointimallin tulee tukea päätöksentekijää päätöksentekoprosessin aikana. Päätöksentekijät toivovat tukea mm. uusien näkökulmien löytämiseen ja päätösperusteiden systemaattiseen käsittelyyn. Lopputuotteena syntyvän menetelmän on oltava arvioinnin yhteydessä käytävää keskustelua ohjaava sekä keskustelun johtopäätösten doku-

mentointia ja kommunikointia tukeva. Näin päätöksentekijä kykenee saamaan tehdystä arviosta riittävän hyödyn.

Arviointimallin on oltava kevyt käyttää. Muuten käy niin, ettei sitä käytetä lainkaan [25], tai jos käytetään, on vaarana, että liika nykytilan analysointi aiheuttaa siihen juuttumisen ja luovuuden kuolemisen [12, s. 20], eli ns. analyysihalvauksen (engl. *paralysis of analysis*). Laajan Renaissance-projektin tuotosten [54] implementoinnissa on ollut vaikeuksia nimenomaan resurssivaatimusten vuoksi.

Menetelmä ei saa rajautua pelkästään käyttäjä- tai kehittäjäorganisaation sisään. Kommunikointia myös tällä välillä tulee tukea. Käyttäjäorganisaatiolla on todellinen tietämys sovelluksen käytön tilasta. Kehittäjäorganisaatiolla taas todellinen tietämys sovelluksen "abstraktion venyvyydestä", eli mihin kaikkeen sovellus taipuu.

Arvioitavaa järjestelmää on tarkasteltava alijärjestelmätasolla [50]. Jako alijärjestelmiin helpottaa jäsentämistä.

Vaatimukset voidaan tiivistää seuraavasti.

V0 Uudistamistarpeen varhainen havaitseminen edellyttää (arvioinnin edellyttämien resurssien mielessä) kevyttä arviointimenetelmää [23]. *Muutospaineiden arviointi on oltava kevyt. Tavoitteena voidaan pitää yhden henkilötyöpäivän enimmäisresursseja yhden järjestelmän arviointiin.*

V1 Havaittiin, että uudistamisen päätöksentekoa tulisi systematisoida [25]. *Tukeakseen systematisointia MODESTin tulee olla hyvin määritelty ja jäljitettävä.*

V2 Päätöksentekijät kertoivat tarvitsevansa tavan avartaa ajattelua ja löytää uusia tarkastelunäkökulmia käsillä olevaan asiaan [25]. *Menetelmän tulee altistaa käyttäjänsä asiantuntijakollegoiden näkemyksille ja tarjota käyttöön sopivaa näkökulmien joukkoa tai tapaa etsiä näkökulmia.*

V3 Intuition osuus päätöksenteosta ei ole läpinäkyvä, koska sitä ei pidetä hyväksyttävänä argumenttina [25]. Sen sijaan pyritään löytämään muita, jopa sepitettyjä, argumentteja, mielellään numeerisia [36]. *Menetelmän tulisi tarjota tapa formalisoida myös intuitiiviset näkemykset.*

V4 Historiatietoa aiemmista päätöksistä on olemassa, mutta sen käytettävyys on yleensä heikko [25]. *Menetelmän tulosten tulisi raportoitua havainnolliseen ja tiiviiseen muotoon.*

V5 Päätöksentekijät ovat yleensä haluttomia palaamaan aiempiin päätöksiin arvioidakseen niiden laatua [25]. *Menetelmän tulisi edellyttää sen tuottamien tulosten myöhempi arviointi ja antaa hyvin määritelty liipaisin sen käynnistämiseen.*

V6 Päätös on aina yhteistoiminnan tulos ja päätöksentekijä, olipa hän sitten yksilö tai ryhmä, on eri lähteistä tulevan tiedon kokoaja [25]. *Menetelmän tulee tukea keskustelua ja viestintää sidosryhmien välillä.*

V7 Ohjelmistoja analysoitaessa, on suositeltavaa toimia alijärjestelmätasolla [50]. *Arvioinnin kohteena oleva järjestelmä tulisi jakaa osiin.*

V8 *Menetelmän tulee tukea itsensä arviointia ja muuttamista. Sen tulee kyetä muuttumaan ja mukautumaan jatkuvasti muuttuvaan toimintaympäristöönsä.*

V9 *Menetelmää käyttäen saadusta tuloksesta pitää olla tehtävissä johtopäätös siitä, voidaanko ylläpitoa nykyisellä evoluutiostrategialla turvallisesti jatkaa.*

Johtopäätöksiä koskeva vaatimus (V9) oli aluksi tiukempi sisältäen myös vaatimuksen pitävästä johtopäätöksestä koskien järjestelmän uudistamistarvetta. Myöhemmin osoittautui, että tämä osa vaatimusta oli käytännössä ristiriidassa keveysvaatimuksen (V0) kanssa ja V9 muotoiltiin uudelleen.

4 Menetelmän kuvaus

Tässä luvussa kuvataan se, miten MODEST rakentuu vaihe vaiheelta perustellen haastatteluaineistosta tehdyillä havainnoilla ja (substanssi- ja metodologisella) kirjallisuudella.

4.1 Menetelmän yleinen rakenne

Tieteellinen lähestymistapa mallin rakentamiseen on nelivaiheinen: havainnointi, yleistys, kokeilu, validointi (engl. *observation, generalization, experimentation and validation*) ja iteratiivinen [40, s. 4].

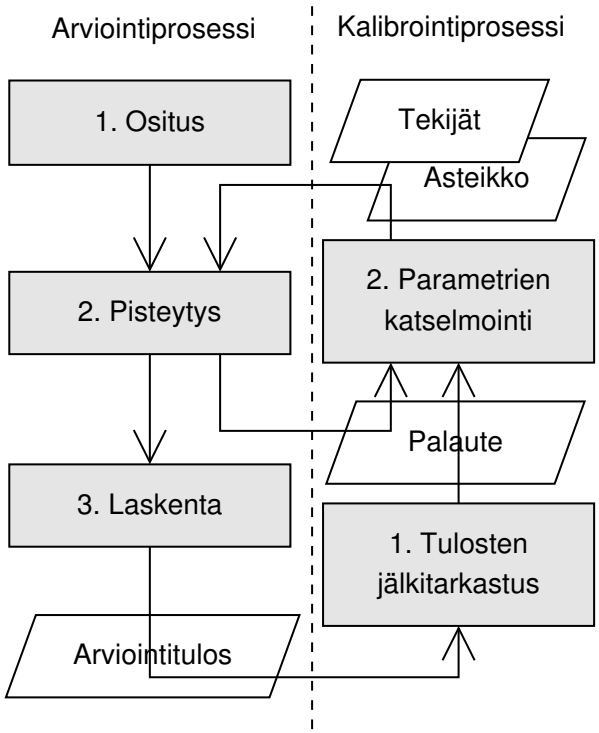
Muutospaineiden ja uudistamistarpeen suhteen havainnointi pohjaa kirjallisuudessa raportoitujen havaintojen ja aiemman menetelmällisen tuen järjestelmälliseen läpikäyntiin sekä edellä kuvattuun empiiriseen tutkimukseen uudistamispäätösten synnystä. Ositusvaatimuksesta (V7) ja lisänäkökulmavaatimuksesta (V2) lähtien päädyimme ajatukseen muutospaineiden arviointimatriisista, jossa kahtena ulottuvuutena ovat näkökulmat uudistamispäätökseen ja arvioinnin kohteen osat.

Havainnot tutkimuksestamme ja kirjallisuudesta yleistettiin muutospaineiden suhteen lineaarisesti määräytyväksi uudistamistarpeeksi (MODEST versiot 1.X). Tältä pohjalta formuloitua menetelmää katselmoitiin projektiryhmän sisällä ja ulkopuolisten asiantuntijoiden toimesta, kokeiltiin projektiryhmän sisäisillä testiarvioinneilla ja todellisilla arvioinneilla. Katselmoinneissa havaittiin valitun mallin edellyttävän pohjatietoja, joita ei asetettujen vaatimusten puitteissa voitu edellyttää olevan käytettävissä. Malliksi vaihdettiin muutospaineiden alarajasta määräytyvä evoluutiostrategian uudelleenarvioinnin tarve. Se edellytti myös tuloksia koskevan vaatimuksen (V9) keventämistä.

Malli perustuu yksittäisten muutospaineiden tunnistamiseen ja niiden voimakkuuden määrittämiseen laadullisesti kuvailtua luokka-asteikkoa hyödyntäen. Tunnistamisessa tukeudutaan muutospaineiden kohdentumiseen arvioinnin kohteena olevaan järjestelmään ja muutospaineiden yleiseen luokitteluun muutospainetekijöiden avulla. Odotettavien kohdentumisten ja tekijöiden läpikäynti tukee muutospaineiden järjestelmällistä tunnistamista. Tekijät käsitetään mallissa muutospaineavaruuden joukkoina. Tekijöiden olisi arvioinnin yksinkertaisuuden kannalta hyvä olla pistevieraita ja niiden yhdiste

mahdollisimman kattava. Kuitenkin siten että tekijät tosiasiassa antavat uusia näkökulmia ja sitä kautta helpottavat muutospaineiden tunnistamista (vrt. V2).

Menetelmä koostuu syöte/tulosdokumenteista ja viidestä aktiviteetista. Aktiviteetit muodostavat kaksi itsenäistä prosessia: Arvioinnin (engl. *evaluation*), joka tuottaa varsinaisen arviointituloksen, muutospainearvion, ja kalibroinnin (engl. *calibration*) joka on tukiprosessi ja käyttää palautetta parametrien hienosäätöön. Menetelmän rakenne on esitelty kuvassa 4.1.



Kuva 4.1: MODEST-menetelmän rakenne

MODEST tukee CMMI:n mukaisen vaatimusten hallinnan (engl. *REQM*) alueen vaatimusten ja toteutuksen keskinäisen poikkeavuuden tunnistamista (engl. *CMMI: Engineering: REQM: SP 1.5-1*) sekä tuotelaadun valvontaa (engl. *CMMI: Support: PPQA: SG 2*).

4.2 Parametrit

Menetelmän parametristo muodostuu *tekijöistä*, jotka toimivat osituksen ohella ajattelun apuna muutospaineita tunnistettaessa (vrt. V2), tunnistettujen muutospaineiden voimakkuuden kuvaamiseen käytetystä *asteikosta* ja arviointituloksen tulkinnassa käytettävistä kahdesta *raja-arvoista*.

Muutospaineen tunnistamisella viitataan tietyn tekijän järjestelmä- tai arviointikohtaisen konkreettisen ilmentymän tunnistamiseen. Arvioinnissa voitaisiin esimerkiksi tunnistaa, että tekijällä "Käytetyn teknologian tuki" on tässä tapauksessa olemassa konkreettinen ilmentymä, "toimittaja on ilmoittanut käyttämämme teknologian tuen loppuvan". MODESTin oletustekijäluettelo perustuu ELTIS-projektin puitteissa toteutettuun kyselyyn [24]. Luetteloon valittiin päätöksentekijöiden tärkeimmiksi arvottamat tekijät. Muu parametristo on laadittu kirjallisuuteen ja päätöksentekijöiden haastattelun tuloksiin [25] pohjautuen. Parametristo ei millään muotoa ole lopullinen tai täydellinen, mutta siitä on pyritty saamaan mahdollisimman hyvin perusteltu ja yleisesti käytettävä lähtökohta parametrien organisaatiokohtaiseen kalibrointiin.

4.2.1 Tekijät

Yritykseen (ja välillisesti prosessien kautta myös järjestelmiin) kohdistuvat paineet tulevat Rivettin [41, s. 21] mukaan seuraavilta tahoilta: paikallinen yhteisö (engl. *local community*), yhteiskunta (engl. *the nation*), omistajat (engl. *owners*) (nämä kolme keskenään kytkettyjä), analyytikot (engl. *financial analysts*, asiakkaat/kuluttajat (engl. *customers/consumers*), kilpailijat (engl. *rivals*), työvoima (engl. *labour (professional), labour (non-professional)*), tavarantoimittajat (engl. *suppliers*).

MODEST-menetelmää rakentaessamme otimme tekijöiden valintaan konkreettisemman lähestymistavan. Keräsimme kirjallisuudesta ohjelmistojen uudistamispäätöksiin vaikuttavia tekijöitä (tähän vino pino viittauksia lähteisiin) ja koostimme niistä yhteenvedon. Uudistamispäätösten asiantuntijat arvottivat tekijät (ks. 3.1.2, s. 14) ja päätimme valita niistä (keveyslinjausta, ks. 3.2, s. 15, noudattaen) 15 eniten painottunutta.

4.2.2 Asteikon määrittely

Tämä on vaikeimpia asioita, koska tulkinnoista syntyy hyvin helposti merkittäviäkin poikkeamia. Toisaalta pitkälle viety täsmennys ja konkretisointi tekee määrittelystä organisaatiospesifin ja yleisesti pätemättömän. Päädyttiin kompromissina geneerisesti kuvailtuun luokka-asteikkoon. Muutospaineita arvioidaan kuusiportaisella asteikolla. Asteikon arvot ja niiden määritelmät ovat seuraavat.

olematon (engl. *none*), 0 Muutoksiin ei ole syytä.

havaittava (engl. *noticeable*), 1 Tekijän suhteen on pieni, mutta havaittava, muutospaine.

selvä (engl. *apparent*), 2 Ongelmia tekijän suhteen on säännöllisesti ja ne on pantu merkille laajalti.

ongelmallinen (engl. *troubling*), 3 Ongelmat tekijän suhteen on otettava huomioon päivittäisessä työskentelyssä.

haitallinen (engl. *defeating*), 4 Resursseja menetetään säännönmukaisesti. Muutos toisi selviä hyötyjä, muttei kiistämättömästi enempää kuin haittoja.

sietämätön (engl. *intolerable*), 5 Tekijän suhteen on tai tulee lähitulevaisuudessa olemaan jatkuvia, vakavia vaikeuksia. Muutospaineeseen reagoinnin tuotos on lähes väistämättä tarvittavaa panostusta suurempi.

