



KUOPION YLIOPISTO

Palveluarkkitehtuurin soveltaminen terveydenhuollossa

Osa 1: hyödyt, kustannukset, arviointi ja hankinnat

SerAPI-projekti	
Tekijät	Juha Mykkänen, Assi Pöyhölä, Tanja Toroi, Pasi Riikonen, Annamari Riekkinen
Päiväys	31.8.2007

Sisällysluettelo

1	Johdanto: palveluarkkitehtuurin perusteet	5
2	SOA ja web services - tavoitellut hyödyt.....	7
2.1	Toiminnalliset ja asiakasorganisaation hyödyt.....	8
2.2	Hyödyt sovellustuotannon kannalta	15
2.3	Tekniset hyödyt.....	19
2.4	Palvelupohjaisen arkkitehtuurin kehitysvaiheet.....	22
2.5	Toiminnalliset hyödyt, sovelluskehityksen hyödyt ja tekniset hyödyt eri kypsyysvaiheissa	23
2.6	Yhteenveto tavoitelluista hyödyistä.....	26
3	Ohjelmistokehityksen osatekijät ja kehitysstrategiat palvelupohjaisessa arkkitehtuurissa.....	29
3.1	Projektiperusteiset tekijät.....	29
3.2	Sisäisen rakenteen ohjelmisto-ominaisuudet.....	29
3.3	Ulkoiset laatuattribuutit	29
3.4	Palvelupohjaisen arkkitehtuurin kehitysstrategiat	30
3.5	Kehitysstrategian vaikutus projektikustannuksiin ja ohjelmisto-ominaisuuksiin..	30
3.5.1	Ohjelmistoattribuutit top-down-menetelmässä	31
3.5.2	Ohjelmistoattribuutit bottom-up-menetelmässä	32
3.5.3	Ohjelmistoattribuutit Meet-in-the-middle-menetelmässä	34
3.5.4	Ristiriitatekijät	34
3.6	Yhteenveto ohjelmistokehityksen osatekijöistä ja kehitysstrategioista palvelupohjaisessa arkkitehtuurissa	35
4	SOA ja Web services-hyötyjen arviointi ja mittaus.....	36
4.1	Tietojärjestelmien arviointimallit	36
4.1.1	Eri arviointitapoja.....	37
4.1.2	VATAM-suositukset tietojärjestelmien arviointiin	40
4.1.3	DeLone ja McLean -mallit.....	42
4.1.4	Muita arviointimalleja.....	44
4.2	Palvelupohjaisten ratkaisujen laadun muodostuminen	45
4.3	Palveluarkkitehtuurin ja Web-sovelluspalvelujen tavoiteltujen hyötyjen arviointi	48
4.3.1	Kooste arviointimittareista	48
4.3.2	Esimerkki arviointimittareiden valinnasta: järjestelmäympäristön tehostunut hallinta ja ylläpito	55
4.4	Arviointiprosessin vaiheet	56
4.5	Yhteenveto hyötyjen arvioinnista	58

5	Web Services ja SOA: taloudellinen arviointi.....	60
5.1	Web Services ja SOA: kannattavuuden arvioinnin mittarit.....	60
5.1.1	Kassavirta-analyysi	60
5.1.2	Takaisinmaksuaika-analyysi.....	62
5.1.3	Sijoitetun pääoman tuotto (ROI) web-sovelluspalveluissa.....	63
5.1.4	Kannattavuusmittareiden soveltuvuus.....	64
5.2	Kustannusten, hyötyjen ja riskien määrittely.....	65
5.2.1	Kustannusten taloudellinen malli	66
5.2.2	Kulut ja kustannukset	67
5.2.3	Riskinhallintaan vaikuttavia tekijöitä	69
5.3	Palvelupohjainen arkkitehtuuri (SOA) ja ROI.....	70
5.4	Yhteenveto kustannusten ja hyötyjen muodostumisesta	72
6	Sovelluspalvelut ja rajapinnat ohjelmistohankinnoissa	73
6.1	Hankintatavat ja hankintaprosessi yleisesti.....	73
6.1.1	Hankintatapojen luokittelua	73
6.1.2	Palveluarkkitehtuurin hallinta ja hankinnat	74
6.1.3	Hankintaprosessien mallit	75
6.1.4	Julkisen hankintaprosessin erityispiirteitä.....	76
6.2	Hankintojen suunnittelu ja ohjaus	79
6.2.1	Hankintastrategia	79
6.2.2	Hankintaprosessin toteuttaminen ja hankintasuunnitelma.....	80
6.2.3	Vaatimusmäärittelyt.....	81
6.2.4	Tarjouspyynnön valmistelu.....	82
6.3	Valinta ja sopimukset.....	83
6.3.1	Toimittajien tunnistaminen.....	83
6.3.2	Sopimusvalmistelu	83
6.3.3	Tarjousten arviointi ja toimittajan valinta.....	85
6.4	Toteutus, hyväksyminen ja seuranta	86
6.4.1	Toimituksen ohjaus	86
6.4.2	Hyväksyminen ja käyttöönotto.....	86
6.4.3	Seurantavaihe	88
6.5	Hankintojen tarkistuslistoja	88
6.6	Yhteenveto ja johtopäätökset hankinnoista.....	93
7	Yhteenveto.....	96

Esipuhe

Terveydenhuollon kansallisen tietojärjestelmäarkkitehtuurin ja arkiston suunnitteluperiaatteeksi on valittu palvelupohjainen arkkitehtuuri (SOA). Useiden muidenkin maiden kansalliset hankkeet sekä suuret terveystalvelujen tuottajat vannovat SOA-ajattelun nimeen. Tietojärjestelmien kehittämislaineiden tarjoajat ja kaupan sekä teollisuuden suuryritykset esittelevät palvelupohjaisia ratkaisujaan. Standardointijärjestöjen rajapintoja ja malleja viedään palveluajattelun suuntaan. Mutta mistä onkaan kyse? Miten palvelupohjainen ajattelu tulee näkymään käytännössä, paikallisessa terveydenhuollon ja muiden toimialojen tietojärjestelmien kehittämisessä ja käyttämisessä? Mitä tavoitteita voidaan saavuttaa palveluarkkitehtuurin avulla ja kuinka niiden toteutumista voidaan seurata?

Tähän dokumenttiin on koottu SerAPI-hankkeen (2004-2007) aikana tehtyjä selvityksiä ja työpapereita palveluarkkitehtuurin tavoitelluista hyödyistä, kustannuksista, näiden mittaamisesta, tietojärjestelmien arvioinnista, hankinnoista ja palvelupohjaisen kehitystyön peruslähestymistavoista. Näihin aiheisiin tunnistettiin hankkeen aikana tarve koota ja jäsentää valmista tietoa sekä soveltaa sitä suhteessa palveluarkkitehtuuriin sekä kehittää työkaluja arviointia ja hallintaa varten. Monet näistä seikoista liittyvät palveluarkkitehtuuria hyödyntävän organisaation tietohallintoon sekä käyttäjien ja johdon tarpeisiin ja vaatimuksiin. Ne myös ohjaavat kokonaisuuden kehittämistä ja linkittävät tietotekniikan kehitystä organisaation toiminnan kehittämiseen.

Dokumentti on ensimmäinen osa "SOA- ja web services-soveltamisoppaasta", joka SerAPI-hankkeen peruslähestymistapaa mukaillen sisältää kolme osaa. Ensimmäiseen osaan on koottu erityisesti tavoiteltuihin hyötyihin, arviointiin ja käyttäjäorganisaation tarve- ja hankintanäkökulmiin liittyviä seikkoja. Toinen osa sisältää kokoelman palvelupohjaisten ratkaisujen ja arkkitehtuurien kehittämiseen liittyviä malleja ja esimerkkejä. Kolmas osa koostuu useista erillisistä tuotoksista ja sisältää etenkin web-sovelluspalvelutekniikoiden teknisiin soveltamistapoihin liittyviä selvityksiä ja suosituksia.

Työ liittyy SerAPI-hankeeseen (Palveluarkkitehtuuri ja Web-sovelluspalvelut Terveydenhuollon Ohjelmistotuotannossa ja -integraatiossa), jossa tutkitaan ja kehitetään web-sovelluspalvelujen ja palvelupohjaisen arkkitehtuurin hyödyntämistä terveydenhuollon tietojärjestelmätarpeisiin ja sovellusintegraatioon ja uusiin sekä olemassa oleviin ohjelmistotuotteisiin. Hanketta ovat rahoittaneet Tekes (päätös nrot 40437/04, 40353/05, 40251/06) sekä Medici Data Oy, Datawell Oy, Fujitsu Services Oy, Pohjois-Savon sairaanhoitopiiri, WM-data Oy, Commit Oy, Intersystems B.V. Finland, Mediconsult Oy, Microsoft Oy, Oracle Finland Oy, Satakunnan sairaanhoitopiiri, Bea Systems Oy, Softera ratkaisut Oy, Helsingin ja Uudenmaan sairaanhoitopiiri, Kuopion kaupunki, Kustannus Oy Duodecim ja Mawell Oy.

Tekijät kiittävät kaikkia oppaaseen aiheita antaneita ja sisältöä kommentoineita osapuolia sekä työpajojen osallistujia.

1 Johdanto: palveluarkkitehtuurin perusteet

Tietojärjestelmien kehittämisessä *palveluarkkitehtuuri* (*service-oriented architecture, SOA*) tarkoittaa lähestymistapaa, jossa sovelluksia tai toimintaprosesseja muodostetaan pienemmistä, määriteltäviä osatehtäviä toteuttavista palveluista. Tietojärjestelmäkokonaisuus hahmotetaan joukkona palveluita (sovelluspalveluita), joita tarpeen mukaan yhdistelemällä voidaan entistä helpommin toteuttaa tai mukauttaa sovelluksia eri käyttötarpeisiin. Vanhoja sovelluksia voidaan liittää osaksi uusia ratkaisuja käyttämällä niiden tietoja tai toimintoja palvelurajapintojen kautta. Suuri osa palveluarkkitehtuurin tuomista muutoksista tapahtuu käyttäjiltä piilossa "konepellin alla", mutta vaikutukset ulottuvat laajasti tietotekniikalla tuettuun toimintaan.

Palveluarkkitehtuuria on erityisesti suositeltu suurten ja monimutkaisten järjestelmäkokonaisuuksien hallintaan. Varsinkin sairaaloissa ohjelmistoja on paljon ja erilaisia, ja uusien kehittämisen sijaan haasteena onkin saada erilliset sovellukset toimimaan järkevästi yhdessä. Terveydenhuollossa on yhä enemmän tietoa ja tietämystä, toiminta pohjautuu asiantuntijuuteen ja sisältää tilanne- tai organisaatiokohtaisia erityispiirteitä. Lisäksi uusi tietämys ja uudet alueelliset ja kansalliset toimintamallit muokkaavat jatkuvasti toimintaa ja toimintaympäristöä aiheuttaen myös tietojärjestelmille kehitystarpeita. Palveluarkkitehtuurin perusajatukset tukevat tällaista toimintaympäristöä.