Tässä annettu kuvaus on tiivis ja sisältää vain määrittelyt. Asteikkoa koskevia konkreettisia tulkintaohjeita esimerkkeineen löytyy käyttöohjeosiosta (ks. luku 5.2.5, s. 37).

Lähtöarvot tulkinnassa käytettäville raja-arvoille ovat yläraja-arvolle (myöhemmin T_u) "sietämätön" (5) ja alaraja-arvolle (myöhemmin T_l) "ongelmallinen" (3).

4.3 Arviointiprosessi

Kohteena olevan järjestelmän arviointi suoritetaan yhden tapaamisen, *arviointi-istunnon*, aikana. Istuntoon täytyy ottaa osaa vähintään yhden asiantuntijan, jolla on tuoretta ensi käden tietoa arvioinnin kohteena olevasta järjestelmästä. Istunnon läpivientiin vaaditaan lisäksi, että puheenjohtajan ja kirjaajan tehtäviin nimetään istunnossa läsnä olevat, kyseisiin tehtäviin kykenevät henkilöt. Arvioinnin aikana tehdyt menetelmää koskevat havainnot kirjataan arviointiraportin palauteosaan. Periaatteessa arvioinnin voi hoitaa yksi henkilö, mikäli hän kykenee toimimaan puheenjohtajana ja kirjaajana ja on arvioinnin kohteena olevan järjestelmän asiantuntija. Menetelmän keskeinen päämäärä sisältyy kuitenkin asiantuntijoiden ohjatusti käymään järjestelmällisesti dokumentoituun keskusteluun, josta ei "pasianssiarvioinnissa" tietenkään päästä hyötymään.

Menetelmän varsinainen käyttäjä on arviointi-istunnon puheenjohtaja. Hänen ei ole välttämätöntä olla perehtynyt arvioinnin kohteeseen. Puheenjohtaja voi olla esim. arvioitavan järjestelmän kehittämisestä tai kehittämispäätöksien esittelystä vastaava toimittajan edustaja tai ulkopuolinen konsultti, käyttäjäorganisaatiossa kohteena olevien järjestelmien kehittämistä koordinoiva henkilö tai erityinen arvioija.

Samaa arvioinnin kohdetta koskien ainoastaan ensimmäinen arviointi tehdään "puhtaalta pöydältä". Seuraavat arvioinnit tehdään uudelleenarviointeina, eli otetaan pohjaksi edellisen arvioinnin raportti ja käydään siitä käsin läpi arviointiprosessi.

Tässä luvussa kuvataan järjestelmän ositus (4.3.1), pisteytys (4.3.2), muutospaineindikaattorin laskenta (4.3.3) ja tuloksen tulkinta (4.3.4). Arviointi-istunnon järjestämiseen liittyviä asioita käsitellään käyttäjän oppaassa (5.2).

4.3.1 Tehtävä 1: Ositus

Arvioinnin kohteen osituksen ainut tehtävä on helpottaa muutospaineiden tunnistamista (vrt. V2 ja V7, luku 3.2, s. 15). Yleisesti ottaen arvioinnin kohteen osittaminen on hyvä tapa auttaa arvioinnin kohteeseen liittyvien asioiden tunnistamista ja yksilöintiä [50].

Osituksen tuloksena saatujen osien yhdisteen tulee kattaa koko arvioinnin kohteena oleva järjestelmä, ja vain se. On luontevaa ja arvioinnin yhteydessä käytävän keskustelun kannalta hyödyllistä, että osat ovat keskenään pistevieraita. Se ei kuitenkaan ole menetelmän tuottaman arviointituloksen kannalta kriittinen seikka.

Arvioinnin kohteena olevan järjestelmän osittaminen tulisi tehdä mahdollisimman luontevalla tavalla pitäen mielessä osituksen tarkoitus. Arvioijan olisi hyvä tunnistaa, onko järjestelmässä osia, jotka ovat peräisin eri lähteistä tai joita ylläpidetään eri osapuolten toimesta. Eri sidosryhmien (loppukäyttäjät, käyttäjäorganisaatio, ylläpitäjät, toimittaja, ...) näkökulmat, sen mukaan miten heitä arvioinnissa voidaan hyödyntää, ovat myös käyttökelpoisia lähtökohtia ositukselle. Lisäksi ositus voi perustua esim. ohjelmistoarkkitehtuuriin (MVC, kerrosrakenne, pipes & filters, ...), käytettyihin teknologioihin, toteutettuihin toiminnallisuuksiin tai tuettuihin liiketoimintaprosesseihin.

Yhtä oikeaa sääntöä osien määrälle ei voida antaa. Hyvänä peukalosääntönä (ellei mitään muita lähtökohtia, tai kokemusta ole käytettävissä) osien määrälle voidaan pitää ns. "oikeaa lukua", eli $7(\pm 2)$ [31]. Mille tahansa epätriviaalin kokoiselle järjestelmälle 2–3 osaa on hyvin todennäköisesti liian karkea palvellakseen tarkoitustaan. Jos taas järjestelmä jaetaan liian moneen osaan tulee arvioinnista tarpeettoman raskas prosessi, eikä jakoa hienontamalla kuitenkaan saavuteta tulosten suhteen lisäarvoa. Noin 25 osaa voi hyvin olla – ja 50 osaa aivan varmasti on – liian paljon. Useiden keskenään ortogonaalisten jakojen käyttöä (ositus tehdään ensin yhdellä periaatteella ja sen jälkeen kaikki osat ositetaan toisella, ortogonaalisella, periaatteella) tulisi varoa, koska siten osien määrä kasvaa helposti hyvin suureksi.

Ellei edellä esitettyjen periaatteiden mukaisesti arvioinnin kohteena olevaa järjestelmää tarkastellen löydy sopivaa tapaa osittaa järjestelmä, voi myös tunnistettuja muutospaineita käyttää apuna osituksen luomisessa. Aloitetaan järjestelmän arviointi tarkastellen järjestelmää yhtenä kokonaisuutena. Aina kun tunnistetaan muutospaine, jonka voimakkuutta ei ole luontevaa kuvata yhdellä arvolla, jaetaan arvioinnin kohde (tai jatkossa vastaavasti jokin osista) osiin siten, että muutospaineelle voidaan kuhunkin osaan kohdistuen luontevasti antaa yksi arvo.

Ositukseen lisätään aina käynnistysvaiheessa "Koko järjestelmä" ja arvioinnin kuluessa tiettyjä erikoistapauksia, mikäli tunnistettu muutospaine ei syystä tai toisesta kohdistu erityisesti mihinkään valitun osituksen osista. Nämä tapaukset kuvataan tarkemmin seuraavassa luvussa.

4.3.2 Tehtävä 2: Pisteytys

Pisteytys muodostuu muutospaineiden tunnistamisesta, nimeämisestä ja voimakkuuden arvioimisesta käyttäen luvussa 4.2 (s. 19) kuvattua luokka-asteikkoa. Tunnistetut muutospaineet lisätään arviointimatriisiin (ks. taulukko 4.1) vastaavan tekijän alle. Muutospaineen voimakkuus arvioidaan asiantuntijoiden, mahdollisuuksien mukaan konsensukseen perustuvaan, näkemykseen pohjautuen (vrt. V3) annetulla luokka-asteikolla. Arvio annetaan jokaisesta muutospaineesta osakohtaisesti.

Tekijä			
Muutospaine	UI	DB	Logiikka
Liiketoimintaympäristö			
Rekisterilainsäädännön uudistus	0	5	0
Toteutuksen laatu			
Kovakoodatut parametrit	1	0	1
Rajapintojen ohitus	2	0	0

Taulukko 4.1: Arviointimatriisin muoto

Pisteytyksessä voidaan edetä tekijöittäin, osioittain tai vapaasti. Tekijöittäin etenemistä suositellaan, koska tekijät ohjaavat vahvasti käytävän keskustelun aihealuetta. Lisäksi tekijöittäisellä etenemistavalla on yhteismitallisuuden kannalta joitakin hyviä piirteitä. Saman järjestelmän myöhemmissä arvioinneissa, eli uudelleenarvioinneissa tulisi käytetty etenemistapa mahdollisuuksien mukaan säilyttää.

Pisteytysperiaatetta voidaan havainnollistaa seuraavan esimerkin kautta. Ajattele itsesi arvioinnin kohteeksi (unohdetaan hetkeksi ositus) ja tunnistetuksi, parasta aikaa arvioitavaksi, muutospaineeksi lämpötila. Kysymys kuuluu: "Aiheuttaako nykyinen lämpöti-

la tarpeen muuttaa jotain?" Jos lämpötila on ihan sopiva, on muutospaine "olematon" (0). Jos lisäksi aurinko paistaa niin kirkkaasti, että tarvitsisit välttämättä aurinkolasit, on lämpötilan muutospaine edelleen 0. Toki tunnistimme uuden muutospaineen "kirkas valo", mutta se on eri muutospaine se. Jos lämpötila on liian kylmä (ja tarvitset lisää vaatetta) tai liian kuuma (ja tarvitset juotavaa), on muutospaine jotain enemmän kuin 0. Erityisesti muutospaineen pisteytetty voimakkuus ei ota kantaa siihen onko tunnistettu muutospaine jossain mielessä hyvä tai huono asia.

Annettujen arvioiden perusteet kirjataan muistiin vähintään korkeimpien muutospaineiden osalta. Mieluiten perustelujen kirjaus tehdään kuitenkin kaikkien niiden muutospaineiden osalta, joiden suhteen ei ole erityistä syytä olettaa pisteytyksen perustelun olevan pelkän muutospaineen nimeämisen perusteella itsestään selvää niille, jotka arviota myöhemmin tulevat hyödyntämään.

Muutospaineet pyritään tunnistamaan samalla abstraktiotasolla. Kun jokin muutospaine kirjataan osalle tai tekijälle tietyllä (abstraktio)tasolla, ei samalle osalle eri tekijälle tulisi kirjata suoraa kausaalista aiheuttajaa tai seurannaisvaikutusta. Tällä ei ole sinänsä vaikutusta arviointituloksena saatavaan muutospaineeseen, mutta arvioinnin tulosraportti saattaa muodostua vaikeaselkoiseksi jos muutospaineita kirjataan suuri määrä eri abstraktiotasoilla. Seuraukset, mikäli ne ovat merkittäviä asioita, tulee kirjata arvioinnin perusteluosaan kyseisen muutospaineen kohdalle. Muutospaineita käsitellään yksittäisinä asioina, ei hierarkioina tai ketjuina.

Tunnistetun muutospaineen kohdistuessa koko järjestelmään, pidetään muutospaineen kohteena kaikkia järjestelmän osia. Jos tunnistetun muutospaineen ei kuitenkaan voida ajatella kohdistuvan millekään järjestelmän osalle, merkitään muutospaineen tason mukaan annetut pisteet erityisille osille riippuen muutospaineen luonteesta. Jos muutospaine kohdistuu koko arvioinnin kohteeseen kokonaisuutena, merkitään muutospaine kohdistuvaksi alussa lisätylle osalle, jonka nimi on "Koko järjestelmä". Jos taas luonteva reagointi tunnistettuun muutospaineeseen olisi uuden osan lisääminen, lisätään ositukseen vastaava uusi osa, ns. näennäisosa (engl. *virtual segment*). Muutospaine merkitään kohdistuvaksi siihen. Uusi osa nimetään samoilla periaatteilla kuin muutkin osat ja nimeen lisätään "Uusi: " -etuliite erotukseksi muista osista.

Periaatteessa on mahdollista, että osituksen osille kohdistumattomia paineita on muitakin kuin kahta edellä mainittua. Menetelmän kehitystyön yhteydessä sellaisia ei kuitenkaan ole tunnistettu, eikä niitä ole tullut vastaan myöskään soveltuvuusvalidoinnin yhteydessä.

Mikäli tunnistetaan muutospaine, joka ei ole minkään konkreettinen ilmentymä, lisätään arviointimatriisin tekijäluetteloon uusi tekijä. Se valitaan siten, että äsken tun-

nistettu muutospaine on kyseisen tekijän ilmentymä ja että tekijän abstraktiotaso ja laajuus vastaa muita luettelon tekijöitä. Nämä uudet tekijät (engl. *emergent factor*) kirjataan muistiin.

Mikäli oletustekijäluettelossa on sellaisia tekijöitä, joilla ei arvioinnin kohteen kontekstissa voi olla konkreettisia ilmentymiä, kirjataan nämä tekijät merkityksettömiksi (engl. *irrelevant*). Tämä on eri asia kuin se, ettei tunnisteta yhtään arvioitavaan järjestelmään kohdistuvaa muutospainetta, joka olisi kyseisen tekijän konkreettinen ilmentymä. Tekijä voi olla merkityksetön esimerkiksi siksi, että se viittaa kokonaisuuteen, jota ei tarvitse mitenkään huomioida arvioinnin kohteen kontekstissa. Kuvitteellinen tekijä "Osakkeenomistajien näkemys" olisi julkishallinnon kontekstissa merkityksetön. Yrityskontekstissa se ei olisi merkityksetön, vaikkei yhtään kyseistä tekijää ilmentävää muutospainetta tunnistettaisikaan.