Palveluarkkitehtuurin soveltaminen vaatii määrätietoista tavoiteltujen hyötyjen määrittelyä. Lähestymistapaan on kohdistettu paljon odotuksia ja lupauksia, joiden toteutumisen mittaaminen vaatii niiden purkamista perusteltuihin osiin. Hyötyjen perustelu myös auttaa tunnistamaan edellytykset, joita niiden saavuttamiseksi on saatava täyttymään.

Palveluarkkitehtuurista haetaan hyötyjä useilla eri tasoilla: organisaation toiminnan näkökulmasta, sovellusten tuottamisen ja integroinnin näkökulmasta sekä teknisellä tasolla. Esimerkiksi terveydenhuollon organisaatioissa tavoitellut hyödyt liittyvät jo hankittujen ratkaisujen uudelleenkäyttöön, yhdenmukaisuuden lisääntymiseen ja päällekkäisyyksien vähenemiseen, ja lopulta entistä helpompiin ja nopeampiin muutoksiin, kun tietojärjestelmiin kohdistuu uusia vaatimuksia. Käyttäjien, johdon tai asiakkaiden esittämiä tarpeita vastaavia sovelluspalveluja voidaan hankkia tai kehittää ja liittää osaksi kokonaisuutta tarvittaessa siten, että kokonaisia järjestelmiä ei tarvitse vaihtaa tai odottaa kokonaisjärjestelmän seuraavia versioita. Tätä kautta muuttuviin tarpeisiin voidaan vastata entistä nopeammin tai järjestelmiin saadaan entistä helpommin lisäpiirteitä.

Palvelupohjainen järjestelmien kehittämismalli poikkeaa suurten kokonaisjärjestelmien tai erillisistä sovelluksista koottujen järjestelmäkokonaisuuksien kehittämisestä. Palveluista koostettuihin järjestelmäkokonaisuuksiin hankittavat lisäosat ja uudet ominaisuudet voidaan määritellä, valita ja liittää toisiinsa tarkemmin, mahdollisesti suuremmasta joukosta tarjolla olevia vaihtoehtoja. Toisaalta samoja palveluita ja rajapintoja voidaan tarjota sekä uusista että vanhoista sovelluksista, mikä helpottaa järjestelmien uusimisessa ja yhteisesti sovittujen sisältöjen tai toimintatapojen käyttöönotossa. Yleisimmin eri yksiköissä tai erillisjärjestelmissä käytetyt tiedot ja toiminnot toteutetaan keskitetyksi *ydinpalveluina*, jolloin samoja tietoja ei tarvitse syöttää useisiin järjestelmiin. Pienemmistä osista muodostettu järjestelmä mukautuu helpommin muutoksiin myös, kun toimintaa organisoidaan uudelleen. Teknisesti hyödynnetään jo valmiiksi netin käyttöön rakennettua infrastruktuuria ja avoimia tekniikoita.

SOA-lähestymistapa yhdistää sovellusintegraation, prosessien mallinnuksen ja komponenttipohjaisen sovellusten kehittämisen ajatuksia. Keskeistä lähestymistavassa on tietotekniikkaratkaisujen läheinen suhde toimintaprosesseihin: monia palveluja mallinnetaan, rakennetaan ja hankitaan siten, että niillä on vastaavuus tuettavan toiminnan kanssa. Prosesseja ja työnkuluja voidaan tukea eri

vaiheisiin sopivien palvelujen avulla, ja prosessien etenemistä voidaan ohjata, seurata ja joiltakin osin myös automatisoida.

Tähän dokumenttiin on koottu palveluarkkitehtuurin tavoitteisiin, hyötyihin, kustannuksiin, kehittämisstrategioihin ja hankintoihin liittyviä malleja ja selvityksiä. Dokumentin luvussa 2 käsitellään ja analysoidaan palveluarkkitehtuurin ja web-sovelluspalvelujen käytöllä tavoiteltuja hyötyjä. Luvussa 3 käsitellään tietojärjestelmien ja ohjelmistojen kehittämiseen liittyviä laatutekijöitä sekä projektivaikutuksia suhteessa palvelupohjaisen arkkitehtuurin prosessilähtöiseen (top-down) ja vanhoista sovelluksista liikkeelle lähtevään (bottom-up) kehittämisstrategiaan. Luvussa 4 esitellään tietojärjestelmien arviointiin liittyviä malleja ja suosituksia, ja esitetään malleja palvelupohjaisten arkkitehtuurien ja ratkaisujen arviointiin ja mittaamiseen. Luvussa 5 kuvataan web-sovelluspalvelujen ja palveluarkkitehtuurin kustannusten ja taloudellisten hyötyjen malleja ja arviointia. Luvussa 6 käsitellään tietojärjestelmä- ja ohjelmistohankintoihin liittyviä suosituksia ja sovelletaan niitä suhteessa sovelluspalveluihin ja rajapintoihin. Monissa luvuissa käytetään viitekehystenä SerAPI-hankkeen kolmitasoista (prosessit, toiminta, käyttäjäorganisaatio / sovellukset ja loogiset ratkaisut, ratkaisujen kehitysnäkökulma / alusta ja tekniset ratkaisut) mallia, jota on kuvattu tarkemmin mm. oppaan osassa 2 " Palveluarkkitehtuurin soveltaminen terveydenhuollossa osa 2: prosessien ja palvelujen määrittely ja suunnittelu".

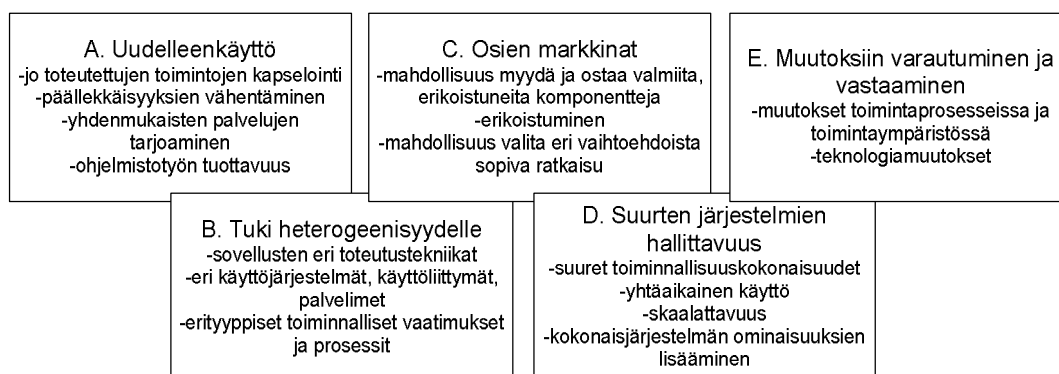
2 SOA ja web services - tavoitellut hyödyt

Tämä luku erittelee palveluarkkitehtuuri-lähestymistavalla (SOA) ja web services -tekniikoilla tavoiteltuja hyötyjä ja esittelee kullekin niistä perusteluita ja hyötyjen saavuttamisen edellytyksiä, erityisesti terveydenhuollon tietojärjestelmien näkökulmasta. Luku kokoaa yhteen, järjestelee ja perustelee monissa eri lähteissä mainittuja tavoitteita sekä käsittelee myös niiden edellytyksiä ja haasteita. Luku toimii pohjana web services -tekniikoita ja palveluarkkitehtuuria hyödyntävien ratkaisujen arvioinnissa sekä menetelmä- ja määrittelyratkaisujen valinnassa ja kehityksessä.

Tavoiteltuja hyötyjä käsitellään SerAPI-projektissa käytettyä viitemallia soveltaen kolmella tasolla:

- organisaation toiminta (process), erityisesti asiakasorganisaatio,
- sovellusten tuottaminen (application),
- tekniset hyödyt (platform).

Kuvassa 1 näkyy yleisiä tietojärjestelmien kehitykseen liittyviä tavoitteita, joita myös palveluarkkitehtuurin ja web-sovelluspalvelujen avulla pyritään saavuttamaan.



Kuva 1. Suurten tietojärjestelmäkokonaisuuksien kehittämisen tavoitteita ja haasteita.

Tässä luvussa kuvataan saatujen kokemusten ja kirjallisuuden pohjalta palveluarkkitehtuuri (SOA) -lähestymistavan ja Web services -tekniikoiden potentiaalisia hyötyjä. Kustakin hyödyistä esitetään:

- lyhyt kuvaus usein esiin tuoduista hyödyistä,
- ne palveluarkkitehtuurin ja web-sovelluspalvelujen piirteet, jotka tukevat suoraan ko. tavoitetta (mihin perustuu),
- seikat joista on huolehdittava, jotta hyöty saavutetaan (edellytykset) sekä
- erityishuomiota vaativia tai hyödyn toteutumista vaikeuttavia seikkoja (haasteet).
- sekä kunkin tavoitellun hyödyn toteutumisen arviointikriteerejä ja mittareita, joissa on tunnistettu tutkimuksissa käytettyjä informaation laadun, järjestelmän laadun, palvelun laadun ja nettohyödyn tai organisaation hyödyn attribuutteja, ja kuinka ne liittyvät Web-sovelluspalveluiden tavoiteltujen hyötyjen arviointiin. Näitä ja muita mittausseikkoja käsitellään tarkemmin luvussa 4.3.

Tavoiteltuja hyötyjä sekä niihin liittyviä mittareita on koottu monista eri lähteistä, samoin kuin hyötyjen perusteluja. Hyötyjä on ryhmitelty ja yhdistelty pyrkien poistamaan päällekkäisyyksiä, mutta koska monet tavoitelluista hyödyistä liittyvät toisiinsa tai ovat ylä- ja alakäsitteitä, on myös hyväksyttävä päällekkäisyyksien olemassaolo. Keskeisiä lähteitä ovat olleet (Wilkes 2003, Campbell ym. 2006, Marquis 2006, Central Computer & Telecommunications Agency 2002; OGC 2002; OGC 2004, Doll & Torkzadeh 1988, Webmethods 2006, Schmelzer 2005, Honey 2006).