4.3.3 Tehtävä 3: Laskenta

Arvioitavan järjestelmän muutospaineindikaattori (myöhemmin p) on kaikessa yksinkertaisuudessaan pisteytyksessä annettujen arvojen maksimi. Vastaavasti osiokohtainen muutospaineindikaattori on pisteytyksessä muutospaineille kyseistä osiota koskien annettujen pisteytysten maksimi.

Formalisoidaan tämä käyttäen matemaattista mittateoriaa ja ulkomitan käsitettä [11]. Olkoon X "ohjelmistoavaruus", Σ sen σ-algebra ja $\mu : \Sigma \to [0, \infty]$. Olkoon $\mu'(A, U) = \max\{p(a, b) | a \in A, b \in U\}$, jossa $A \subset X$ on arvioinnin kohde, U on kaikkien muutospaineiden joukko, $p(x, y) \in \{0, \ldots, 5\}$ muutospaineen voimakkuus $y \in P$ pisteessä $x \in A$. Määritellään muutospaineindikaattori p arvioinnin kohteelle A (oli se sitten järjestelmä, jokin osa tai kokonaisuus) seuraavasti

$$p = \mu(A) = \mu'(A, U). \tag{4.1}$$

Osaan S kohdistuvan muutospaineen voimakkuus muutospaineen u suhteen on $\mu^*(S, \{u\})$. Sovitaan lisäksi, että $\mu'(\emptyset, U) = 0$. Olkoon nyt $A \subset B$.[△]

$$\begin{aligned} & \mu(A) \\ = \ & \mu'(A, U) \\ = \ & \max\{p(a, b) | a \in A, b \in U\} \\ \triangle \ \leq \ & \max\{p(a, b) | a \in B, b \in U\} \\ = \ & \mu'(B, U) \end{aligned}$$

$$
\begin{aligned}
&= \mu(B) \\
&\therefore \quad A \subset B \Rightarrow \mu(A) \leq \mu(B)
\end{aligned} \tag{4.2}
$$

Myös numeroituva sub-additiivisuus pätee. Olkoon $i_{max} \in \mathbb{N}$, jolle pätee $\mu(A_{i_{max}}) = \max\{\mu(A_i)|i \in \mathbb{N}\}$.$^{\square}$ Nyt mille tahansa jonolle $\{A_i\}_i$

$$
\begin{aligned}
& \mu(\bigcup_{i\in\mathbb{N}} A_i) \\
= \;& \max\{p(a,b) \,|\, a \in \bigcup_{i\in\mathbb{N}} A_i, b \in U\} \\
= \;& \max\{\max\{p(a,b) \,|\, a \in A_i, b \in U\}|i \in \mathbb{N}\} \\
= \;& \max\{\mu(A_i)|i \in \mathbb{N}\} \\
\square \;= \;& \mu(A_{i_{max}}) \\
\leq \;& \mu(A_{i_{max}}) + \sum_{i\in\mathbb{N}\setminus i_{max}} \mu(A_i) \\
= \;& \sum_{i\in\mathbb{N}} \mu(A_i)
\end{aligned} \tag{4.3}
$$

Nyt μ on monotoninen (ks. 4.2), numeroituvasti sub-additiivinen (ks. 4.3) ja tyhjälle joukolle nolla. Siten μ on ulkomitta. Helposti nähdään, että määrittääkseen $\mu(A)$:n riittää tunnistaa yksi muutospaineista $u_m \in U$, jolle $\exists x \in A$ siten, että $\mu(A) = p(x, u_m)$. Käytännössä ei tietystikään ole, arviointia tekemättä, etukäteen mahdollista selvittää tuota muutospainetta, joten on tunnistettava ne muutospaineet $u \in U$, joille $\exists x \in A$ siten, että $p(x, u) > 0$.

Ulkomitta merkitsee tässä tapauksessa mittauksen kohteen "koon" ylärajaa. Muutospaineindikaattori $p = \mu(A)$ kertoo kuinka iso ongelma arvioinnin kohteena olevan järjestelmän sisälle mahtuu piiloon. Tämä on täsmälleen se, mitä halutun johtopäätöksen (ks. V9) tekeminen edellyttää. Mittaominaisuuden ja sitä kautta numeroituvan additiivisuuden vaatiminen mutkistaisivat MODESTia merkittävästi ja vaatimusta käytön keveydestä (V0) olisi mahdoton täyttää.

4.3.4 Arviointitulos

Arviointiraportin tulososa (jota käytetään arviointituloksen raportointiin) sisältää seuraavat tiedot.

- Arvioinnin kohteen tunniste

- Arviointi-istunnon tiedot
  - Paikka, aika ja kesto
  - Kutsutut ja osanottajat
- Tekijäluettelo, mukaan lukien lisätyt uudet tekijät
- Käytetty etenemisjärjestys
- Ositus mukaan lukien näennäisosat ja mahdollinen erikoistapaus "Koko järjestelmä"
- Tunnistetut muutospaineet, niiden nimet ja osiokohtaiset pisteytykset
- Osakohtaiset muutospaineindikaattorit ja kokonaisindikaattori
- Pisteytysten perustelut
- Mahdollisia osituksen, pisteytyksen määrittelyn ja tekijöiden tulkintaepäselvyyksiä koskevat huomiot ja epäselvyyksien ratkaisut

Arviointiraportin palauteosa (jota käytetään arviointisession raportointiin menetelmän kalibroinnista vastaavalle katselmointitiimille) sisältää seuraavat tiedot.

- Arviointi-istunnon tunniste (esim. arvioinnin kohteen tunniste ja arviointiajankohta)
- Uudet ja merkityksettömät tekijät ja mahdolliset perustelut uusien tekijöiden lisäämiselle oletuslistaan ja merkityksettömien poistamiselle sieltä
- Havaitut osituksen, pisteytyksen määrittelyn ja tekijöiden tulkintaepäselvyydet ja epäselvyyksien ratkaisut
- Menetelmää ja sen käyttöä koskevat yleiset huomiot ja palaute

Arvioinnin tuloksena saatavat muutospaineindikaattori(t) kuvaavat järjestelmään kohdistuvan muutospaineen voimakkuuden ja jakauman (V4). Tässä jakaumalla viitataan pisteytysten "reunajakaumiin" osien ja tekijöiden/muutospaineiden suhteen. Jos muutospaine on pieni ($p < T_l$), voidaan tehdä se johtopäätös, että nykyinen evoluutiostrategia on toimiva, eikä uudelleenharkintaan ole tarvetta. Jos taas muutospaine on suuri ($p \geq T_u$), tiedetään ainoastaan, että edellistä johtopäätöstä *ei* voida tehdä. Johtopäätös tässä tilanteessa onkin se, että tarvitaan jatkoselvitys parhaan evoluutiostrategian löytämiseksi. Kohtalaisen muutospaineen tapauksessa ($T_l \leq p < T_u$) on hyvä pitää järjestelmän kehitystä tarkemmin silmällä, esim. tihennetyn seurannan avulla.

Arvioinnin yhteydessä kerrytetty tieto tunnistetuista muutospaineista ja niille arvioitujen voimakkuuksien perustana olevasta argumentaatiosta on tarkoitettu palvelemaan mahdollisen järjestelmän evoluutiostrategiaan kohdistuvan jatkoselvityksen pohjana. MODEST-arviointi osoittaa eksplisiittisesti ne kohdat ja näkökulmat, joihin tarkemmassa tarkastelussa tulisi erityisesti kiinnittää huomiota.

Käytettäessä arviointituloksia *pitkittäisesti* (engl. *longitudinal*), seurataan muutospaineindikaattorin p kehitystä ajan yli. Nousevaa trendiä tulisi pitää merkkinä tulossa olevasta tarpeesta tehdä evoluutiostrategian uudelleenarviointi ja mahdollisesti siihen liittyviä evolutiivisia toimenpiteitä kuten uudistaminen. Kun arviointituloksia käytetään muutospaineiden vertailuun eri arvioinnin kohteiden välillä, puhutaan *poikittaisesta* (engl. *transversal*) käytöstä. Vertailutietoa voidaan käyttää kokonaisuuksien seurantaan ja priorisointiin, sekä laajempien strategisten linjausten tukena.

4.4 Kalibrointiprosessi

Kalibrointi on tukiprosessi, joka muodostuu *arviointitulosten jälkitarkastuksesta*, eli arviointituloksille myöhemmin tehtävistä arvioinneista ja *katselmoinnista*. Katselmoinneissa käsitellään arviointitulosten tarkastuksen ja itse arviointien tuottama palaute. Tuloksena on tehtyihin havaintoihin perustuen korjattu menetelmän parametristo.

4.4.1 Tehtävä 1: Tulosten jälkitarkastus

Aina päätettäessä reagoida muutospaineisiin, jotka kohdistuvat järjestelmään, joka on ollut MODEST-arviointien kohteena, tulee sopia arviointitulosten jälkitarkastuksen ajankohdasta ja osallistujista. (vrt. V5)

Tämän tarkastuksen kohteena ovat evoluutioaktiviteetin kohteen MODEST-arviointien tulokset. Tarkastuksen tavoite on tuottaa järjestelmällistä seurantatietoa menetelmän parametriston ylläpidon tueksi.

Jälkitarkastuksessa käydään eri näkökulmista läpi kysymystä: Olisiko menetelmän antamia arviointituloksia voinut parantaa menetelmän parametreja muuttamalla ja jos olisi, niin miten? Keskeiset kysymykset ja näkökulmat esitellään tarkemmin käyttäjän oppaassa (ks. luku 5.3.1, s. 42).

4.4.2 Tehtävä 2: Parametrien katselmointi

Katselmoinnissa käydään läpi arvioinneista ja jälkitarkastuksista kertynyt palaute. Tavoitteena on sovittaa menetelmän parametristoa sopimaan siihen (muuttuvaan) toimintaympäristöön, jossa menetelmää käytetään. Raja-arvojen T_u ja T_l kalibrointiin voi käyttää mieleistään tilastollista menetelmää. Johtuen asteikon ja siten myös tulosten suurpiirteisyydestä, ei hienostuneista menetelmistä välttämättä ole vastaavaa hyötyä verrattuna kysymyksen, "Mitä raja-arvojen tulisi olla", kysymiseen. On syytä aina pitää mielessä, että menetelmän keskeisin tavoite on kuitenkin käyty keskustelu ja sen formaali ja kompakti muistiointi.

Menetelmä ei määrittele deterministisiä sääntöjä parametriston muokkaamiseen. Kaikki päätökset, joita käsiteltyyn palautteeseen pohjautuen tehdään parametriston muuttamiseksi perustuvat viime kädessä vain ja ainoastaan katselmointitiimin asiantuntijoiden harkintaan. (ks. V8)

Vaikkakin pitkällä juoksulla parametristoon tehtävät muutokset ovat välttämättömiä, tulisi kaikkiin muutoksiin, erityisesti yhteismitallisuuteen merkittävästi vaikuttaviin, suhtautua erityisellä varovaisuudella. Pitkäjänteinen konservatiivinen linja parametriston käsittelyssä on suositeltavaa. Tästä poiketen käyttöönottovaiheessa voidaan, ja on kannattavaakin, reagoida suhteellisen herkästi ja tehdä kalibrointikierroksia tiheämpään tahtiin.

Raja-arvojen muuttaminen ei vaikuta yhteismitallisuuteen, kuten eivät myöskään organisaatiotason linjamuutoksia heijastelevat vähäiset muutokset asteikon määrittelyyn ja oletustekijäluetteloon. Myöskään epäselvyyksien ja tulkinnanvaraisuuksien korjausten implementoinnissa ei kannata olla ylikonservatiivinen.

4.4.3 Kalibrointitulos

Kalibrointi tuottaa tuloksena uuden parametriston, eli oletustekijäluettelon, arviointiasteikon määrittelyn ja raja-arvot (ks. luku 4.2, s. 19).

5 Käyttäjän opas

Tässä luvussa kuvataan MODEST-menetelmän käyttökohteet ja käyttötapa.

5.1 Johdanto

MODEST on menetelmä työpajaistuntona suoritettavaan arviointiin, jossa kootaan raportiksi läsnä olevien asiantuntijoiden näkemykset siitä, millaisia muutospaineista arvioitavaan järjestelmään tällä hetkellä kohdistuu. Arviointitulos kertoo, voidaanko nykyisellä evoluutiostrategialla[1] jatkaa, vai onko strategiaa tarpeen tarkistaa tai muutospaineisiin syytä reagoida.

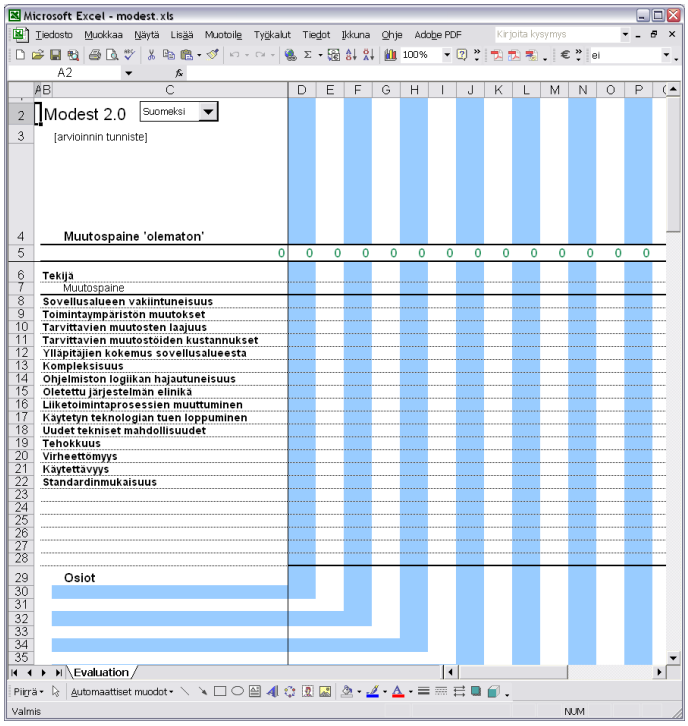
Arviointi suoritetaan joko ns. "puhtaalta pöydältä" (ensiarviointi) tai käymällä läpi edellinen arviointitulos (uudelleenarviointi). Arvioinnin kohde voi olla mikä tahansa tietojärjestelmä tai sen osa. Menetelmän käyttäjä (arviointi-istunnon vetäjä) voi olla esim. järjestelmän kehittämisestä vastaava toimittajan edustaja tai ulkopuolinen konsultti, käyttäjäorganisaatiossa järjestelmien kehittämistä koordinoiva henkilö tai erityinen arvioija. Menetelmää voi soveltaa myös itsenäisesti suoritettavaan arviointiin, mutta se on ensisijaisesti suunniteltu muutospaineita kartoittavan työpajaistunnon tueksi. Tämä ohje kattaa menetelmän istuntomuotoisen käytön.

Menetelmän kokonaisuus koostuu seuraavista osista:

- Syötedokumentit
  - Arviointiasteikon ja raja-arvojen määrittely (sisältyy menetelmään)
  - Arvioinnin kohteen ositus (syntyy arviointi-istunnossa)
  - Muutospaineiden tekijäluettelo (sisältyy menetelmään)
  - Muutospaineiden luettelo (syntyy arviointi-istunnossa)
- Arviointiraportti
  - Tulososa

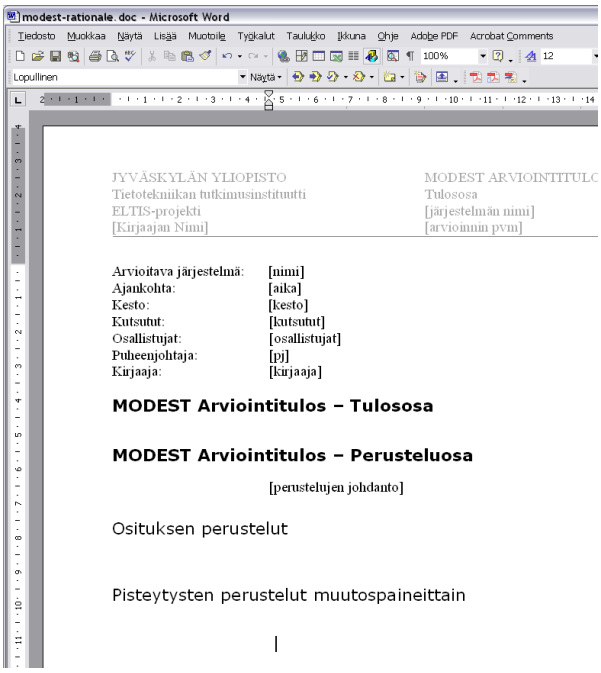
[1] Evoluutiostrategialla tarkoitetaan niitä yleisperiaatteita, joiden mukaisesti järjestelmää on tarkoitus ylläpitää ja kehittää. Evoluutiostrategia voi olla esim. ylläpito vain välttämättömät korjaukset tehden, uudistaminen perustakenteita muuttaen tai korvaaminen toisella vastaavalla järjestelmällä.

  - Perusteluosa
  - Palauteosa
- Menetelmän käyttöä tukeva materiaali
  - Käyttäjän opas (tämä luku)
  - Arviointimatriisin sisältävä arviointilomake (Excel-työkirja, ks. kuva 5.1, ei sisällytetty)
  - Tulosraportin dokumenttipohja (Word-asiakirja, ks. kuvat 5.2 ja 5.3)
- Aktiviteetit, jotka muodostavat kaksi itsenäistä prosessia
  - Arviointiprosessi (engl. *evaluation*) tuottaa varsinaisen arviointituloksen, muutospainearvion. Kuvataan tarkemmin luvussa 5.2.
  - Kalibrointi (engl. *calibration*) on tukiprosessi ja käyttää arvioinneista kertynyttä palautetta parametrien hienosäätöön. Kuvataan tarkemmin luvussa 5.3.

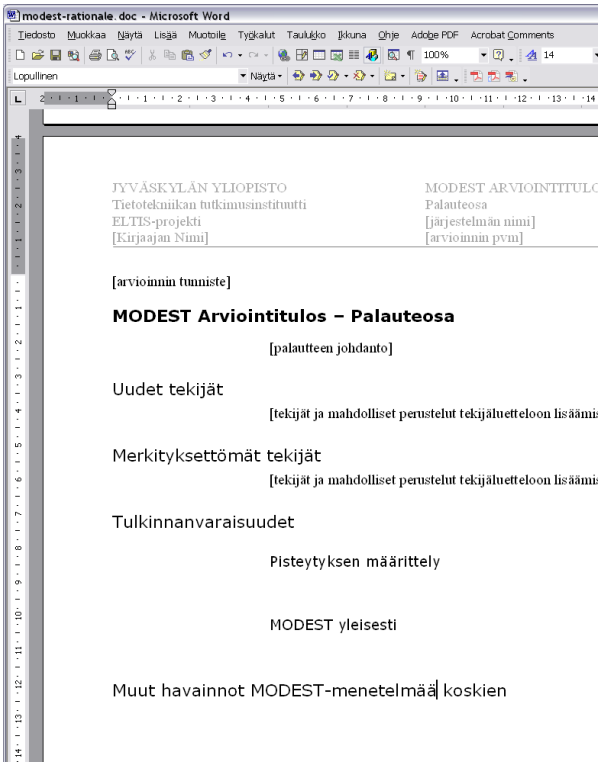


Kuva 5.1: Ruutukaappaus arviointilomakkeesta

Tuloksia voidaan käyttää pitkittäisesti tai poikittaisesti. Pitkittäisessä käytössä seurataan yhden arvioinnin kohteen muutospaineiden kehittymistä. Tavoitteena on ennakoida kehitystrendien perusteella tulevia tarpeita evoluutiostrategian tarkistukseen



Kuva 5.2: Ruutukaappaus raporttipohjan tulos- ja perusteluosasta



Kuva 5.3: Ruutukaappaus raporttipohjan palauteosasta

ja muutospaineisiin reagointiin. Poikittaisessa käytössä jäsennetään esim. järjestelmäportfolion (tai järjestelmän osien joukon) tilannetta muutospaineiden suhteen, jotta voidaan suunnata resursseja akuuteimpiin kehittämiskohteisiin.

5.2 Arviointi

Arviointi toteutetaan työpajaistuntona. MODEST-arviointi-istunnossa kootaan raportiksi läsnä olevien asiantuntijoiden näkemys siitä, millaisia muutospaineita arvioitavaan järjestelmään tällä hetkellä kohdistuu. Tässä kohdassa ohjeistetaan arviointi-istunnon järjestelyt, läpivienti ja raportointi.

5.2.1 Valmistautuminen

Arviointi-istunnossa on oltava paikalla

Puheenjohtaja joka vastaa istunnon kokoonpanosta, kutsuu istunnon koolle ja johtaa puhetta. Hän on menetelmän varsinainen käyttäjä.

Kirjaaja joka toimii istunnon sihteerinä ja viimeistelee tulosraportin. Kirjaajana voi toimia kuka hyvänsä istuntoon osallistujista.

Asiantuntija(t) jo(t)ka osallistuvat puheenjohtajan ohjaamaan keskusteluun ja esittävät näkemyksensä kirjaajan muistiin merkittäväksi.

Uudelleenarvioinnin tapauksessa on syytä ottaa huomioon edellisten arviointi-istuntojen kokoonpano sekä osallistujia valittaessa, että myöhemmin tuloksia arvioitaessa. Osallistujien valintatapa on hyvä olla sovittuna etukäteen ja dokumentoituna tulosraportin perusteluosan johdantoon.

Arvioinnin onnistuminen riippuu istunnon osallistujista. Tulosten hyödyllisyyteen vaikuttaa lisäksi arviointien yhteismitallisuus, johon palataan luvussa 5.2.8. Ryhmän kokoonpano on tasapainoilua laajapohjaisen edustuksen ja toimivan keskustelun kannalta sopivan koostumuksen löytämisessä. Arvioinnin onnistumista auttaa se, että osallistujat tietävät arviointi-istunnon päämäärän ja tuntevat riittävän hyvin etenemistavan, tulosten tulkinnan ja raportointitavan.

Jos istunnossa on paikalla voimakkaasti dominoivia henkilöitä, ei kaikkea tietämystä ehkä saada arviossa hyödynnettyä. Riskiä voidaan pienentää valmistelulla. Osallistujat

voivat tehdä muutospainearvioinnin ennen istuntoa itsenäisenä arviointina (edellyttäen että he hallitsevat menetelmän riittävän hyvin) ja toimittavat tulokset istunnon puheenjohtajalle, joka koostaa niistä anonymisoidun yhteenvedon. Arviointi suoritetaan tämän yhteenvedon pohjalta kuten uudelleenarviointi.

Istunnon puheenjohtajan tarkistuslista valmistautumiseen:

- Istunnolle on määritelty puheenjohtaja, kirjaaja ja osallistujat. Heille (ja mahdollisesti tarpeen mukaan muille) on tiedotettu valinnasta.
- Istunnossa on paikalla kaikkien arvioinnin kannalta olennaisten sidosryhmien edustus.
- Käytettävissä on rauhallinen tila.
- Käytettävissä on tarvittava välineistö (esim. dataprojektori), jotta osallistujat voivat istunnon aikana seurata tulosten kirjausta.
- Osallistujilla on käytettävissään arviointiasteikon määrittely.
- Kirjaajalla on käytössään tarvittavat asiakirjapohjat (ja niiden versio on oikea) ja osallistujilla heidän mahdollisesti tarvitsemansa tukimateriaali (esim. edellisen arvioinnin tulos, menetelmän esittelymateriaalia).
- Osallistujat tuntevat riittävällä tasolla MODEST-arvioinnin etenemisen, tavoitteen ja tulosten käytön, tai heille varataan istunnon yhteyteen aika näiden asioiden käsittelyyn.
- Osallistujat tietävät odotetaanko heiltä valmistautumista ja jos odotetaan niin mitä.

5.2.2 Istunnon läpivienti

Arviointi-istunto etenee seuraavasti:

1. Istunto käynnistyy alustuksella, jossa puheenjohtaja tarpeelliseksi näkemällään tavalla varmistaa, että läsnäolijat tuntevat riittävän hyvin arvioinnin etenemisen, tavoitteen ja tulosten käytön.
2. Alustuksen jälkeen määritellään arvioinnin kohde, nimetään ja ositetaan se.
3. Kolmannessa vaiheessa edellä tehtyä ositusta ja tekijälistaa hyväksi käyttäen tunnistetaan muutospaineet, nimetään ne ja arvioidaan ne annetulla asteikolla. Ositus ja tekijät toimivat muutospaineiden tunnistamisen apuvälineinä.

4. Lopuksi puheenjohtaja tiivistää arviointituloksen.

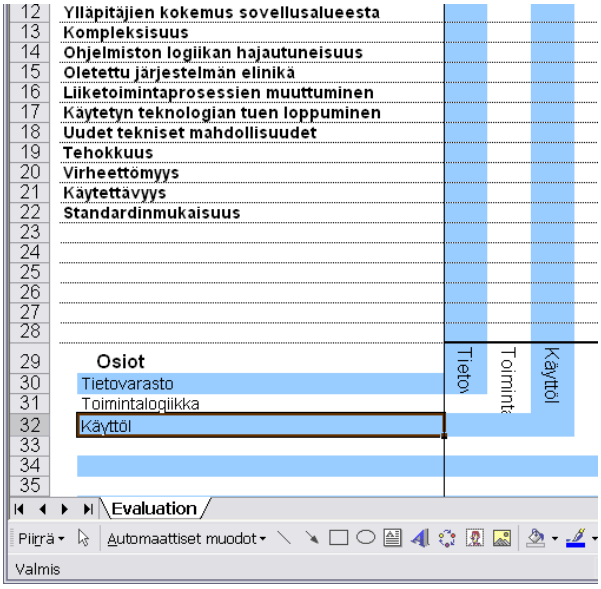
Puheenjohtajan on syytä valmistautua palauttamaan keskustelijat usein takaisin aiheeseen, koska luonteva keskustelu ei koskaan etene lineaarisesti. Rönsyily itsessään ei ole huono asia, koska se auttaa muutospaineiden tunnistamista.

Koska aiheeseen liittymätöntä mielenkiintoista keskusteltavaa tulee jatkuvasti esiin, kannattaa pitää kirjaa myös esiin nousseista myöhemmin keskusteltavista asioista. Istunnon päätteeksi tähän "villiin listaan" kirjatuista asioista sovitaan, käsitelläänkö niitä ja jos, niin missä.

Istunnon puheenjohtajan tarkistuslista istunnon ohjaamiseen:

- Ositus ja tekijät ovat sivuseikkoja. Pääasia on muutospaineiden tunnistaminen.
- Älä rajoita keskustelua, ellei se ole tarpeellista. Rajoita mieluummin puheen määrää kuin aiheita.
- Ohjaa keskustelua ensisijaisesti muistuttamalla arviointi-istunnon päämäärästä.
- Kirjauta muistiin kaikki esille tulleet myöhempää huomiointia edellyttävät asiat.