2.1 Toiminnalliset ja asiakasorganisaation hyödyt

Tavoite	Toiminnallinen joustavuus
Kuvaus	Tavoitteena on mahdollistaa nopea reagointi organisaation toiminnassa ja toimintaympäristössä tapahtuviin muutoksiin, esim. uudet tietovaatimukset, toimintaprosessin muuttaminen, uudet kumppanit, lakimuutokset. Tavoitteena on myös mahdollistaa uusien toimintojen ja eri paikoissa sijaitsevien toimijoiden "helppo lisättävyys" kokonaisjärjestelmään sekä toimintoihin ja prosesseihin tarvittavien muutosten nopea vieni tuotantoon. Näin riippuvuus tuoteversioiden toteutusrytmeistä vähenee. Joustavuus sisältää sekä mahdollisuuden tehdä muutoksia helposti (adaptability) että nopean reagoinnin uusiin tarpeisiin ja vaatimuksiin (responsiveness). Yleisinä tavoitteina ovat uusien toimintamahdollisuuksien toteuttamisen helpottaminen (kustannustehokkaasti) ja toimintaan liittyvien riskien vähentäminen (virheiden ja tappioiden välttäminen) - organisaation kilpailukykyyn parantaminen.
Mihin perustuu	Avoin arkkitehtuuri, yhteentoimivuus, järjestelmien jako järkeviin toiminnallisiin osiin, osien välisen kytkennän riittävä löysyys. Rajapinta voi piilottaa pienet toiminnalliset muutokset kokonaan. Rajapinnan avulla "eristetty" osa voidaan tarvittaessa toteuttaa uudelleen helpommin kuin kokonainen järjestelmä. Osa teknisten yksityiskohtien sopimisesta jää pois kehitystyöstä.
Edellytykset	Muutosta tukeva arkkitehtuuri, jossa eri osat eivät ole "yhden sovelluksen sisällä" tai muuten tiukasti sidoksissa keskenään. Standardeista ja arkkitehtuurista on muodostettava pelisäännöt muutosten tekemiselle.
Haasteet	Miten yleiskäyttöisistä osista saadaan toiminnan tarkkoihin tarpeisiin vastavia? Mitkä ovat "oikean kokoisia" palveluita kulloisiinkin vaatimuksiin? Suuren määrän tietoa ja toiminnallisuutta sisältävät sovelluspalvelut (operaatiot) eivät ole niin uudelleenkäytettäviä kuin pienet, ja niiden käyttäminen vuorovaikutteisuutta lisäävissä tilanteissa on haasteellista. Uuden ohjaavan arkkitehtuurin käyttöönotto vaatii hankinta- ja käyttöönottoprosessin kehittämistä. Kokonaishyödyn mittaaminen ja arviointi on vaikeaa, koska varaudutaan tulevaisuuden tarpeisiin, joita ei tunneta tarkasti ratkaisuja tehtäessä.

Miten arvioidaan	<p>Saatavaa hyötyä voidaan arvioida esim. vaikutusten arvioinnilla, jolloin voidaan arvioida ratkaisujen vaikutuksia toimintaan sekä uusien toimintamahdollisuuksien toteuttamisen nopeutta ja helppoutta. Myös käyttöönottoon kuluva aika, joka tarvitaan uusien piirteiden ja palveluiden lisäämiseen, ja kuinka kustannustehokkaasti uudistus toteutuu, ovat tunnistettuja arvioinnintarpeita. Lisäksi arviointiin kuuluu toimintaan, palveluihin ja järjestelmään liittyvien riskien analysointi. Monissa alla olevissa kohdissa on tähän liittyviä osatekijöitä.</p> <p>Mittareita: toimintaprosessien mittarit, käyttäjien kokemat työnkuvan ja työmäärän muutokset, muutokset työnkulkuihin ja kommunikaatioon, muutospyyntöjen lukumäärä tietyllä aikavälillä, toteutettujen muutosten lukumäärä ja suhteellinen määrä muutospyyntöihin, järjestelmästä löydettyjen virheiden määrä, käyttäjien tekemien virheiden määrä.</p>
-------------------------	---

Tavoite	Sovellusten parantunut liitettävyys
Kuvaus	Pyritään siihen, että eri järjestelmät ja palvelut ovat entistä vähemmällä työllä liitettävissä yhteentoimiviksi kokonaisuuksiksi ja siihen, että tarvitaan entistä vähemmän paikallista räätälöintiä. Vapautuvaa työpanosta voidaan siten suunnata varsinaiseen toiminnan kehittämiseen teknisten yksityiskohtien hiomisen sijaan. Standardien mukaisuus voi lisätä hankintavaihtoehtoja.
Mihin perustuu	Rajapinnat, riippumattomuus toteutustekniikoista (avoimet tekniikat), tekniset ja sisällölliset standardit, arkkitehtuurin joustavuus.
Edellytykset	Toiminnallinen ja tiedollinen yhteensopivuus. Tekniikoiden yhdenmukainen soveltaminen. Standardien käyttö ja sopiminen. Yhdenmukaiset toimintatavat järjestelmien integroinnissa palvelupohjaisesti.
Haasteet	Joustavuuden ja standardoinnin yhteensovittaminen. Jo pelkällä teknisten standardien soveltamisella saavutetaan liitettävyshyötyjä, mutta vasta sisällöllinen standardointi lisää laajemmassa mittakaavassa yhteentoimivuutta. Standardeja voidaan soveltaa useilla eri tavoilla.
Miten arvioidaan	Uudelleenkäytön arvioinnin (ks. sovellusten uudelleenkäyttö) lisäksi voidaan arvioida integrointihankkeisiin käytetyt kustannukset ja työmäärät, kuinka työ jakautuu paikallisen räätälöinnin ja yleisen liitettävyyden välillä. Tekijät voivat arvioida liittämisen helppoutta ja suoraviivaisuutta. Lisäksi voidaan tarkastella tiettyä rajapintaa tukevien tuotevaihtoehtojen lukumäärä sekä ovatko integroinnissa käytettävät toimintatavat yhdenmukaisia ja vakioituja? Mittareita: Työmäärä, integroinnista aiheutuvat kustannukset, integroinnista johtuvien virheiden lukumäärä, tunnistettujen käyttötilanteiden ja järjestelmien lukumäärä, joihin ratkaisulla saavutetaan liitettävyys.

Tavoite	Jo tehtyjen investointien hyödyntäminen
Kuvaus	Olemassa olevia järjestelmiä, verkkoja ja niihin liittyvää osaamista voidaan hyödyntää ja käyttää uudelleen. Tehtyjä palveluja voidaan käyttää useissa eri prosesseissa ja järjestelmissä.
Mihin perustuu	Palvelujen, sovellusten ja infrastruktuurin (esim. verkko, palvelimet jne.) uudelleenkäyttö, avoimuus. Ks.sovellusten uudelleenkäyttö.
Edellytykset	Uudelleenkäytettävien osien tunnistaminen, kehitystä ohjaava arkkitehtuuri, standardointi.
Haasteet	Vanhojen ratkaisujen piirteet voivat ohjata liikaa uusien yksityiskohtia. Vanhojen järjestelmien sisällöllinen mukauttaminen ulkoisiin määräyksiin tai standardeihin on työlästä.
Miten arvioidaan	Ks. uudelleenkäyttö. Lisäksi: Hyötyä arvioitaessa tarkastellaan mm. kuinka laajaa, yleistä tai kohdennettua uudelleenkoulutusta tarvitaan? Kuinka paljon aikaa käytetään käyttöönottoihin liittyviin koulutuksiin ja kenen toimesta sitä annetaan? Kuinka paljon resursseja käytetään perehdyttämiseen tai täydennuskoulutukseen? Mittareita: ks. taloudelliset mittarit

Tavoite	Sovellushankinta- ja integraatiokustannusten aleneminen
Kuvaus	Tavoitteena on, että organisaatioiden on mahdollista hankkia uudet sovellusten ominaisuudet ja eri sovellusten liitännät entistä vähemmän kustannuksin. Valinnanvara hankittavien osien ja kumppanien osalta voi kasvaa.
Mihin perustuu	Sovellusten liittämiseen ja tuottamiseen tarvittavat tekniikat perustuvat piirteisiin, joita tuetaan osana käyttöjärjestelmien, palvelinten ja vähitellen myös sovellusten uusien versioiden perusominaisuuksia. Sovellusten liittämiseen käytettävät mekanismit ovat yksinkertaisia ja yhdenmukaisia.
Edellytykset	Standardointi ja yhdenmukaisen arkkitehtuurin noudattaminen.
Haasteet	Tämä kehitys on vasta alkuvaiheessa. Perusominaisuuksien lisäksi tarvittavista lisäominaisuuksista on edelleen maksettava. Vaikka tekniset ominaisuudet ovatkin helposti liitettävissä eri ohjelmistoihin, vaatii toiminnallisten ominaisuuksien yhdistäminen kehitystyötä, standardointia ja paikallista sopimista. Myös teknisellä tasolla voidaan samoja tekniikoita hyödyntää useilla tavoilla, jotka eivät ole keskenään yhteensopivia.
Miten arvioidaan	Hyötyä voidaan arvioida integraation vaatimien kustannusten osalta ja kuinka ne jakautuvat eri toimijoiden välillä? Sekä tarjoavatko käyttöjärjestelmät, palvelimet jne. liittämiseen tarvittavat tekniikat "vakiona"? Prosessien joustavuuden kustannusvaikuttavuus on vaikea arviointikohde. Ks. SOA ja Web Services-soveltamisopas, osio "Kustannukset ja hyödyt". Mittareita: Yleiset investoinnin kannattavuusmittarit - ks. taloudelliset mittarit. Virheiden määrä, virheiden esiintyvyyshäiriö, virheiden korjaukseen kuluva aika, muutospyyntöjen lukumäärä tietyllä aikavälillä. Kuinka paljon IT -hankintoihin on satsattu tietyllä aikavälillä, hankinta- ja integraatiokustannusten muutos ja suhde toteutettuihin uusiin piirteisiin.

Tavoite	Tietojärjestelmäympäristön vähittäinen kehitys
Kuvaus	Kokonaisjärjestelmää ja järjestelmäympäristöä voidaan kehittää vähitellen, "pala kerrallaan" sitä mukaa, kun organisaation toimintaa kehitetään. Kehittämisen riskit ja poikkeamat havaitaan nopeammin pienemmistä kokonaisuuksista.
Mihin perustuu	Avoimet tekniikat, yleiskäyttöiset palvelut.
Edellytykset	Kehittämisen- ja siirtymäpolun määrittely. Yhdenmukainen arkkitehtuuri ja kehittämisperiaatteet.
Haasteet	Siirtyminen sovellustasoisesta kokonaisjärjestelmän kehittämisestä sovellusta pienempien palvelujen avulla tapahtuvaan. Jos pieniä kokonaisuuksia on runsaasti, voi riskien ja poikkeaminen havaitseminen vaikeutua.
Miten arvioidaan	Ks. inkrementaalinen kehittäminen. Hyötyä arvioitaessa tarkastellaan tuleeko käyttökatoja (määrä ja kesto), onnistuuko toimintatapojen vähittäinen muuttaminen ja kuinka suuri osa toimintatapojen muutoksista voidaan tehdä suunnitellusti? Voidaan arvioida myös ovatko kehittämisperiaatteet ja arkkitehtuuri määritelty ja noudatetaanko niitä ja miten laajasti sekä vanhat että uudet toimintatavat ja järjestelmät ovat yhtä aikaa käytössä? Ennen-jälkeen -arviointia voidaan käyttää esimerkiksi eri vähittäisen kehittämisen vaiheissa, jolloin arvioidaan tilanne ennen muutosta ja kuinka muutos on vaikuttanut ja onko se kokonaistavoitteen suuntainen. Mittareita: Kehittämisperiaatteiden saatavuus ja niiden noudattamisen seuranta. Uusien piirteiden toteuttamisen vaatimuksista käyttöönottoon nopeus. Palvelun saatavuuden osalta = $(\text{Sovittu palveluaika} - \text{aika jolloin palvelu ei ole käytössä ei-sovitusti}) / \text{sovittu palveluaika} * 100$ – (Huom. Sovitusti voi olla poissa käytöstä vaikka palvelun saatavuus olisi 100 %), työmäärä.

Tavoite	Olemassa olevien sovellusten uudelleenkäyttö
Kuvaus	Tavoitteena on, että perinnejärjestelmiä tai muita jo hankittuja ohjelmistoja voidaan "kuoruttaa" palvelurajapinnoilla ja käyttää osana uusia ratkaisuja ja prosesseja.
Mihin perustuu	Palvelurajapinnat, kapselointi.
Edellytykset	Järjestelmiin pystytään tekemään tarvittavat liitännät.
Haasteet	Vanhoja järjestelmiä ei ole suunniteltu tällaista uudelleenkäyttöä varten. Jos palvelun käyttö laajenee suuresti, huolehdittava etteivät vanhat järjestelmät ylikuormitu. Järjestelmien sisäiset yksityiskohdat voivat vaikuttaa rajapintoihin.
Miten arvioidaan	Hyödyn arvioinnissa punnitaan sitä saadaanko perinnejärjestelmistä tarjottua palveluita uusien tekniikoiden avulla sekä millä tasolla ja kuinka isoina osina jo hankittuja järjestelmiä tai ohjelmistoja uudelleenkäytetään? Arvioinnin kohteena on myös, kuinka paljon työtä uudelleenkäyttö vaatii, tuleeko vanhoihin järjestelmiin kuormitus- ja hallittavuusongelmia ja kuinka suuria sisäisiä muutoksia uudelleenkäytettäviin järjestelmiin tarvitaan? Mittareita: uusien piirteiden toteuttamisen työmäärä ja nopeus, myös verrattuna aikaisempaan tilanteeseen, uudelleenkäytöstä aiheutuvat kustannukset, uudelleenkäytöstä aiheutuvien virheiden lukumäärä (sinällään ja suhteessa aikaisempaan kehitysmalliin).

Tavoite	Prosessien määrittely ja tukeminen
Kuvaus	Organisaation toimintaprosesseja voidaan määrittellä ja muuttaa keskitetysti sekä koostaa prosesseja eri tehtäviä suorittavista ohjelmistopalveluista. Prosessien määrittelyyn ja muokkaamiseen ei tarvita ohjelmointitaitoja, ja niitä voi tehdä sovellusalueen asiantuntija myös asiakasorganisaatiossa. Prosessien osia voidaan joustavasti ohjata eri yksiköihin tai kumppaneille. Prosesseja voidaan yhdenmukaistaa, seurata ja osin automatisoida. Tietojärjestelmän aiheuttamat rajoitteet toiminnan muuttamiselle vähenevät.
Mihin perustuu	Yleiskäyttöiset, eri tilanteisiin sopivat palvelut saatavilla. Prosessia ohjaavat ohjelmistot (esim. toiminnanohjausjärjestelmä, workflow engine). Palvelujen välinen löysä kytkentä.
Edellytykset	Kokonaisprosessit on ulkoistettu yksittäisistä sovelluksista. Prosessien osiin oltava käytettävissä palveluita, jotka vastaavat toiminnallisesti prosessin osana olevia tehtäviä. Prosessimäärittelyjä tukevat välineet ja menetelmät.
Haasteet	Tason määrittely, jolla prosesseja kuvataan. Prosesseihin kuuluvien "ihmistehtävien" kuvaus ja liittäminen automatisoitaviin osiin. Poikkeustilanteet. Prosessien määrittelijän on tunnettava toiminta, johon ratkaisua tehdään. Prosessimuutokset etenkin organisaatioiden välillä vaativat myös sopimusten tekemistä.
Miten arvioidaan	Hyödyn arvioinnissa tarkastellaan onnistuuko prosessien muuttaminen ilman muutoksia tietojärjestelmiin, tai millaisia muutoksia tietojärjestelmiin tarvitaan? Arviointia tehdään myös kuinka prosessit on kuvattu, näkyykö prosesseista mihin kohtiin tietojärjestelmät liittyvät, näkyykö näissä kohdissa tarkasti, mitkä osatehtävät ja tiedot liittyvät sovelluksiin ja palveluihin? Arvioidaan myös vastaavatko käytettävissä olevat palvelut prosessien osana olevia tehtäviä ja minkä verran prosessimuutosten toteuttamiseen tarvitaan hankintoja tai ohjelmointityötä? Lisäksi arvioidaan saadaanko toimivista prosesseista seurantatietoa ja voidaanko niihin vaikuttaa sekä ovatko prosesseihin liittyvät vastuut määritelty ja tiedossa eri toimijoille? Mittareita: kuinka suuri osa tarvittavista muutoksista saadaan tehtyä keskitettyihin prosessi-määrittelyihin, millaista asiantuntemusta tarvitaan prosessimuutoksien tekemiseen (pehmeä mittari), uusien osapuolten prosesseihin mukaan ottamisen nopeus, tietojärjestelmämuutosten määrä ja laajuus, prosessien seurantatietojen saatavuus.

Tavoite	Järjestelmäympäristön tehostunut hallinta ja ylläpito
Kuvaus	Isot kokonaisjärjestelmät voidaan jakaa hallittavan kokosiin kokonaisuuksiin. Palveluista koostettua kokonaisjärjestelmää voidaan kehittää ja tarkastella osa kerrallaan. Tarjolla on hallintavälineitä palvelupohjaisen tuotantoympäristön seurantaan (mm. reaaliaikainen prosessien seuranta + yhteenvedot) ja hallintaan (mm. ongelmatilanteiden selvittely) sekä teknisellä että toiminnallisella tasolla. Päällekkäinen samojen tietojen ja toimintojen ylläpito eri järjestelmissä vähenee.
Mihin perustuu	Palveluista koostuvan järjestelmän eri osia voi ylläpitää ja päivittää ilman laajoja vaikutuksia muihin osiin, esim. yksittäisen osan uusiminen tai käyttöönotto ei vaadi koko järjestelmän alasajoa tai päivitystä.
Edellytykset	Yhdenmukaistaminen uusien palvelujen hankinnan yhteydessä, vanhojen ratkaisujen poistaminen järjestelmäympäristöstä.

Haasteet	Kokonaisuuden monimutkaisuus ei sinällään vähene uusia sovelluspalveluja tuottamalla, jos samalla ei pyritä yhdenmukaistamaan järjestelmäympäristöä. Kokonaisuus voi monimutkaistua jos se koostuu monista pienistä palveluista. Hallintamekanismit ovat alusta- tai tuotekohtaisia.
Miten arvioidaan	Arvioitaessa saavutettavaa hyötyä tarkastellaan kokonaisjärjestelmän osien lukumäärä ja koko, ovatko toimintatavat määritellyt keskeisiin hallinta- ja ylläpitotehtäviin ja ovatko ne yhdenmukaisia sekä kuinka paljon päällekkäisyyksiä on eri järjestelmissä, rekistereissä ja palveluissa. Järjestelmän osien ylläpidosta ja päivityksestä aiheutuvia sivuvaikutuksia muihin osiin arvioidaan mm. tarkastelemalla kuinka usein päivityksiä tehdään, kuinka suuressa osassa tulee ongelmatilanteita ja kuinka kauan menee järjestelmän toipumiseen? Yhdenmukaisten ja keskitettyjen hallintamekanismien saatavuutta ja kattavuutta voidaan arvioida ja kuinka paljon työtä ja erilaista osaamista tarvitaan ympäristön hallintaan? Mittarit: ks. luku 4.3.2.(esimerkki)

Tavoite	Järjestelmien parantunut käytettävyys
Kuvaus	Tavoitteena on, ettei käyttäjien tarvitse syöttää samoja tietoja useisiin eri järjestelmiin. Käytettävyyttä lisäävät esim. kertakirjautuminen, monissa järjestelmissä tarvittavien tietojen keskitetty ylläpito, ajantasaisten yhteisten tietojen saatavuuden varmistaminen eri sovelluksissa ja yhdenmukaisen näkymän ja toiminnallisuuden tarjoaminen käyttäjälle. Järjestelmiä voidaan myös muokata helpommin tai personoida käyttäjien tarpeiden mukaan.
Mihin perustuu	Yhteisten palvelujen käyttö eri sovelluksissa. Infrastruktuurin tarjoamat kuormantasaus- ym. mahdollisuudet käyttökatkojen ehkäisemiseksi. Käyttäjän näkymää voidaan yhdenmukaistaa esim. työpöytäintegraation tai portaalien avulla.
Edellytykset	Sovellusten valmiudet käyttää yhteisiä rakennusosia.
Haasteet	Karkeajakoisten palvelujen ja vuorovaikutteisuuden yhteensovittaminen. Palvelujen teknisen saatavuuden varmistaminen (palvelutaso). Järjestelmien muokkaaminen käyttäjätarpeiden mukaan on mahdollista vain vaihtamalla palveluita, muokkaamalla niiden liitoksia tai asetuksia tai muuttamalla palveluita (joita pääsee muuttamaan); tämä ei toteudu automaattisesti palveluita käyttämällä.
Miten arvioidaan	Hyötyä voidaan arvioida esimerkiksi, kuinka paljon samoja tietoja syötetään samaan tai useampaan järjestelmään, paljonko käyttäjälle näkyy virheitä ja kuinka virheet vaikuttavat työhön. ? Käytettävyydestä kertoo myös käyttäjän tekemien virheiden määrä (tiedon syöttövirheet, väärän tiedon perusteella tehdyt virheet), käyttäjän eri toimintoihin kuluttama aika tai yksinkertaisten käyttöliittymätoimintojen määrä? Käytettävyydessä arvioidaan myös vastako ratkaisun vuorovaikutteisuus käyttäjätarpeita, toimiiko ratkaisu kuten sen oletetaan toimivan (toiminnan oikeellisuus), työnkulkuihin käytettyä aikaa, prosessien läpimenoaikaa ja järjestelmän vasteaikoja? Lisäksi voidaan arvioida työnkulkujen muutoksista johtuvaa säästynyttä aikaa? Mittareita: Asiakas- ja käyttäjätyytyväisyyskyselyt, virheiden määrä, virhefrekvenssi, virheistä toipumisaika, muutospyyntöjen määrä, toteutettujen muutosten määrä, palvelun saatavuus, käyttäjien tekemien virheiden lukumäärä.