5.2.3 Arvioinnin kohteen ositus



Kuva 5.4: Ruutukaappaus osituksen kirjaamisesta arviointilomakkeelle

Arvioinnin kohde jaetaan jäsentämisen helpottamiseksi osiin ja osat nimetään mahdollisimman lyhyellä, kuvaavalla ja yksikäsitteisellä tavalla. Ositukseen lisätään aina

osa “Koko järjestelmä”. Sen käytöstä tarkemmin seuraavassa luvussa. Ositus merkitään muistiin arviointilomakkeelle (ks. kuva 5.4). Ositus voi perustua esim. arkkitehtuuriin, teknologioihin, toiminnallisuuksiin tai liiketoimintaprosesseihin. Tärkeätä on se, että jako on arvioijille luonteva.

Varmista, että

- Ositus kattaa arvioinnin kohteen ja vain arvioinnin kohteen.
- Osat eivät ole liian tulkinnanvaraisia.
- Osissa ei ole huomattavaa päällekkäisyyttä.
- Osien määrä on sopiva, hyvä nyrkkisääntö on $7(\pm 2)$ [31].

Osituksen tekoon ei kannata juuttua. Ositus on arvioinnin lopputuloksen kannalta sivuseikka kunhan vaatimukset täyttyvät.

5.2.4 Muutospaineiden tunnistaminen ja nimeäminen

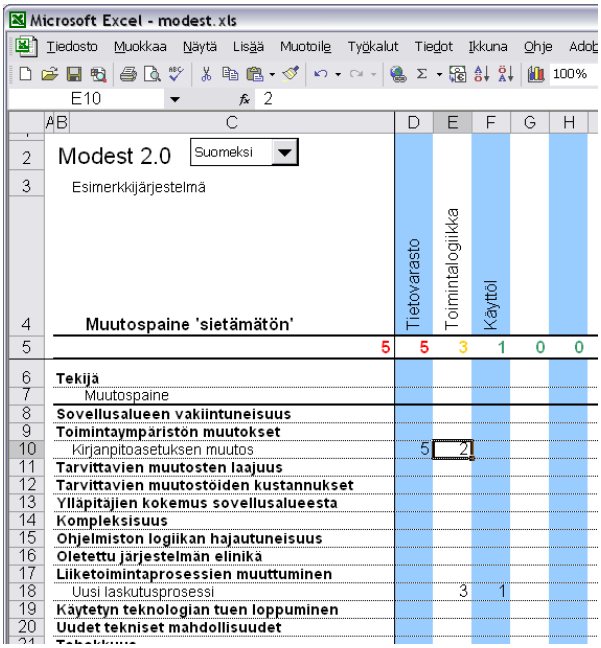
Osituksen jälkeen käydään arviointilomakkeelta löytyvä tekijälista läpi tekijä kerrallaan ja tunnistetaan kuhunkin tekijään liittyvät muutospaineet. Tunnistetut muutospaineet nimetään mahdollisimman lyhyellä, kuvaavalla ja yksikäsitteisellä tavalla. Muutospaineet lisätään arviointilomakkeelle (ks. kuva 5.5).

Jos tunnistetulle muutospaineelle ei ole oletustekijälistassa luontevaa paikkaa, lisätään tekijälistaan sopiva uusi tekijä. Uusi tekijä merkitään myös raportin palauteosaan. Lisäksi kirjataan perustelut, mikäli arvioijien mielestä nyt lisätty tekijä olisi tarpeellista lisätä myös oletuslistaan.

Esimerkki: Oletetaan oletustekijälistaksi: virheettömyys, tehokkuus, kompleksisuus ja käytettävyys. Jos nyt tunnistettaisiin muutospaineeksi henkilötietolain muuttuminen, pitäisi lisätä uusi tekijä, koska henkilötietolain muutos ei ole virheettömyydestä, tehokkuudesta, kompleksisuudesta tai käytettävyydestä johtuva asia. Uusi tekijä voisi olla esim. lainsäädäntö.

Jos tunnistettu muutospaine on jonkun jo tunnistetun muutospaineen seurausvaikutus, sitä ei kirjata erillisinä muutospaineina. Seuraukset kirjataan perusteluosaan syynä olevan muutospaineen kohdalle. Näin kaikki olennainen tulee raportoitua, mutta arviointitulos pysyy riittävän yksinkertaisena.

Mikäli tekijöitä läpikäytäessä joku tai jotkut niistä osoittautuvat arvioinnin kohteen näkökulmasta merkityksettömiksi, tämä merkitään muistiin palauteosaan. Huomaa,



Kuva 5.5: Ruutukaappaus muutospaineiden kirjaamisesta arviointilomakkeelle

että jos tekijää vastaavia muutospaineita ei tunnisteta, ei se tarkoita sitä, että tekijä olisi merkityksetön. Merkityksetön se on tämän menetelmän mielessä siinä tapauksessa, että ei ole missään olosuhteissa käytännössä mahdollista, että kyseistä tekijää vastaavia muutospaineita voitaisiin tunnistaa.

5.2.5 Muutospaineiden voimakkuuden arviointi

Tunnistettujen ja nimettyjen muutospaineiden voimakkuudet arvioidaan osakohtaisesti seuraavaa asteikkoa käyttäen. Arviot esittävät asiantuntijoiden näkemyksen muutospaineista. Niiden ei tarvitse perustua mihinkään aiempiin laskelmiin tai arvioihin. Arviot merkitään muistiin arviointilomakkeelle (ks. kuva 5.5).

olematon (engl. *none*), 0 Muutoksiin ei ole syytä.

havaittava (engl. *noticeable*), 1 Tekijän suhteen on pieni, mutta havaittava, muutospaine.

Kaikki muutospaineiksi tunnistetut asiat ovat vähintään tällä tasolla. Tämä tasoiset muutospaineet ovat yleensä pieniä huomioita tai hyvin yleisiä ja epämääräisiä.

Esimerkki: Yksi arvioijista on saanut arvioitavan järjestelmän käyttäjiltä palautetta siitä, että valikkotoiminnot on järjestetty eri tavalla kuin muissa heidän

käyttämissään sovelluksissa ja että yhdenmukaisuus voisi olla kiva asia.

selvä (engl. *apparent*), 2 Ongelmia muutospaineen suhteen on silloin tällöin ja ne on yleisesti tunnustettu.

Muutospaineet, jotka tulevat säännöllisesti esiin keskusteluissa, kuuluvat vähintään tälle tasolle. Olennainen ero edelliseen on muutospaineen yksilöinti ja sen tunnistamisen laajuus.

Esimerkki: Yksi arvioijista on saanut arvioitavan järjestelmän käyttäjiltä palautetta siitä, että eräs tietty valikkotoiminto sijaitsee epäintuitiivisessa paikassa ja sen löytäminen tuottaa monille hankaluuksia.

ongelmallinen (engl. *troubling*), 3 Ongelmat muutospaineen suhteen on otettava huomioon päivittäisessä työskentelyssä.

Muutospaineet, joihin reagoimattomuus on huomioitava työn organisoinnilla tai muilla järjestelyillä, ja joilla ei ole oleellista resurssivaikutusta, kuuluvat tälle tasolle. Olennainen ero edelliseen on muutospaineen kiistattomasti osoitettavissa olevat konkreettiset vaikutukset työskentelyyn.

Esimerkki: Arvioitavan järjestelmän käyttäjien on muistettava tehdä tarvitsemansa raportoinnit ennen iltapäivän eräajoja, joiden aikana järjestelmä on suljettu. Suoranaista lisätyötä sulkuajat eivät kuitenkaan aiheuta. Muutospaine (joka voisi olla nimetty "eräajoista johtuva alentunut saavutettavuus") olisi tällöin tasolla 3.

haitallinen (engl. *defeating*), 4 Työaikaa menetetään säännönmukaisesti. Muutospaineeseen reagointi toisi selviä hyötyjä. Ei kuitenkaan niin merkittäviä, että investointi olisi kiistämättä kannattava.

Muutospaineet, joihin reagoimattomuus on huomioitava työn järjestelyillä, joilla on oleellisia resurssivaikutuksia, kuuluvat tälle tasolle. Olennainen ero edelliseen on muutospaineen kiistämättömästi osoitettavissa olevat resurssivaikutukset.

Esimerkki: Arvioinnin kohteena on hajautetusti ylläpidettävä rekisteri. Sen tiedon syötössä ei ole tarkistuksia. Tästä johtuen tietosisältöä on säännöllisin väliajoin katselmoitava ja korjattava. Tarkistusten lisäämisen odotettavissa olevat vaikutukset (järjestelmän muutoskustannukset ja vaikutukset ylläpitoprosessiin) olisivat todennäköisesti suuret suhteessa riittävän säännöllisen katselmoinnin kustannuksiin. Muutospaine (joka voisi olla nimetty "eheystarkistusten puute") olisi tällöin tasolla 4.

sietämätön (engl. *intolerable*), 5 Tekijän suhteen on tai tulee lähitulevaisuudessa olemaan jatkuvia, vakavia vaikeuksia. Muutospaineeseen reagoinnin tuotos on lähes väistämättä tarvittavaa panostusta suurempi.

Esimerkki: Arvioitavan järjestelmän ylläpidossa käytetyn kehitysympäristön tuki loppuu tai muuttuu erittäin kalliiksi. Jossain vaiheessa ongelmia kohtuullisen varmasti ilmenee ja ongelmat saattavat olla äärimmäisen vaikeita, kalliita tai jopa mahdottomia kiertää tai korjata. Muutospaine (joka voisi olla nimetty "kehittimen tuen loppuminen") olisi tällöin tasolla 5.

Jos tunnistettu muutospaine ei kohdistu millekään järjestelmän osalle, merkitään muutospaineen tason mukaan annetut pisteet erityisille osille riippuen muutospaineen luonteesta. Jos muutospaine kohdistuu koko arvioinnin kohteeseen kokonaisuutena, merkitään muutospaine kohdistuvaksi osituksen yhteydessä lisätylle osalle, jonka nimi on "Koko järjestelmä". Jos taas luonteva reagointi tunnistettuun muutospaineeseen olisi uuden osan lisääminen, lisätään ositukseen vastaava uusi osa, ns. näennäisosa (engl. *virtual segment*). Muutospaine merkitään kohdistuvaksi siihen. Uusi osa nimetään samoilla periaatteilla kuin muutkin osat ja nimeen lisätään "Uusi: " -etuliite erotukseksi muista osista.

5.2.6 Arvioinnin dokumentointi

Kirjaaja kokoaa arvioinnista kolmeosaisen tulosraportin. Tulos- ja perusteluosat sisältävät itse muutospainearvion ja siihen liittyvän argumentaation. Palauteosa sisältää kalibrointiprosessin tarvitsemat palautetiedot arvioinneista.

Arviointi-istuntoon valmistauduttaessa kirjaaja alustaa tarvitsemansa asiakirjapohjat (arviointilomakkeen ja raportointipohjan). Uuden arvioinnin tapauksessa luodaan uudet asiakirjat tyhjistä asiakirjapohjista ja täydennetään niihin asianmukaiset tiedot. Uudelleenarvioinnin tapauksessa käytetään pohjana edellisen arvioinnin arviointilomaketta ja raporttia.

Kirjaajan tarkistuslista arviointiin valmistautumiseen:

- Varmista, että käytettävissäsi on oikea versio arviointilomakkeesta ja raporttipohjasta.
- Täydennä perustiedot: arvioinnin kohde, puheenjohtajan nimi, kirjaajan nimi, ajankohta ja kutsutut.

- Uudelleenarvioinnin tapauksessa tyhjennä perusteluosasta kohta “Muut havainnot MODEST-menetelmää koskien”.

Itse arviointi-istunnon aikana olisi hyvä, että kaikki osallistujat pystyisivät seuraamaan kirjaamista. Tämä voidaan järjestää esim. dataprojektorilla tai piirtoheittimellä.

Arviointi-istunnon kuluessa kirjaaja merkitsee ensin arviointilomakkeelle osituksen (tekijälistan alapuolella) ja sen jälkeen muutospaineet ja niitä vastaavat osakohtaiset pisteytykset sekä mahdolliset uudet tekijät. Muutospaine lisätään lisäämällä halutun tekijän alapuolelle arviointimatriisiin uusi rivi ja kirjoittamalla riville muutospaineen nimi. Tekijä lisätään kirjoittamalla sen nimi tekijälistan perään ensimmäiselle tyhjälle riville.

Raporttipohjaan kirjaaja täydentää arvioinnin kuluessa perusteluosaan ositusta koskevat täsmennykset ja tarvittaessa (ainakin korkeiden muutospainetasojen tapauksessa) pisteytysten perustelut muutospaineittain. Palauteosaan kirjaaja merkitsee muistiin kaikki arvioinnin aikana lisätyt ja merkityksettömäksi todetut tekijät. Vastaavasti merkitään muistiin myös mahdolliset perustelut tekijöiden oletustekijäluetteloon lisäämiselle tai siitä poistamiselle. Lisäksi kirjataan tulkinnanvaraisuudet (menetelmässä, arviointiasteikossa, osituksessa, tekijöissä) ja niiden ratkaisut sekä mahdolliset muut istunnon osanottajien menetelmän kalibrointia ajatellen relevantiksi kokemat asiat.

Arviointitulos kirjataan raportin tulososaan. Tämä tapahtuu helpoiten kopioimalla arviointilomakkeen täytetty osa raportin tulososaan.