Tavoite	Tietotekniikan ja toiminnan lähentäminen
Kuvaus	Tavoitteeseen kuuluu se, että palvelut vastaavat käsitteellisesti todellisia prosesseja, tietokokonaisuuksia ja työnkulkuja tai niiden osia, jolloin toimialan asiantuntijat voivat tarkemmin tai itsenäisesti määritellä ratkaisuja. Palvelujen sisältämät prosessit tai kokonaisuudet vastaavat esim. asiakkaiden tai kumppanien kanssa tehtävää toimintaa, jolloin palveluja voidaan suoraan hyödyntää niissä.
Mihin perustuu	Verrattuna aikaisempiin teknisiin malleihin palvelut ja rajapinnat voidaan helpommin määritellä vastaamaan reaali maailman käsitteitä.
Edellytykset	Mikäli tähän pyritään, on tämä asetettava erityistavoitteeksi palveluiden ja prosessien suunnittelussa.
Haasteet	Toimialan asiantuntijatkaan eivät aina kykene ilmaisemaan todellisia tarpeita, tai ne tarkentuvat vasta kehitystyön yhteydessä. Tosi elämän kokonaisuuksissa ja prosesseissa on paljon poikkeuksia. Abstraktiotason nostaminen reaali maailman tasolle voi johtaa ratkaisuihin, joiden tarkkuustaso ei vastaakaan todellisia tarpeita. Taustalla olemassa olevien järjestelmien sisäiset mallit voivat vaikuttaa ratkaisuihin.
Miten arvioidaan	Tavoitteiden toteutumista arvioidessa tarkastellaan kuinka kohdealueen asiantuntijat tunnistavat tutut käsitteet ja prosessit palveluiden ja prosessien malleista? Arviointiin kuuluu onko ratkaisujen tuottamiseen tarjolla välineitä tai menetelmiä, joilla vähennetään teknisten toteutusvaiheiden tarvetta käyttöönotossa, voidaanko jäljittää polku kohdealueen toiminnasta vaatimukseen ja teknisiin ratkaisuihin sekä onko vaatimukset dokumentoitu yksikäsitteisesti? Arvioinnissa tarkastellaan myös onko saatavilla ollut ja onko käytetty yhteisiä terminologioita ja sanastoja ratkaisujen kuvaamiseksi? Arvioidaan myös projektissa/pilotissa mukana olevien osapuolten mielipiteet saavutetuista tuloksista: miten on voinut vaikuttaa, miten linjassa oman organisaation / yrityksen linjausten kanssa? Mittareita: Asiakastytyväisyyskysely, käsiteanalyysi käyttäjien ja järjestelmäratkaisujen välillä.

Toiminnallisiin ja asiakasorganisaation hyötyihin liittyy runsaasti taloudellisia mittareita. Näitä mittareita on kuvattu tarkemmin luvussa 5 "Web Services ja SOA: Taloudellinen arviointi". Taloudellisiin mittareihin liittyy mm. tulevien kassavirtojen muuttaminen nykyarvoksi (Present Value, PV), käyttämällä diskonttokorkoa. Lisäksi kassavirtojen analyyseissä arvioidaan investoinnin kuluja ja kustannuksia sekä saavutettavia hyötyjä. Kassavirta-analyysin mittareita ovat mm. nettonykyarvo (Net Present Value, NPV) ja/tai sisäinen korkokanta (Internal Rate of Return, IRR). Kassavirta-analyysin lisäksi voidaan käyttää takaisinmaksuaika-analyysiä (Payback Period), jossa lasketaan, missä ajassa investointi maksaa itsensä takaisin. Lisäksi yleinen taloudellinen mittari on sijoitetun pääoman tuotto (Return on Investment, ROI), joka ilmaisee pääoman tuoton prosentteina tarkasteltavalle aikavälille. Taloudellisia mittareita voi soveltaa myös useisiin sovellustuotannon ja teknisiin hyötyihin.

2.2 Hyödyt sovellustuotannon kannalta

Tavoite	Uusien palvelujen ja sovellusten nopea toteuttaminen
Kuvaus	Uusiin vaatimuksiin voidaan reagoida nopeasti, tuottamalla uusia palveluita, muuttamalla prosessien määrittelyä tai yhdistämällä palveluita uudella tavalla. Uusia sovelluksia voidaan toteuttaa nopeasti valmiista osista. Ratkaisujen käyttöönotto nopeutuu verrattuna aikaisempiin toimintamalleihin. Kun valmiita järjestelmien perusosia on valmiina saatavilla, voidaan keskittyä haastavampien, lisäarvoa tuottavien osien toteuttamiseen.
Mihin perustuu	Metadatan (rajapintatekniikoiden standardit, WSDL) rikas hyödyntäminen, joka mahdollistaa välineautomaation. Palvelujen uudelleenkäyttö. Uudelleenkäyttöä ja konfigurointia tukeva ohjelmistoprosessi, sovellusarkkitehtuuri ja välineet. Standardointi sikäli kuin se edistää teknistä yhteentoimivuutta tai yhteensopivien osien valmiina saatavuutta. Ratkaisun eri osien erottaminen siten, että niitä voidaan kehittää rinnakkaisesti erillään toisistaan tai siten, että esim. sama toiminnallisuus on hyödynnettävissä eri tietosisällöillä.
Edellytykset	Sovellusten modulaarisuus. Ohjelmistoprosessi ja käytännöt palvelupohjaiseen sovellusten tuottamiseen.
Miten arvioidaan	Hyötyä voidaan arvioida tutkimalla aikaa, joka tarvitaan uusien vaatimusten kirjaamisesta tuotantokäyttöön pääsemiseen sekä tarkastelemalla valmiina saatavien osien lukumäärää(+suhteellinen määrä), yhdisteltävyyttä ja soveltuvuutta uusiin käyttötilanteisiin. Ennen-jälkeen -arviointia voidaan käyttää kehittämisen eri vaiheissa ja muutostilanteissa sen arvioimiseen kuinka muutos on vaikuttanut ja onko se halutun tavoitteen suuntainen. Metadatan hyödyntämisestä voidaan välinetasolla tutkia, kuinka pitkälle toteutuksiin päästään automaattisesti rajapintakuvauksen avulla. Voidaan tarkastella millainen on dokumentaation määrä ja ymmärrettävyys. Rinnakkaisen kehittämisen mahdollisuuksia voi arvioida tutkimalla onko ratkaisun osat erotettu toisistaan riittävästi. Mittareita: Toteuttamiseen kuluva aika, kustannukset ja työmäärä.

Tavoite	Integroinnin tehostuminen kumppanijärjestelmiin
Kuvaus	Integraatioita voidaan toteuttaa entistä nopeammin, toistettavammin ja vähemmällä paikallisella sovitustyöllä. Uusia kumppaneita voidaan entistä helpommin sisällyttää yhteistyöketjuihin. Erityyppisiin tarpeisiin (EAI, EDI, B2B) tehtävien ratkaisujen yhdenmukaistuminen, jolloin osaamista, välineitä ja infrastruktuuria voidaan yhdenmukaistaa.
Mihin perustuu	Palvelukuvaukset ja rajapinnat. Tekniset ja sisällölliset standardit, yleisten internet-tekniikoiden käyttö. Samat perustekniikat ovat käytettävissä erityyppisiin tarpeisiin.
Edellytykset	Erilaisiin tarpeisiin vaadittavien teknisten ratkaisujen ominaisuuksien tunnistaminen. Riittävän yhdenmukaiset soveltamistavat.
Haasteet	Erityyppisiä soveltamistapoja on runsaasti, ja kukin niistä vaatii erilaista tekniikoiden soveltamista. On järkevää tunnistaa muutama perustapa erilaisiin tarpeisiin ja luoda tarkkoja ohjeistuksia niiden toteuttamiseen.

Miten arvioidaan	Integroinnin tehostumisesta tulevaa hyötyä voidaan tarkastella arvioimalla onnistuuko integrointi aikaisempaa vähemmällä työllä, onko ratkaistava vähemmän ongelmia kuin aikaisemmin ja kuinka nopeasti integraatio saadaan toteutettua? Voidaan myös arvioida kuinka nopeasti käyttöönottoprosessi etenee, seuraako integraation käyttöönotosta käyttökatkoja ja kuinka pitkiä sekä kuinka paljon paikallista sovitustyötä tarvitaan? Välineiden ja infrastruktuurin yhdenmukaistamisen arviointi: kuinka paljon on käytössä eri ympäristöjä ja alustoja? Ennen-jälkeen -arviointia voidaan käyttää myös käyttöönottoprosessien yhteydessä arvioimalla mm. käyttäjien osaamista, koulutustarpeita ja oppimisprosesseihin kuluvaan aikaa. Tilannetta voidaan tarkastella ennen käyttöönottoa ja jonkun ajan kuluttua käyttöönoton jälkeen (esim. puoli vuotta tai vuosi). Mittareita: Palvelun saatavuus, työmäärä, kustannukset ja tuotot, aikataulu.
-------------------------	---

Tavoite	Palveluiden ja komponenttien uudelleenkäyttö
Kuvaus	Kerran toteutettuja sovelluspalveluita voidaan hyödyntää useissa eri tuotteissa ja prosesseissa sekä eri asiakkailla. Samoja ominaisuuksia ei toteuteta toistuvasti (eri tavoin) eri tuotteisiin. Osa palveluista ja toiminnallisuudesta voidaan hankkia kumppaneilta tai ulkoistaa niiden toteuttaminen. Uudelleenkäytössä virheet voidaan korjata yhteen paikkaan, ja toistuvista toteutuksista johtuvat virheet vähenevät.
Mihin perustuu	Verkon yli käytettävät palvelut ovat uudelleenkäytettäviä. Rajapintamäärittely muodostaa sopimuksen, jolla palvelua voi käyttää eri tilanteissa. Palvelujen ja niitä käyttävien sovellusten välillä on vähemmän riippuvuuksia kuin aikaisemmissa integrointimalleissa.
Edellytykset	Palvelujen yleiskäyttöisyys. Sovelluskehitysprosessi, jossa hyödynnetään valmiita palveluita, mm. palvelujen varasto ja löytäminen. Yhteinen arkkitehtuuri ja kokonaisuuden hallinta.
Haasteet	Uudelleenkäytön suunnittelu lisää työmäärää ensimmäisessä toteutuksessa, vähentää vasta toistuvassa käytössä (tasapainoteltava tarkkojen paikallisten ratkaisujen aiheuttamien toistuvien kustannusten ja yleiskäyttöisyyden välillä). Huomioitava myös mahdollisia tulevaisuuden käyttötilanteita ja mukautettavuus- tai konfigurointivaihtoehtoja; ratkaistava kuinka toimitaan "melkein sopivien" palveluiden uudelleenkäytössä ja mukauttamisessa. Eri toimittajien palveluita käytettäessä määriteltävä vastuu "sovelluksen" kokoamisesta ja mahdollisista palvelujen käytön korvauksista.
Miten arvioidaan	Arvioinnissa tarkastelun kohteena voi olla palvelun koko (mitä suurempi toiminnallisesti, sitä vaikeampi yleisesti uudelleenkäyttää) ja vastaavuus tuettavan toiminnan kanssa vaikuttaa joustavuuteen. Uudelleenkäytettävyyden arviointi: kuinka paikallisia piirteitä palvelussa on toteutettu. Kuinka laajoja mukautus- tai konfigurointimahdollisuuksia on tarjolla (esim. parametointi), vai tarvitaanko koodin muokkaamista? Kenen hallinnassa käytettävät palvelut ovat? Millaisia paketointivaihtoehtoja käytettävät välineet ja suoritusympäristöt tarjoavat? Kuinka kehitys- ja suoritusympäristössä tuetaan uudelleenkäytettävien palvelujen löytämistä ja soveltuvuuden arviointia uudessa tilanteessa? Kuinka palvelun dokumentaatioissa on kuvattu käyttötilanteet ja käytön edellytykset? Uudelleenkäytön seurannassa selvitetään, kuinka monessa sovelluksessa tai prosessissa tiettyä palvelua on käytetty. Mittareita: Työmäärä, palveluiden koko, kuinka monessa eri tilanteessa / prosessissa tai sovelluksessa tiettyä ratkaisua on käytetty.

Tavoite	Inkrementaalinen kehittäminen
Kuvaus	Järjestelmiä voidaan kehittää vähitellen: ensin toteutetaan tärkeimmät ydin-osat (jotka saadaan nopeammin tuotantoon), ja uutta toiminnallisuutta lisätään vähitellen. Voidaan keskittyä yhteen osaan kerrallaan. Riskien havaitseminen ja kohdistus yksittäisiin osiin tehostuu.
Mihin perustuu	Palvelupohjainen toimintamalli perustuu osista vähitellen koostamiseen.
Edellytykset	Sovellusten jakaminen palveluiksi. Inkrementtien suunnittelu ja projektinhallinta.
Haasteet	Jos osat tulevat eri lähteistä, arkkitehtuuri ja kokonaisuuden hallinta korostuvat.
Miten arvioidaan	Hyötyä arvioitaessa tarkastellaan sitä onnistuuko vähittäinen kehittäminen hankkeessa, miten suoritussympäristö tukee uusien osien tai versioiden lisäämistä saumattomasti tuotantoon (hot deployment)? Jos on tarpeen käyttää useita yhtäaikaista versioita samoista palveluista, arvioidaan kuinka se onnistuu? Arviointi kohdistuu myös siihen, miten käytettävät välineet ja kehitysprosessit tukevat inkrementtien ja versioiden hallintaa? Kokonaisuuden modulaarisuutta arvioidaan sen perusteella, ovatko rajapinnat selkeästi kuvattuja (vastuut) ja onko osien välillä mitään muita riippuvuuksia kuin rajapintojen kautta (arkkitehtuurisuunnittelu). Arvioidaan myös palvelupohjaisen kokonaisuuden hallintavälineiden saatavuus ja ominaisuudet? Inkrementaaliseen kehittämiseen sopii formatiivinen arviointi, koska se tarjoaa tietoa, jota käytetään tuotteen kehittämisessä ja sen tulokset ovat yleensä voimassa vain kohtalaisen lyhyen aikaa. Arvioinnin tarkoituksena on tällöin antaa tietoa ja tukea niille, jotka tekevät parannuksia esim. kehitystyön kohteena olevaan prosessiin tai järjestelmään. Mittareita: Sovelluksen koko (LOC, sovelluspalveluiden/komponenttien lukumäärä, transaktioiden lukumäärä), uusien osien käyttöönoton ja kehittämisen työmäärä ja kustannukset, versionhallinnan kustannukset ja työmäärä, osien riippuvuuksien lukumäärä (ja tyytit).

Tavoite	Kehitysympäristöjen valinnanvara
Kuvaus	Ratkaisujen tuottamiseen voidaan valita standardeja tukevia välineitä, joilla tehdyt ratkaisut toimivat yhteen muilla välineillä tehtyjen ratkaisujen kanssa. Välineet voivat olla edullisia tai ilmaisia (usein keskittyen vain palvelujen toteutukseen), tai tarjota edistyneitä mallinnus-, seuranta- tai hallintaominaisuuksia esim. sovellusarkkitehtuurin eri osiin tai projektinhallintaan. On mahdollista valita integroitu kehitysympäristö tai yhdistellä erityyppisiä välineitä.
Mihin perustuu	Avoimet tekniikat.
Edellytykset	WS-I-tyyppiset tekniikoiden soveltamisohjeet, joita tuetaan välineissä ja soveltamistavoissa.
Haasteet	Palvelutekniikoiden peruspiirteissä yhteentoimivuus voidaan saavuttaa yhtenäisten soveltamisohjeiden avulla, mutta erityisesti edistyneemmissä ja uusissa määrittelyissä eri välineet eivät ole yhteensopivia.

Miten arvioidaan	Näiden välineiden arvioinnissa tarkastellaan kehitys- ja suoritussympäristön hankinta- ja ylläpitokustannuksia. Hankintavaiheessa arvioidaan välineiden tukemia standardeja ja profiileja. Tarkemmin yhteentoimivuutta muilla välineillä tehtyjen ratkaisujen kanssa arvioidaan testaamalla niitä yhdessä. Paikallisia ja projektikohtaisia vaatimuksia voi verrata ja arvioida mallinnus-, seuranta- ja hallintaominaisuuksien osalta välineissä. Standardoidut mallien siirtoformaattit, yleiskäyttöisiin kehitysympäristöihin (IDE) integroituvat moduulit ja standardoidut suoritussympäristöt (esim. J2EE, .NET) helpottavat eri välineiden käyttöä ja yhdistelyä. Integroitujen ympäristöjen arvioinnissa tutkitaan, kuinka paljon tarvittuja piirteitä kuuluu valmiina kehitysympäristöön. Mittareita: Saatavilla olevien kehitysympäristöjen lukumäärä, hinta ja ominaisuudet, kehitys- ja suoritussympäristöjen hankinta- ja ylläpitokustannukset.
-------------------------	---

Tavoite	Teknologian keveys ja helppo opittavuus
Kuvaus	Palvelujen toteuttamisessa ja liittämässä käytettävät tekniikat ovat helposti opittavissa ja liitettävissä jo käytettyihin välineisiin. Kehittämiseen on saatavilla myös halpoja ja ilmaisia välineitä, jotka madaltavat markkinoille pääsyn kynnyksiä.
Mihin perustuu	Välineautomaatio, teknisimpien yksityiskohtien piilottaminen kehittäjältä, laaja tuki eri väline- ja infrastruktuuritoimittajilta.
Edellytykset	Pitäytyminen laajasti välinetoimittajien tukemissa soveltamistavoissa. Tehokkaiden ja edistyneiden välineiden saatavuus ja käyttö.
Haasteet	Tekniset yksityiskohtat ovat monimutkaisia, ja myös niitä voidaan käsitellä. Teknologia ei ole "kevyttä" esim. tiedonsiirron ja suorituskäytön suhteen, koska XML-merkkijonomuotoisessa tiedonsiirrossa on laajasti esitettyä tietoa.
Miten arvioidaan	Opittavuutta voidaan arvioida oppimiseen käytetyllä ajalla ennen kuin saavutetaan osaamistaso, jolloin vasta tulokset syntyvät. Myös koulutuksen saatavuutta, saatavilla olevan materiaalin määrää, laatua ja saatavuutta voidaan arvioida. Teknologian keveyden osalta voidaan arvioida, kuinka monia erilaisia yksityiskohtia kehittäjä joutuu käsittelemään ratkaisujen kehittämisessä, toteutuksessa tarvittavien työvaiheiden määrää ja eri vaiheisiin tarvittava aikaa. Tietoliikenteen ja palveluiden kuormitustestauksilla puolestaan arvioidaan järjestelmän suorituskäytön. Mittareita: Työmäärä, sovelluksen koko, oppimiseen käytetty aika, koulutuksen ja tiedon saatavuus, kuormitustestaus, kehittäjäkyselyt.

2.3 Tekniset hyödyt

Tavoite	Tekninen joustavuus
Kuvaus	Perustekniikoita tukeva monet eri alustat ja toteutusvälineet, joita tuetaan laajasti. Eri laitteistoilla, käyttöjärjestelmillä ja ohjelmointikielillä toteutettuja palveluita voidaan yhdistellä. Uusien teknisten ratkaisujen helppo yhdistäminen olemassa oleviin. SOA:ssa voidaan pyrkiä myös tekniikoiden piilottamiseen arkkitehtuurissa.
Mihin perustuu	Käytettävien avointen tekniikoiden ja niiden soveltamistapojen sopiminen. Ks. myös eri tekniikoilla tehtyjen sovellusten ja palvelujen liittäminen. Eri tekniikoita (esim. salaus, tekniset reititys- ym. seikat, kun ratkaisua laajennetaan esim. organisaatioiden väliseksi) voi lisätä ratkaisuun ilman, että se vaikuttaa toiminnallisiin rajapintoihin. Jos pyritään toteuttamaan SOA:a tekniikkariippumattomasti, on piilotettava myös rajapintatekniikat.
Edellytykset	Määritelty taso, jolla joustavuutta tavoitellaan. Jo WSDL/SOAP-standardeilla saavutetaan riippumattomuus toteutustekniikoista ja palvelujen sijainnista. Täyden tekniikkariippumattoman (riippumattomuus myös ratkaisun arkkitehtuurillisista ominaisuuksista tai tarkoista rajapintojen merkityksistä) ratkaisun tavoittelussa voidaan käyttää mallipohjaista lähestymistapaa (esim. MDA), ja on rakennettava abstrakti eristyskerros, joka piilottaa tekniikat.
Haasteet	Käytännössä tavoitteeksi riittää yleensä yhdenmukainen, riittävän joustava tekninen arkkitehtuuri, ei suurin mahdollinen joustavuus. Mallipohjaisten lähestymistapojen erityyppisiä soveltamistapoja paljon. Valittava, millaista joustavuutta halutaan, esim. missä vaiheessa sidonta (esim. rajapintakuvaukseen ja palvelun verkko-osoitteeseen) tapahtuu. Joustavuus lisää monimutkaisuutta.
Miten arvioidaan	Arvioinnissa voidaan tarkastelemalla ovatko ratkaisun eri osat (esim. tiedot, toiminnot, turvallisuus, viestinvälitys) erotettu toisistaan siten, että niitä voi muuttaa erillään toisistaan? Tarkastelun kohteena on myös voidaanko eri osia helposti tehdä eri tekniikoilla, mikä on tarvittavien väliohjelmistojen ja ratkaisun komponenttien vähäinen määrä (helppo yhdistettävyyttä), onko lisälajennuksia saatavissa joustavasti valmiisiin järjestelmiin ja onko tapa soveltaa tekniikoita määritelty yhdenmukaisesti? Mittareita: Saatavilla olevien välineiden lukumäärä ja ominaisuudet, ratkaisun eri osien riippuvuudet, ratkaisun teknisten osien määrä.