Kirjaajan tarkistuslista arviointi-istuntoon:

- Täydennä/päivitä istunnon alussa 1) loput taustatiedot ja 2) tarvittaessa perusteluosan johdanto.
- Merkitse muistiin aina
  - Ositus (arviointilomake)
  - Muutospaineiden nimet (arviointilomake, lisää rivi)
  - Muutospaineiden pisteytykset (arviointilomake)
- Merkitse muistiin tarvittaessa
  - Istunnon osallistujien valintatapa ja muut mahdolliset istunnon koostumukseen vaikuttaneet asiat.
  - Pisteytysten perustelut (raportin perusteluosa)
  - Uudet tekijät mahdollisine lisäysperusteluineen (raportin palauteosa) (arviointilomake, listan jatkoksi)

  - Merkityksettömäksi todetut tekijät mahdollisine poistoperusteluineen (raportin palauteosa)
  - Havaitut tulkinnanvaraisuudet ja käytetyt ratkaisut (raportin palauteosa)
- Istunnon lopuksi merkitse muistiin kesto.

5.2.7 Tulos

Muutospaine on maksimi kaikista annetuista pisteytyksistä. Osiokohtainen muutospaine on maksimi kaikista kyseiselle osiolle kohdistetuista pisteytyksistä. Jos muutospaine on vähintään alaraja-arvon verran (ongelmallinen, 3), tulee järjestelmän tilaa seurata tavanomaista tarkemmin. Jos muutospaine on vähintään yläraja-arvon verran (sietämätön, 5), tulisi järjestelmän evoluutiostrategia ja muutospaineisiin reagointi ottaa viivyttelemättä arvioitavaksi.

Suurempaa arviointien joukkoa voidaan lisäksi käyttää pitkittäiseen ja poikittaiseen vertailuun. Niistä on kerrottu luvussa 5.1.

5.2.8 Yhteismitallisuuden säilyttäminen

Arviointitulosten hyödyllisyyteen vaikuttaa arviointien yhteismitallisuus. Se tulisi huomioida arviointien suunnittelussa, läpiviennissä ja menetelmän kalibroinnissa.

Tärkeimmät asiat ovat yhteismitallisuus yhden arvioinnin sisällä eri osien välillä ja saman arvioinnin kohteen eri arviointien välillä. Mikäli tämä on kunnossa, voidaan menetelmää käyttää pitkittäisesti, arvioinnin kohteen muutospaineiden kehittymisen seurantaan. Arvioinnin tekijäkohtaisella etenemistavalla minimoidaan arvioinnin aikana tapahtuneiden muutosten (esim. arvioinnin osanottajia saapuu tai lähtee kesken arviointitilaisuuden) vaikutus arvioinnin sisäiseen yhteismitallisuuteen. Pitkittäisen yhteismitallisuuden takaamiseksi menetelmän parametrien (oletustekijät, asteikko, ositus) on pääsääntöisesti oltava eri arvioinneissa samat. Poikittainen yhteismitallisuus edellyttää sitä, parametrit ositusta lukuun ottamatta (oletustekijät, asteikko) ovat samat. Tulkinnanvaraisuuksien ratkomiseen tähtäävät muutokset niihin ovat luonnollisesti hyväksyttäviä. Pitkittäisen yhteismitallisuuden takaamiseksi arvioivan asiantuntijaryhmän koostumuksen ja arviointi-istunnon puheenjohtajan olisi hyvä olla eri arvioinneissa samat. Poikittaisessa tapauksessa arvioivan asiantuntijaryhmän koostumuksen tulisi olla eri arvioinneissa profiililtaan samankaltainen (yhtä kattava, vastaavan asiantuntemuksen edustus).

Tarkistuslista yhteismitallisuuden huomioimiseksi tulosten pitkittäistä käyttöä ajatellen:

- Etene arvioinnissa tekijä kerrallaan.
- Säilytä parametrit (oletustekijät, asteikko, ositus) samoina. Poikkeukset: uusia tekijöitä ja näennäisosia voi lisätä (kunhan aiemmat eivät muutu) ja tulkinnanvaraisuuksia ratkoa.
- Pyri pitämään arviointi-istunnon kokoonpano suhteellisen vakaana eri arviointien välillä.

Tarkistuslista yhteismitallisuuden huomioimiseksi tulosten poikittaista käyttöä ajatellen:

- Oletustekijät ja asteikko samat eri arvioinneissa. Poikkeukset: uusia tekijöitä voi lisätä (kunhan aiemmat eivät muutu) ja tulkinnanvaraisuuksia ratkoa.
- Pyri saamaan eri arviointeihin profiililtaan riittävän samankaltainen ja kattavuudeltaan vastaava asiantuntijaryhmä.

5.3 Kalibrointi

Kalibrointi (engl. *calibration*) on tukiprosessi, joka käyttää arvioinneista kertynyttä ja tuloksia jälkeenpäin arvioimalla koottua palautetta menetelmän parametrien hienosäätöön.

5.3.1 Tulosten jälkitarkastus

Arviointien tulokset tulee aina ottaa jälkitarkastukseen siinä vaiheessa kun niitä on mahdollista arvioida "jälkiviisaasti". Aina kun päätetään reagoida muutospaineisiin, tulee tarkastuksen ajankohdasta ja osallistujista sopia. Tarkastuksen kohteena ovat evoluutioaktiviteetin kohteen MODEST-arviointien tulokset. Jonkun tarkastuksen osallistujista olisi hyvä olla itse arviointeihin osallistunut henkilö.

Katselmoinnissa käydään läpi ja merkitään muistiin seuraavat asiat:

- Oliko arviointitulosten perusteella jossain vaiheessa pääteltävissä muutostarve, mikä oli kyseinen ajankohta? Jos ei, miksi?

- Oliko arviointitulosten perusteella riittävän ajoissa pääteltävissä tuleva tarve reagoida muutospaineisiin? (Antoiko MODEST käyttötarkoituksensa mukaisen "ennakkovaroituksen"). Jos ei, miksi?
- Mikä olisi tarkastajien näkemyksen mukaan ollut ajankohta, jolloin arviointitulosten perusteella olisi pitänyt pystyä havaitsemaan muutostarve?
- Olisiko menetelmän tuottamia arviointituloksia voinut parantaa menetelmän parametreja muuttamalla? Jos, niin mitä?

5.3.2 Parametrien kalibrointi

Parametrien kalibrointi tulisi tehdä säännöllisesti sen mukaan kuinka paljon käsittelyä vaativia arviointien palauteosia ja tulosten jälkitarkastusraportteja kertyy. Menetelmän käyttöönottovaiheessa kalibrointia on suotavaa tehdä varsin tiheästi, esim. aina noin 5–10 arviointien palauteosan ja jälkitarkastusraportin kertymisen jälkeen. Pitkällä tähtäimellä kalibrointia on hyvä tehdä noin 6–18 kk välein.

Kalibrointiin tulisi ottaa osaa vähintään jonkun arviointien puheenjohtajana toimineen sekä jonkun menetelmän ylläpidosta ja koulutuksesta vastaavan.

Kalibroinnissa käydään ensiksi läpi edellisen kalibroinnin jälkeen kertyneet arviointien palauteosat ja tulosten jälkitarkastusraportit. Palautteista tehdään kooste arviointiasteikon määrittelyn, tekijäluettelon, raja-arvojen sekä lopuksi menetelmän ja sen ohjeistuksen näkökulmasta. Lopuksi tarkastellaan koostetta ja tarpeen mukaan aiempia vastaavia koosteita ja muutoslokia ja arvioidaan voitaisiinko menetelmää kehittää sen ohjeistusta ja parametreja korjaamalla. Tarpeellisiksi koetut muutokset implementoidaan menetelmään ja kuvaus muutoksista perusteluineen kirjataan menetelmän muutoslokiin.

Konkreettinen lopputulos kalibroinnista on päivitetty käyttäjän opas ja arvioinnin tukimateriaali.

5.4 Pikaohje

Tässä on avainsanoin tiivistettynä MODEST-arvioinnin läpivientiin liittyvät keskeiset asiat.

Istunnon eteneminen

1. Alustus
2. Arvioinnin kohteen osittaminen
3. Muutospaineiden
   - tunnistaminen,
   - nimeäminen ja
   - voimakkuuden arviointi
4. Loppuyhteenveto

Muista nämä

- Ennen arviointi-istuntoa varmista että:
  - Istunnolla on puheenjohtaja, kirjaaja, asiantuntijat, paikka ja ajankohta.
  - Asianosaisia on informoitu.
  - Istunnon aikana on käytössä tarvittava välineistö (jotta kirjausta voi seurata reaaliajassa) ja tukimateriaali.
  - Istunnon tiedot on kirjattu dokumenttipohjiin.
- Arviointi-istunnon aikana varmista että:
  - Muutospaineiden tunnistaminen ja arviointi on pääasia.
  - Istunnon tiedot on kirjattu.
  - Ositus kattaa arvioinnin kohteen, eivätkä osat ole päällekkäisiä.
  - Muutospaineet ja osat on nimetty yksikäsitteisesti ja kuvaavasti.
  - Näennäisosilla on etuliitteenä “Uusi: ”.
  - Ainakin korkeimmat pisteytykset on perusteltu.
  - Palauteosan asiat on kirjattu.
  - Riittävä yhteismitallisuus on huomioitu.
- Arviointi-istunnon jälkeen
  - Kirjaa istunnon kesto ja viimeistele raportti.
  - Arkistoi raportin tulososa ja palauteosa asianmukaisesti.

Asteikko

0 olematon (engl. *none*) Ei syytä muutoksiin

1 havaittava (engl. *noticeable*) Pieni mutta havaittava muutospaine

2 selvä (engl. *apparent*) Muutospaine on yleisesti tunnustettu

3 ongelmallinen (engl. *troubling*) Muutospaine aiheuttaa järjestelyjä

4 haitallinen (engl. *defeating*) Muutospaine aiheuttaa työajan menetystä

5 sietämätön (engl. *intolerable*) Muutospaine aiheuttaa vakavia ongelmia

Tavallisia ongelmatilanteita

Seuraavassa on kuvattu joitakin tavanomaisimpia ongelmatilanteita ja se, kuinka niissä on suositeltavaa toimia, jotta arviointi etenee tavoitteiden mukaisesti.

Asia X on tulkinnanvarainen. Sovi ratkaisu. Kirjaa tulkinnanvaraisuus ja ratkaisu raportin palauteosaan asianmukaiseen kohtaan.

Muutospaineelle ei löydy sopivaa tekijää. Lisää sopiva tekijä listan jatkoksi. Lisää muutospaine uuden tekijän alle. Kirjaa lisätty tekijä palauteosaan asianmukaiseen kohtaan.

Tekijä X tässä kontekstissa järjetön. Merkitse X raportin palauteosaan merkityksettömien tekijöiden kohtaan. Lisää perustelut, jos tekijä arvioijien näkemyksen mukaan tulisi poistaa oletuslistasta.

Muutospaine ei kohdistu mihinkään osaan. Muutospaineen kohdistuessa järjestelmään kokonaisuutena, merkitse se osalle “Koko järjestelmä” ja jos taas muutospaine kohdistuu uuden osan lisäämiseen, lisää kyseinen osa, nimeä se käyttäen etuliitettä “Uusi: ” ja kohdista paine kyseiselle osalle.

6 Soveltuvuusvalidointi

Menetelmän soveltuvuusvalidointi on tehty pilottikäytöllä ja asiantuntijakatselmoinneilla. Tavoitteena on ollut mahdollisimman kattavasti selvittää menetelmälle asetettujen vaatimusten toteutuminen.

6.1 Pilottikäyttö

Menetelmän pilottikäytössä kohteena oli 7 erillistä perinnejärjestelmää (K1–K7). Järjestelmät olivat käytössä eri organisaatioissa ja ne olivat peräisin eri toimittajilta. K1 ja K3 ovat asiakasrekistereitä, K2 sopimusten mukaisesti tapahtuvan kustannusten kohdentamisen ja laskutuksen järjestelmä, K4 erikoistunut tapauksenhallintajärjestelmä, K5 liikkeenjohdon tukityökalu ohjelmistoprosessien kehittämiseen, K6 ylläpitokohteiden kuntorekisteri ja huoltosuunnittelun tuki ja K7 sidosryhmätietojen rekisteri. Perustiedot järjestelmistä ja arvioinneista on koottu taulukkoon 6.1. Tiedot järjestelmistä ovat järjestelmiä kehittävien ja käyttävien organisaatioiden ilmoittamia. Arviointia koskevat tiedot on koottu arviointiraporteista.

Teknologialla viitataan tämänhetkiseen pääasialliseen toteutuskieleen tai kehitysympäristöön. Useiden sovellusten kohdalla on historian aikana käyty läpi eriasteisia teknologisia murroksia. Etenemisen osalta E_t merkitsee etenemistä tekijöittäin ja E_v vapaassa järjestyksessä. Ositusperusteissa O_t merkitsee toiminnallisuuksiin, O_a järjestelmäarkkitehtuuriin ja O_l liiketoimintaprosesseihin perustuvaa. johtopäätöksissä J_j on jatkoselvityksen tekeminen, J_t järjestelmän kehityksen tiiviimpi tarkkailu ja J_y ylläpidon jatkaminen kuten tähänkin asti.

Arviointeihin osallistuneilta koottiin jälkikäteen strukturoimattomalla haastattelulla palautetta menetelmää ja arviointituloksia koskien. MODEST auttoi arvioijien mukaan löytämään järjestelmän evoluution kannalta ongelmallisimmat osat (K1, K2, K3, K7). Menetelmä puki asiantuntijoiden esittämät näkemykset tiiviiseen, havainnolliseen ja hyvin kuvaavaan esitysmuotoon (K3, K4, K5, K6). Yleisesti ottaen alustusta käyttöön tarvittiin varsin vähän. Menetelmä on, siinä laajuudessaan kuin arviointi-istunnon osallistujien se tulee tuntea, nopea ja selkeä kuvailla. Menetelmä auttoi huomaamaan asioita, joita arvioijien mukaan ei muuten olisi tullut esille lainkaan (K3, K4, K6, K7).