Tavoite	Infrastruktuurin uudelleenkäyttö
Kuvaus	Olemassa olevat verkot, palvelimet ja alustatuotteet ovat käytettävissä eri sovelluksissa ja prosesseissa. Ei tarvita erikseen hankittavia alustatuotteita, joita on tuettava kaikissa osallistuvissa sovelluksissa. Erityyppisissä ratkaisuissa voidaan soveltaa esim. samoja turvallisuusratkaisuja.
Mihin perustuu	Web-sivuja varten (ennestään) käytetyt verkot, palvelimet ja ohjelmistot ovat hyödynnettävissä sovellusten kommunikaatiossa ja suoritusympäristössä.
Haasteet	Perusominaisuuksissa infrastruktuuri on hyvin uudelleenkäytettävää. Yhdenmukainen hallinta, seuranta yms. vaativat kuitenkin usein ei-standardoituja ja tuotekohtaisia laajennuksia. Joissain tilanteissa uuden infrastruktuurin hankinta on kuitenkin tarpeellista, esim. ylläpidon yksinkertaistamisen tai toimintavarmuuden parantamisen vuoksi.

Miten arvioidaan	<p>Hyötyä arvioitaessa tutkitaan onko hankkittava uusia verkkoja, palvelimia tai alustoja ja toimivatko tehdyt palvelut jo käytetyllä infrastruktuurilla? Summatiivista arviointia voidaan käyttää infrastruktuurin uudelleenkäytön arvioinnissa, sillä sen tarkoituksena on tuottaa tietoa tilanteisiin, joissa päätetään palvelun tai toimintamallin jatkamisesta ja antaa arvio uudelleenkäytön vaikutuksista ja tehokkuudesta. Arvioinnissa on tarkoitus myös tarjota tietoa tuotteen ominaisuuksista.</p> <p>Mittareita: Työmäärä, uudelleenkäytöstä aiheutuvat kustannukset / säästöt, uudelleenkäytöstä aiheutuvien virheiden lukumäärä.</p>
-------------------------	---

Tavoite	Välineautomaatio ja itsensä kuvaavat rajapinnat
Kuvaus	Kehitys- ja integrointivälineet generoivat rajapintakuvauksista automaattisesti rungon toteutuksille eri järjestelmiin, tai toteutuksista saadaan helposti tuotettua rajapintakuvaukset. Ratkaisujen kehitys ja markkinoillepääsy nopeutuu. Palvelun käyttäjälle riittää rajapintakuvauksen saanti palvelun käyttämiseksi.
Mihin perustuu	Palvelu- ja rajapintakuvaukset, joissa on metadatan rikas käyttö. Välineet, joilla rajapintakuvauksesta tuotetaan tietylle tekniikalle toteutusta varten rungot. Rajapinnan yksinkertainen dokumentaatio voi myös olla osana itse rajapintakuvausta. Suuri osa "itsensä kuvaamisesta" muodostuu operaatioiden ja parametrien ymmärrettävästä nimeämisestä. Itsensä kuvaavuutta voidaan lisätä käyttämällä koneellisesti käsiteltäviä koodistoja ja sanastoja.
Edellytykset	Rajapintamäärittelyt kuvaavat yleensä vain rajapintojen syntaksin, eivät eri tietojen tai toimintojen merkityksiä, joiden dokumentoinnista on huolehdittava.
Haasteet	Monimutkaisten sisältöjen (laajat skeemat) käyttö on järkevämpää dokumentti- kuin generointi-tyyppisesti. Sisällöllisiin ja toiminnallisiin muutoksiin reagointia ei voi automatisoida helposti. Rajapintamäärittelystä generoidut ratkaisut sisältävät perustuvat vain syntaksiin, lisäksi on sovittava ja kuvattava merkitys.
Miten arvioidaan	<p>Tavoitteen saavuttamista voidaan arvioida sillä, onko välineissä generointia tukevia ominaisuuksia ja ovatko generoidut palvelurajapinnat yhteensopivia muilla välineillä tehtyjen kanssa? Kuinka kauan kestää toteutuksen tekeminen? Lisäksi voidaan arvioida sitä, miten paljon dokumentaatiota tarvitaan rajapintamäärittelyn lisäksi palvelun ymmärtämiseksi ja käyttämiseksi?</p> <p>Mittareita: määrittelyistä toteutuksiin kuluva aika ja työmäärä, toimivatko eri välineillä tehdyt yhdessä, rajapintojen lisäksi tarvittavan lisädokumentaation määrä.</p>

Tavoite	Eri tekniikoilla ja alustoilla toteutettujen sovellusten ja palvelujen liittäminen
Kuvaus	Samoja liitäntätekniikoita voidaan käyttää sekä sisäisten että ulkoisten (kumppani)liitäntöjen käyttöön. Web services-tekniikoilla voidaan myös sitoa rajapinnat muihin kuin esim. http- ja SOAP-protokolliin esim. tehokkuussyistä.
Mihin perustuu	Avoimet tekniikat. Yksinkertainen perusmekanismi sovellusten liittämiseen, jolla saavutetaan riippumattomuus palveluiden sijainnista, toteutuksessa käytetyistä tekniikoista, ja yhä enemmän myös esim. käyttöliittymässä käytettävistä laitteista. Rajapinnat piilottavat järjestelmien tekniset toteutusyksityiskohdat. Tekniikoissa erotettu toisistaan siirtoprotokollat laajennuksineen ja rajapintamäärittelyt.
Haasteet	Käytännössä vain web-tekniikoihin (http, XML) tapahtuvat sidonnat on standardisoitu, joten muut tekniikkavaihtoehdot ovat alustakohtaisia tai sopivia organisaation sisäiseen käyttöön.
Miten arvioidaan	Saavutettavan hyödyn arvioinnissa selvitetään onko yhteistyökumppaneilla samat liitäntätekniikat käytössä, tukevatko käytetyt välineet ja alustat useita vaihtoehtoisia liikennöinti- ym. ratkaisuja ja paljastavatko rajapinnat järjestelmien sisäisiä teknisiä yksityiskohtia? Mittareita: vastaukset yllä kuvattuihin kysymyksiin.

Tavoite	Sovelluspalvelujen ja niiden hyödyntäjien välinen löysä kytkentä
Kuvaus	Palvelut voidaan löytää tarvittaessa saatavilla olevien palvelujen varastosta, tai jopa dynaamisesti ajon aikana. Niitä voidaan kehittää itsenäisesti ja päivittää ilman laajoja vaikutuksia kokonaisjärjestelmän muihin osiin, mikä tehostaa ylläpitoa ja joustavuutta. Yhteyden molemmissa päissä ei tarvita samaa kehitystekniikkaa, komponenttimallia tai kehystuotetta. Ratkaisujen uudelleenkäytettävyys paranee. XML mahdollistaa ratkaisujen laajentamisen eri tasoilla.
Mihin perustuu	Myöhäinen sidonta, hakemistopalvelut, osoitemäärittelyt. Web services-tekniikoilla tehdyt ratkaisut ovat löysemmin toisiinsa liittyviä (mm. http-tekniikan ansiosta) kuin monet aikaisemmat väliohjelmistot.
Haasteet	Palvelujen sijainnin tai rajapinnan syntaksin selvittäminen ajon aikana ei millään tavoin takaa yhteensopivuutta. Yleiset palveluhakemistot, joista palvelukuvausten perusteella voitaisiin valita yhteistyökumppaneita, eivät ole yleistyneet.
Miten arvioidaan	Tavoitteen toteutumisessa arvioidaan onko palvelut kytketty löysästi toisiinsa => ovatko aina kaikki tarvittavat tiedot siirrossa mukana, käytetäänkö vain tunnisteita, onko tietotyypit ja koodistot tiukasti määritellyt, onko järjestelmien väliset kutsusuhteet ja monenvälisyydet tiukasti määritellyt, onko palveluissa tietoa hyödyntäjäkohtaisesta tilasta? Arviointiin liittyy myös nojaututaanko yhteisten hakemisto- tms. palveluiden käyttöön? Mittareita: ovatko kaikki tiedot mukana, käytetäänkö pelkkiä tunnisteita, ovatko tietotyypit määritellyt ja yhdenmukaisia, onko kutsusuhteet tiukasti määritellyt (mikä kriteeri - onko kuvattu peräkkäisprotokolla esim. toisiaan seuraavat operaatiot, onko järjestelmien vastuut kuvattu tarkasti ja siten että ne ovat aina samat), ovatko palvelut tilallisia (ja mitä tilaa ne säilyttävät), oletetaanko suoritusympäristössä yhteisiä teknisiä palveluita.

Tavoite	Globaalit tekniset web services-standardit
Kuvaus	Rakennetut ratkaisut ovat helpommin liitettävissä ja sovitettavissa ratkaisuihin eri maissa. Välinetuki on laajaa.
Mihin perustuu	Teknisiä standardeja hyödynnetään samalla tavoin eri välineissä, sovelluksissa ja maissa. Esimerkkeinä XML-standardit, WSDL, SOAP, http jne.
Haasteet	Standardit ovat yhdenmukaisia lähinnä teknisellä tasolla; siirryttäessä toiminnalliselle tasolle yhtä laajaa yhdenmukaistamista ei ole tehty. Uudet tekniikat eivät tule korvaamaan jo tehtyjä ratkaisuja, vaan voivat täydentää niitä.
Miten arvioidaan	Hyötyä arvioitaessa tarkastellaan sitä, ovatko teknisten standardien hyödyntämistavat määritelty ja yhdenmukaisia kumppanien kanssa (esim. WS-I)?