Tiedot	K1	K2	K3	K4	K5	K6	K7
Arvioinnin kohteena oleva järjestelmä							
koko (kLOC)	-	-	⋆⋆	≈ 85	≈ 230	≈ 32	-
teknologia	COBOL	RPG	Plex⋆⋆	C++	Java	Delphi	COBOL
käyttäjiä	≈ 200	≈ 100	≈ 200	≈ 3000	≈ 30	≈ 40	≈ 10⋆
kehitetty	80-luku	89–92	05	95–98	95–01	90–92	-
investointi	-	-	-	-	-	1 M€	-
Arviointi-istunto							
kutsuttuja	3	2	7	6	2	6	3
paikalla	3	2	6	6	2	6	3
eteneminen	E_t	E_v	E_t	E_t	E_t	E_t	E_t
kesto (min)	130	45	135	100	75	150	120
MODEST versio	1.4	1.4	2.0	2.0	2.0	2.0	2.0
Arviointi							
ositusperuste	O_t	O_l	O_t	O_a	O_a	O_t	O_a
osia	13	19	15	7	7	13	12
näennäisiä	2	-	2	-	-	-	1
muutospaineita	NA	NA	20	11	11	10	12
uusia tekijöitä	1	1	1	-	-	1	-
merkityksettömiä	-	-	2	2	1	2	2
perustelu (sanaa)	63	68	449	63	92	134	58
p	5	2	4	5	4	5	4
johtopäätös	J_j	J_y	J_t	J_j	J_t	J_j	J_t

⋆ Suorien käyttäjien lisäksi palvelurajapintojen kautta käyttäjänä noin 50 sovellusta

⋆⋆ All Fusion Plex on Computer Associatesin mallinnuspohjainen sovelluskehitin, ks. `http://www3.ca.com/solutions/Product.aspx?ID=258` (viitattu 27.4.2006). Generoitua RPG-koodia (palvelinosat) järjestelmässä on 313 kLOC ja Java-koodia (asiakasosat) 21,6 Mt (näyteaineiston perusteella noin 26 kt/kLOC, eli noin 850 kLOC), joka käsittää 789 luokkaa.

Taulukko 6.1: Tiedot käyttötapauksista

Erityisesti yksi K3:n arvioijista kertoi yllättyneensä perin juurin siitä, miten näinkin yksinkertaisella keskustelun ohjauksella saatiin esiin useita sellaisia varsin keskeisiä asioita, jotka eivät aiemmin pitkän tiiviin asiakas-toimittaja -suhteen aikana olleet tulleet ilmi.

Tapauksissa K1 ja K2 vastaavaan johtopäätökseen tähtäävä arviointi- ja päätöksentekoprosessi käytiin organisaatioissa läpi myös MODEST-arviosta riippumattomasti. Tapauksessa K1 riippumaton päätös oli järjestelmän uudelleensuunnittelu ja radikaali uudistaminen (uudelleentoteutus). Tapauksessa K2 riippumaton päätös oli jatkaa järjestelmän käyttöä ja ylläpitoa totuttuun tapaan ilman jatkoselvityksiä tai muita aktiviteetteja. Molemmissa tapauksissa siis MODEST-arvioinnin tulos vastasi riippumattomasti syntynyttä johtopäätöstä evoluutiostrategiasta. Mitä sitten MODESTia käyttäen oikein voitettiin, jos kerran samaan tulokseen päädyttiin myös ilman? Organisaatiokohtainen ad-hoc selvitys vei aikaa usealta ihmiseltä useita työpäiviä kun taas MODEST-arviointiin kulunut aika lasketaan kaikissa tapauksissa tunneissa.

6.2 Asiantuntijakatselmoinnit

Asiantuntijakatselmointeja järjestettiin kaksi. Kuusi ohjelmistoteollisuuden palveluksessa olevaa kokenutta asiantuntijakonsulttia otti osaa katselmointi-istuntoihin. Asiantuntijoiksi valittiin henkilöitä, jotka kuuluvat MODESTin potentiaaliseen käyttäjäryhmään. He eivät olleet aiemmin olleet tekemisissä MODESTin kanssa. Ennen katselmointitilaisuutta asiantuntijoille toimitettiin esittelymateriaalia menetelmää koskien ja tilaisuuden alussa käytiin menetelmä yksityiskohtaisesti läpi.

Itse katselmointitilaisuudessa asiantuntijat keskustelivat menetelmästä yleisellä tasolla ja asetettujen suunnitteluvaatimusten näkökulmasta. Vastasin menetelmän kehittämisestä vastaavan ominaisuudessa asiantuntijoiden menetelmää koskeviin kysymyksiin. Asiantuntijoiden näkemykset kirjattiin muistiin. Asiantuntijat olivat yksimielisiä suunnitteluvaatimusten täyttymisestä seuraavilta osin.

- MODEST on riittävän yksinkertainen, helppokäyttöinen ja resurssivaatimuksiltaan matala, jotta se soveltuu suunniteltuun tarkoitukseen. (V0)
- MODEST on systemaattinen tapa tunnistaa ja ennakoida mahdollisia tulevia uudistamistarpeita käyttäen pitkittäistä tarkastelua aikasarjaan joko järjestelmän nykytilan arvioinneista tai evoluutioskenaarioiden arvioinneista. (V1)

- MODEST kykenee nostattamaan keskustelua ja parantamaan sidosryhmien välistä kommunikointia tarjoamalla tavan esittää järjestelmän tila (evoluutiomielessä) erittäin tiiviissä ja havainnollisessa muodossa. (V4, V6)

7 Yhteenveto

MODEST on menetelmä tietojärjestelmään kohdistuvien muutospaineiden arviointiin. Arviointi tapahtuu työpajaistuntona ja arviointikohteena on tietojärjestelmä, sen osa tai järjestelmäkokonaisuus. Arviointituloksen perusteella voidaan tehdä johtopäätös siitä, kannattaako jatkaa arvioinnin kohteen ylläpitoa nykyisen evoluutiostrategian mukaisesti vai ei. Erityisesti päätöstä evoluutiostrategian muutoksesta ei periaatteessa voi tehdä yksin MODEST-arvioinnin perusteella (vrt. luku 4.3.4, s. 26).

Tässä luvussa luodaan kokonaiskuvaa tähänastisista tuloksista ja hahmotetaan menetelmän jatkokehitysnäkymiä.

7.1 Vaatimusten toteutuminen

V0 *Muutospaineiden arviointi on oltava kevyt. Tavoitteena voidaan pitää 1 htp enimmäisresursseja yhden järjestelmän arviointiin.*

Menetelmää katselmoineet asiantuntijat olivat yhtä mieltä vaatimuksen täyttymisestä.

Pilottikäytön seitsemässä tapauksessa arviointien läpivientiin kului kokonaisuudessaan henkilötyöaikaa 55 tuntia, joka tarkoittaa noin 1,1 henkilötyöpäivää arviointia kohden. Tätä voidaan kohtuudella pitää asetetun vaatimuksen toteuttavana tuloksena, koska valtaosa pilottitapauksissa arviointi-istuntoon osallistuneista ei ennakolta tuntenut menetelmää. Pidemmällä aikajänteellä tilanne oletettavasti ei tule olemaan tämä.

V1 *Tukeakseen systematisointia MODESTin tulee olla hyvin määritelty ja jäljitettävä.*

Menetelmää katselmoineet asiantuntijat olivat yhtä mieltä yksiselitteisyyden vaatimuksen täyttymisestä. Näkemystä tukee myös pilottikäytöstä saatu palaute. Jäljitettävyysvaatimuksen täyttyminen on menetelmän rakenteesta johtuen ilmeinen.

V2 *Menetelmän tulee altistaa käyttäjänsä asiantuntijakollegoiden näkemyksille ja tarjota käyttöön sopivaa näkökulmien joukkoa tai tapaa etsiä näkökulmia.*

Arviointimatriisin dimensioista tekijät tarjoavat asiantuntijakollegoiden näkemyksen ja ositus systemaattisen tavan etsiä näkökulmia. Pilottikäytöstä saadut kokemukset osoittavat, että uusien näkökulmien haku myös käytännössä onnistuu. Neljässä tapauksessa seitsemästä arvioijien mielestä menetelmä auttoi tunnistamaan asioita, joita ei muutoin olisi havaittu lainkaan.

V3 *Menetelmän tulisi tarjota tapa formalisoida myös intuitiiviset näkemykset.*

Menetelmä on ilmeisen neutraali sen suhteen, ovatko annetut arviot ja niiden mahdolliset kirjatut perustelut johdettuja pelkästään joistain hyvin määritellyistä sisäisistä metriikoista, vai perustuvatko ne yksin intuitiiviselle asiantuntijanäkemykselle.

V4 *Menetelmän tulosten tulisi raportoitua havainnolliseen ja tiiviiseen muotoon.*

Menetelmää katselmoineet asiantuntijat olivat yhtä mieltä vaatimuksen täyttymisestä. Neljässä pilottitapauksessa seitsemästä arvioijat kuvasivat arviointituloksen esitysmuotoa havainnolliseksi.

Arviointiraportin tulososan olennainen sisältö on arviointimatriisi, lasketut indikaattorit ja kirjatut perustelut. Pilottitapauksissa järjestelmät ositettiin keskimäärin 12,3 osaan (7-19) ja muutospaineita tunnistettiin 12,8 (10-20). Arviointimatriisit olivat siten kohtuullisen kokoisia ja niitä oli helppo tarkastella kokonaisuutena. Sanallista kuvausta arviointiraportin tulososat sisälsivät keskimäärin 132 sanaa (58-449), joka lähes muotoilusta riippumatta sopii yhdelle A4-sivulle.

V5 *Menetelmän tulisi edellyttää sen tuottamien tulosten myöhempi arviointi ja antaa hyvin määritelty liipaisin sen käynnistämiseen.*

Menetelmän kalibrointiprosessin tehtävä 1, eli tulosten jälkitarkastus (ks. luku 4.4.1, s. 28) ja arviointien toistojen tapahtuminen uudelleenarviointimuodossa (ks. luku 4.3, s. 21) pitävät huolen tämän vaatimuksen toteutumisesta.

V6 *Menetelmän tulee tukea keskustelua ja viestintää sidosryhmien välillä.*

Menetelmää katselmoineet asiantuntijat olivat yhtä mieltä vaatimuksen täyttymisestä. Myös pilottikäytöstä saatu palaute tukee tätä näkemystä.

Muutospaineiden tunnistaminen ja niiden yksilöinti nimeämällä auttavat keskustelulle välttämättömän sanaston luonnissa ja yhtenäistämisessä.

V7 *Arvioinnin kohteena oleva järjestelmä tulisi jakaa osiin.*

Arviointiprosessin tehtävä 1, eli ositus (ks. luku 4.3.1, s. 22) huolehtii tämän vaatimuksen täyttymisestä. Pilottikäyttö osoittaa, että tarkoituksenmukainen ositus onnistuu menetelmässä esitetyllä tavalla.

V8 *Menetelmän tulee tukea itsensä arviointia ja muuttamista. Sen tulee kyetä muuttumaan ja mukautumaan jatkuvasti muuttuvaan toimintaympäristöönsä.*

Kalibrointiprosessi (ks. luku 4.4, s. 28) huolehtii tämän vaatimuksen täyttymisestä. Kalibrointiprosessia ei ole pilottikäytössä vielä viety läpi, joten kokemuksia sen toimivuudesta ei toistaiseksi ole käytettävissä.

V9 *Menetelmää käyttäen saadusta tuloksesta pitää olla tehtävissä johtopäätös siitä, voidaanko ylläpitoa nykyisellä evoluutiostrategialla turvallisesti jatkaa.*

Teoriassa on ilmeistä, että menetelmä täyttää tämän vaatimuksen. Vakuuttava empiirinen validointi ja selittävyyden arviointi edellyttäisi sitä, että käytettävissä olisi jälkikäteen absoluuttinen informaatio optimaalisesta evoluutiostrategiasta. Se on ongelman mutkikkuuden vuoksi käytännössä mahdotonta. Tulosten jälkitarkastuksen kautta kertyvän datan avulla jonkin asteinen empiirinen validointi lienee kuitenkin mahdollista.

7.2 Johtopäätökset

Ohjelmistojen uudistamisen alueen olemassa oleva menetelmällinen tuki ei sovellu muutospaineiden säännölliseen, pitkäaikaiseen seurantaan. Järjestelmällistä seurantaa kuitenkin tarvitaan varhaisen havaitsemisen takaamiseksi, joten menetelmälliselle tuelle on tarvetta. MODEST täyttää tämän tarpeen. Sen soveltuvuus ja käyttökelpoisuus on empiirisesti validoitu asiantuntijakatselmoinnein ja todellisilla arvioinneilla.

Kerättyyn empiriaan pohjautuen voidaan sanoa MODESTin olevan hyvinmääritelty, helppokäyttöinen ja käyttötarkoitukseensa soveltuva menetelmä.

7.3 Pohdintaa

MODESTin arviointitulos, järjestelmään kohdistuvien muutospaineiden alaraja, on ns. ulkoinen metriikka ja samoin ovat myös menetelmän oletusparametriston tekijät. Menetelmä ei ota millään tavalla kantaa siihen, mistä (jos eksplisiittisesti mistään) sisäisistä metriikoista nämä johdetaan. Tällä perusteella menetelmää voisi helposti kritisoida

pinnalliseksi. “Pinnallisuus” on kuitenkin tarkoituksellista, koska MODESTin fokus on muutospaineiden havaitsemisessa ja nimeämisessä. Pelkällä muutospaineindikaattorin arvolla on lopulta varsin vähän käyttöä ilman tunnistettuja ja nimettyjä muutospaineita ja arvioiden taustalle muistiin merkittyä argumentaatiota. Toisin päin taas ilman muutospaineindikaattorin arvoa arviointitulos on edelleen varsin käyttökelpoinen dokumentti järjestelmän evoluutiostrategian analysoinnissa tai valinnassa.

Tässä vaiheessa on varsin luontevaa kysyä, “Olemmeko ratkaisseet ongelman?” Pilottitapauksia katsottaessa ilman MODESTia järjestelmät olisi tapauksissa K1 ja K2 kuitenkin arvioitu ja tulokset olisivat olleet täsmälleen samat kuten myös tulosten pohjalta tehdyt johtopäätökset ja vastaavat päätökset toimenpiteistä. MODESTin käytöstä saatava lisäarvo syntyi näissä tapauksissa, kuten se syntyy yleisestikin, vähäisestä resurssitarpeesta ja asiantuntijanäkemysten kompaktista formaalista muistioinnista. Arviot saatiin vähemmällä työllä ja ne ovat keskenään (vähintäänkin muodollisesti) yhteismitallisia.

Miksi tätä varten sitten tarvitaan formaali menetelmä? Eivätkö arviointi-istuntoon osallistuneet asiantuntijat olisi joka tapauksessa saman tietämyksensä perusteella voineet tarvittaessa antaa aloitteen evoluutiostrategian tarkistamiseen? Periaatteessa kyllä, mutta käytännössä aloitteen tekeminen on täysin riippuvainen siitä, miten asiantuntijat saavat näkemyksensä kommunikoitua päättäjille. Juuri tässä on käytännössä havaittu systematiikan puutteesta johtuen ongelmia [25]. Ilman formalisointia asiantuntijoiden näkemykset tulevat kuulluiksi ja huomioiduksi lähinnä sattumalta. Lisäksi, kuten pilottikäytöstä saadusta palautteesta nähdään, systemaattinen menetelmä auttaa havaitsemaan asioita, jotka muuten jäisivät pimentoon.

Menetelmällä on myös tiettyjä rajoituksia. Ulkomittaan turvautuminen yksinkertaistaa arviointiprosessia, mutta ei salli pidemmälle menevien johtopäätösten tekemistä järjestelmän tilasta. Riittävän yhteismitallisuuden säilyttäminen edellyttää tarkkaavaisuutta. Pilottikäyttö ei toistaiseksi kata kalibrointiprosessia.

7.4 Jatkokehitys

Menetelmän soveltuvuutta erilaisiin organisaatioihin ja konteksteihin voitaisiin helposti kehittää kokoamalla valmiita pohjakriteeristöjä, joista organisaatiot voisivat valita pohjaksi itselleen sopivimman tai sopivimmat. Nykyisestä kriteeristöstä lähtien kalibrointiprosessi ei välttämättä kovin nopeasti konvergoi käyttäjäorganisaation näkökulmasta optimaaliseen kriteeristöön. Esimerkiksi järjestelmäportfolioiden arviointi ei ko-

vin luontevasti onnistu, koska nykyinen kriteeristö on koottu yksittäisen järjestelmän näkökulmasta.

Osien keskinäisten suhteiden ja riippuvuuksien mallintaminen muutospaineiden propagoitumisella osasta toiseen niiden keskinäisten riippuvuuksien mukaan saattaisi tehdä arviointituloksesta jatkohyödyntämisen kannalta käyttökelpoisemman. Toisaalta vapaamuotoisesti laadittavan osituksen suhteen riittävän kuvaavan riippuvuusverkon laatiminen on kaikkea muuta kuin helppo ja nopea tehtävä. Arvioinnin kohteen ositus olisi siten muotoiltava kokonaan uudestaan tukemaan tätä käyttötapaa.

Hakemisto

8 Viitteet

[1] R.S. Arnold: *Software Reengineering*, IEEE Computer Society Press, 1993.

[2] Rajiv D. Banker *ym.*: "Software complexity and maintenance costs", *Commun. ACM*, 36(11), ss. 81–94, 1993, ISSN 0001-0782.

[3] K. Bennett, M. Ramage ja M. Munro: "Decision model for legacy systems", *IEE Proceedings – Software*, 146(3), ss. 153–159, 1999.

[4] Barry W. Boehm: "Software Engineering Economics", *IEEE Transactions on Software Engineering*, SE-10(1), ss. 4–21, tammikuu 1984.

[5] Ned Chapin *ym.*: "Types of software evolution and software maintenance", *Journal of Software Maintenance and Evolution: Research and Practice*, 13(1), ss. 3–30, 2001.

[6] Norman K. Denzin ja Yvonna S. Lincoln (toim.): *The Handbook of Qualitative Research*, 2. painos, Sage Publications, 2000.

[7] A. Eastwood: "Firm fires shots at legacy systems", *Computing Canada*, 19(2), 1993.

[8] D. Edelstein: "Report on the IEEE STD 1219-1993 - Standard for Software Maintenance", *ACM SIGSOFT Software Engineering Notes*, 18(4), s. 94, 1993.

[9] Len Erlikh: "Leveraging Legacy System Dollars for E-Business", *IT Professional*, 3(2), ss. 17–23, 2000.

[10] Ilkka Haikala ja Jukka Märijärvi: *Ohjelmistotuotanto*, 8. painos, Satku, 2002. URL `http://www.cs.tut.fi/~otm/kirja/`

[11] Paul Halmos: *Measure Theory*, D. van Nostrand and Co., 1950.

[12] Michael Hammer ja Steven A. Stanton: *The Reengineering Revolution*, Harper-Business, 1995.

[13] Maarit Harsu: *Ohjelmien ylläpito ja uudistaminen*, Korkeakoulu-sarja, Talentum, 2003.

[14] Donald E. Harter, Mayuram S. Krishnan ja Sandra A. Slaughter: "Effects of process maturity on quality, cycle time, and effort in software product development", *Manage. Sci.*, 46(4), ss. 451–466, 2000, ISSN 0025-1909.

[15] S. Huff: "Information systems maintenance", *The Business Quarterly*, 55, ss. 30–32, 1990.

[16] IEEE: *IEEE Standard for Software Maintenance*, 1992, IEEE standard 1219-1992.

[17] IEEE: *Standard for Software Maintenance*, 1998, IEEE standard 1219-1998.

[18] IEEE: *Standard for Software Engineering - Software Life Cycle Processes - Maintenance*, 2006, iEEE standard 14764-2006.

[19] M. Jørgensen: "Experience with the accuracy of software maintenance task effort prediction models", *IEEE Transactions on Software Engineering*, 21(8), ss. 674–681, 1995.

[20] I. Kankaanpää *ym.*: "Challenges in IS Evolution Benefit Assessment", teoksessa "Proceedings of 9th International Conference on Business Information Systems", Springer Verlag, 2006.

[21] C. F. Kemerer ja S. Slaughter: "An empirical approach to studying software evolution", *IEEE Transactions on Software Engineering*, 25(4), ss. 493–509, 1999.

[22] J. Koskinen *ym.*: *Evaluation of Software Modernization Estimation Methods Using NIMSAD Meta Framework*, Tekninen Raportti 15/2004, Tietotekniikan tutkimusinstituutti, Jyväskylän yliopisto, 2004.

[23] J. Koskinen *ym.*: *Evaluation of the Software Evolution Options*, Tekninen Raportti 14/2004, Tietotekniikan tutkimusinstituutti, Jyväskylän yliopisto, 2004.

[24] J. Koskinen *ym.*: "Software Modernization Decision Criteria: An Empirical Study", teoksessa "Proceedings of The Ninth European Conf. on Software Maintenance and Reengineering", (ss. 324–331), IEEE Comp. Soc. Press, 2005.

[25] Jussi Koskinen *ym.*: "Empirical Study of Industrial Decision Making for Software Modernizations", teoksessa "Proc. of the 4th Int. Symposium on Empirical Software Engineering", (ss. 227–236), IEEE Comp. Soc. Press, 2005.

[26] M. M. Lehman: "Programs, Life Cycles and Laws of Software Evolution", *Proc. IEEE Special Issue on Software Engineering*, 68(9), ss. 1060–1076, 1980.

[27] M. M. Lehman *ym.*: "Metrics and Laws of Software Evolution - The Nineties View", teoksessa "Proceedings of the Fourth International Software Metrics Symposium (METRICS'97)", (ss. 20–32), 1997.

[28] M.M. Lehman ja L.A. Belady: *Program Evolution – Process of Software Change*, Academic Press, 1985.

[29] Bennet P. Lientz ja E. Burton Swanson: "Problems in application software maintenance", *Communications of the ACM*, 24(11), ss. 763–769, 1981.

[30] J. R. McKee: "Maintenance as a function of design", teoksessa "Proceedings of the AFIPS National Computer Conference", (ss. 187–193), 1984.

[31] George A. Miller: "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information", *The Psychological Rev.*, 63, ss. 81–97, 1956.

[32] H.W. Miller: *Reengineering legacy software systems*, Digital Press, 1998.

[33] J. Moad: "Maintaining the competitive edge", *Datamation*, 36(4), ss. 61–62, 66, 1990.

[34] Paul Oman ja Jack Hagemeister: "Metrics for assessing a software system's maintainability", teoksessa "Proceerdings of Conference on Software Maintenance", (ss. 337–344), IEEE, 1992.

[35] Paul Oman, Jack Hagemeister ja D. Ash: *A Definition and Taxonomy for Software Maintainability*, Tekninen Raportti 83843, University of Idaho, 1991.

[36] D.L. Parnas ja P.C. Clements: "A Rational Design Process: How and why to Fake it", *IEEE Transactions on Software Engineering*, SE-12(2), ss. 251–257, 1986.

[37] Markus Pizka: "Adaptation of Large-Scale Open Source Software", teoksessa "Proceedings of the Eghth European Conference on Software Maintenance and Reengineering", (ss. 147–153), IEEE, 2004.

[38] O. Port: "The software trap – automate or else", *Business Week*, 3051(9), ss. 142–154, 1988.

[39] Roger S. Pressman: *Software Engineering – A Practitioner's Approach*, 5. painos, McGraw-Hill, 2000, eurooppalainen versio, sovittanut Darrel Ince.
URL `http://www.pressman5.com/`

[40] Patrick Rivett: *Principles of Model Building*, John Wiley & Sons, 1972.

[41] Patrick Rivett: *Model Building for Decision Analysis*, John Wiley & Sons, 1980.

[42] L. H. Rosenberg: *Software re-engineering*, Tekninen raportti, Software Assurance Technology Center, lokakuu 1996.

[43] Izzet Sahin ja Fatemeh Zahedi: "Control limit policies for warranty, maintenance and upgrade of software systems", *IIE Transactions*, 33(9), ss. 729–745, syyskuu 2001.

[44] Izzet Sahin ja Fatemeh Zahedi: "Policy analysis for warranty, maintenance, and upgrade of software systems", *Journal of Software Maintenance and Evolution: Research and Practice*, 13(6), ss. 469–493, marraskuu/joulukuu 2001.

[45] Robert C. Seacord, Daniel Plakosh ja Grace A. Lewis: *Modernizing Legacy Systems*, The SEI Series in Software Engineering, Addison-Wesley, 2003.

[46] Harry Sneed: "Planning the reengineering of legacy systems", *IEEE Software*, 12(1), ss. 24–34, 1995.

[47] Ian Sommerville: *Software Engineering*, 5. painos, Addison-Wesley, 1995.

[48] T. Tilus: "ELTIS julkaisut", verkkojulkaisu, 2006.
URL `http://www.titu.jyu.fi/eltis/julkaisut.html`

[49] T. Tilus *ym.*: *MODEST: A Method for Early System Modernization Pressure Estimation*, Tekninen raportti, Tietotekniikan tutkimusinstituutti, Jyväskylän yliopisto, 2005.
URL `http://titu.jyu.fi/eltis/papers/non-ref/Tilus%20-%20MODEST%20-%20A%20Method%20for%20Early%20System%20Modernization%20Pressure%20Estimation.pdf`

[50] Giuseppe Visaggio: "Value-based decision model for renewal processes in software maintenance", *Annals of Software Engineering*, 9, ss. 215–233, 2000.

[51] W. Paul Vogt: *Dictionary of Statistics and Methodology: A Nontechnical Guide for the Social Sciences*, 2. painos, Sage, 1998.

[52] Evan Wallace, Paul C. Clements ja Kurt C. Wallnau: "Discovering a system modernization decision framework: a case study in migrating to distributed object technology", teoksessa "Software Maintenance 1996, Proceedings., International Conference on", (ss. 185–195), 1996.

[53] Ian Warren: *Renaissance of Legacy Systems*, Springer Verlag, 1999, ISBN 1-85233-060-0.

[54] Ian Warren ja Jane Ransom: "Renaissance: a method to support software system evolution", teoksessa "Proceedings of the 26th Annual International Computer Software and Applications Conference", (ss. 415–420), IEEE Computer Society, 2002.

[55] M. Zelkowitz, A. Shaw ja J. Gannon: *Principles of Software Engineering and Design*, Prentice-Hall, 1979.

[56] Nicholas Zvegintzov: "Software should live longer", *IEEE Software*, 15(4), ss. 19, 21, heinä-elokuu 1998.