Tavoite	Järjestelmien hajautus
Kuvaus	Järjestelmien toteuttamisessa voidaan hyödyntää verkon yli eri palvelimille sijoitettuja palveluja, joilla voidaan sijoittaa tietojenkäsittelykapasiteettia sinne, missä sitä tarvitaan, lisätä järjestelmien skaalattavuutta, luotettavuutta sekä palvelu useita yhtäaikaista käyttäjiä.
Mihin perustuu	Viestipohjainen kommunikointi palvelujen ja niitä käyttävien ohjelmistojen välillä. Internet-tekniikoiden hyödyntäminen. Palvelimet, kuormanhallintaa tukeva infrastruktuuri, ym.
Haasteet	Kokonaisuuden hallinta ja vastuiden määrittely, "luotettujen palvelujen" määrittely, sopimukset, suorituskyky.
Miten arvioidaan	Tavoitteen onnistumista arvioitaessa pohditaan tukevatko palvelut ja niiden käyttämä infrastruktuuri kuormanhallintaa ym. ominaisuuksia? Mittareita: Luotettavuus- ja skaalattavuusmittarit, kuormitustestaus, järjestelmän saatavuus, käyttäjätyytyväisyyskyselyt etenkin suorituskyvyn osalta, järjestelmien virhelokit jne.

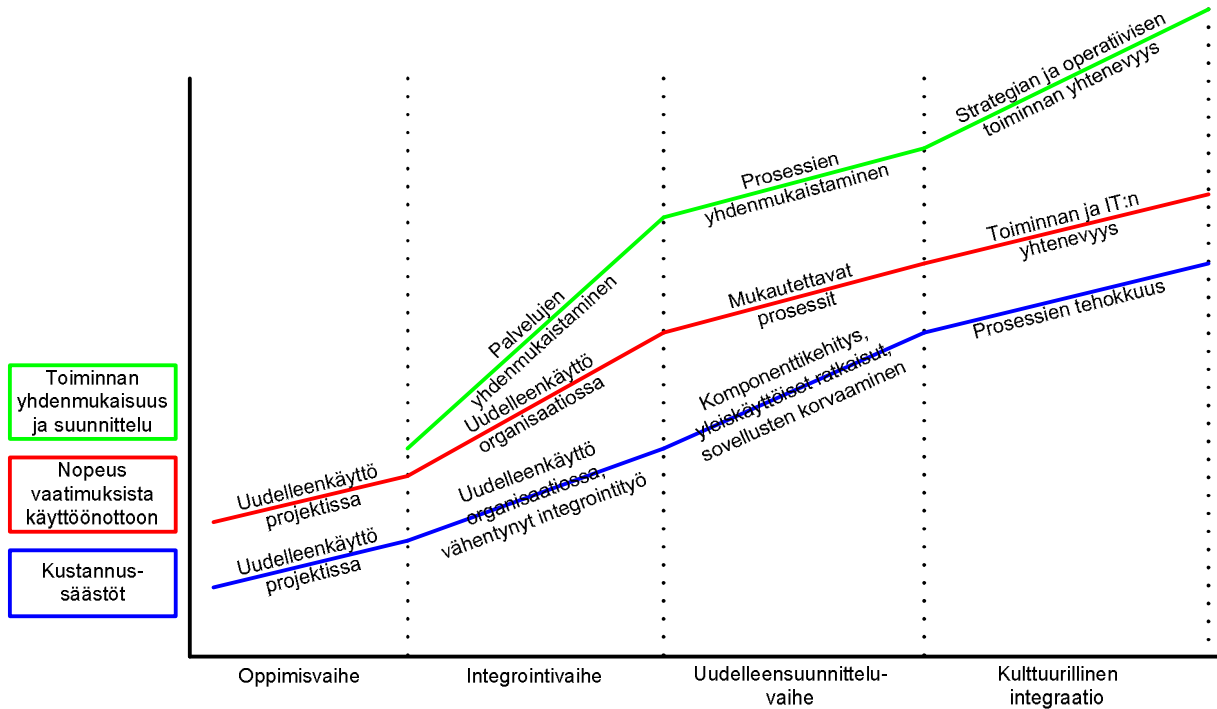
2.4 Palvelupohjaisen arkkitehtuurin kehitysvaiheet

Palvelupohjaiseen arkkitehtuuriin siirtyminen tapahtuu yleensä melko pitkällä aikavälillä ja myös siitä saatavat hyödyt jakaantuvat pitkälle aikavälille. Palvelupohjaisessa arkkitehtuurissa on havaittavissa eri kehitysvaiheita, joiden edetessä saavutetaan palvelupohjaisen arkkitehtuurin hyötyjä vähitellen. Seuraavassa on esitelty yleiskuva eri kehitysvaiheissa saavutettavista eduista. (Spratt 2006)

1. Oppimisvaiheessa (I) saavutettavat hyödyt ovat lähinnä projektitasolla tapahtuvasta uudelleenkäytöstä saatavia hyötyjä, joita voidaan saavuttaa projektien sisällä tai eri projektien välisessä yhteistoiminnassa.
2. Integraatiovaiheessa (II) organisaatio saa kokoon kriittisen massan standardoituja palveluja, jolloin se pystyy hyötymään laajasta palvelujen uudelleenkäytöstä sekä alenevista integraatiokustannuksista.
3. Uudelleensuunnitteluvaiheessa (III) pystytään kehittämään toimintaprosessien ominaisuuksia, jolloin aiemmin toteutettujen sovelluspalvelujen kokoelmaa uudistetaan tehostamalla ja komponenttoimalla. Tällä kypsyyssasteella hyötyjä saavutetaan prosessien paremmasta yhtenäisyydestä ja muokattavuudesta.
4. Viimeinen kypsyyssaste on kulttuurillisen integraation vaihe (IV), jossa saavutetaan korkean tason organisatorinen palvelukonsepti. Tällöin on pystytty parantamaan korkean tason liiketoiminta- ja

tietotekniikkaprosessien tehokkuutta ja mahdollistetaan strategisia hyötyjä. Tällaisia hyötyjä ovat esim. liiketoimintastrategian tekeminen siten, että ennustettavissa olevissa muutoksissa suunnittelu voidaan tehdä nopeasti, nopea reagointi tunnistettuihin riskeihin ja kustannustehokkuuden parantaminen.

Kuvassa 2 on esitetty palvelupohjaisen arkkitehtuurin hyötyjä eri kehitysvaiheissa, ja miten saavutettava kokonaishyöty kasvaa vaiheiden edetessä (Spratt 2006).



Kuva 2 Palvelupohjaisen arkkitehtuurin hyötyjen realisoituminen eri kypsyyssvaiheissa.

Arvioitaessa web-sovelluspalvelu- ja SOA-investointien kannattavuutta on tärkeää, että osataan arvioida, milloin eri hyötyjä voidaan saavuttaa ja mikä on niiden rahallinen arvo. Edellä esitetty yleiskuva antaa mallin siitä, missä vaiheissa eri hyötyjä on mahdollista saavuttaa ja auttaa siten arvioimaan taloudellisia hyötyjä. Mitä paremmin saavutettavat hyödyt ja niiden realisoitumisajankohda osataan arvioida, sitä paremmin voidaan tehdä paikkaansa pitäviä kannattavuuslaskelmia.

2.5 Toiminnalliset hyödyt, sovelluskehityksen hyödyt ja tekniset hyödyt eri kypsyyssvaiheissa

Tässä kappaleessa kuvataan palvelupohjaisesta arkkitehtuurista saatavia hyötyjä arkkitehtuurin kypsyyssasteiden näkökulmasta. Palvelupohjaisen arkkitehtuurin hyödyt jakautuvat pitkälle aikavälille, ja siksi aluksi tarkastellaankin hyötyjen realisoitumista eri kypsyyssvaiheissa. Tämän jälkeen käydään läpi yksityiskohtaisemmin, mitä toiminnallisia sovelluskehityksen ja teknisiä hyötyjä web-sovelluspalveluilla ja SOA:lla voidaan saavuttaa. Arvioitaessa sijoitetun pääoman tuottoa tulee laskeissa ottaa huomioon nämä mahdollisesti saavutettavat hyötynäkökohdat. Hyötyjen arvioinnissa on otettava huomioon kenelle hyödyt kohdistuvat, pystytäänkö kyseisiä hyötyjä saavuttamaan ja mikä on niiden rahallinen arvo. Luetelluista hyödyistä toiminnalliset hyödyt kohdistuvat pääasiassa järjestelmää käyttävälle organisaatiolle, kun taas sovelluskehityksen hyödyt koskevat pääasiassa ohjelmistotuottajaa. Tekniset hyödyt kohdistuvat sekä palvelujen toteuttajille että käyttäjille mahdollistaen toiminnallisia ja sovelluskehityksen hyötyjä. Hyötyjä käsitellään tarkemmin SerAPI-

projektin "SOA ja Web Services - tavoitellut hyödyt" luvussa sekä "SOA ja Web services-hyötyjen arviointi ja mittaus" luvussa. Kyseisissä luvuissa selvitetään se, mihin eri hyödyt perustuvat, mitä edellytyksiä ja haasteita niillä on sekä se, miten eri hyötyjä voidaan arvioida.

Hyötyjen kohteen lisäksi on huomioitava, että kaikkia hyötyjä ei saavuteta heti vaan osa hyödyistä voidaan saavuttaa vasta pitemmällä aikavälillä, jos silloinkaan. Pitemmän aikavälin hyödyt realisoituvat vain, jos arkkitehtuuria saadaan kehitettyä vaiheeseen, joka mahdollistaa hyötyjen saavuttamisen (ks. luku 5.2). Seuraavassa esitettyjä toiminnallisia hyötyjä sekä sovelluskehityksen ja teknisiä hyötyjä arviotaessa tulee miettiä, missä vaiheessa päästään millekin kypsyystasolle ja mitä hyötyjä näillä kypsyystasoilla voidaan saavuttaa sekä mikä on niiden rahallinen arvo. Tämän jälkeen näitä tietoja voidaan soveltaa kannattavuuden arvioinnissa eri investoinnin kannattavuuden mittareilla. Alla olevat hyödyt on suhteutettu palvelupohjaisen arkkitehtuurin eri kehitysvaiheisiin (Sprott 2003) (I - oppimisvaihe, II - integraatio, III - uudelleensuunnittelu, IV - kulttuurillinen integraatio). Lisäksi on esitetty, kohdistuvatko hyödyt kustannussäästöihin tai tuottoihin (**H** - hinta), nopeaan mukautuvuuteen (**A** - aika) vai toiminnan yhdenmukaistamiseen ja kehittämiseen (**Y** - yhdenmukaisuus).

Seuraavassa esitettyjä Web Servicen ja SOA:n hyötynäkökohtia on käsitelty soveltaen edellä mainittua tavoitellut hyödyt lukua artikkelissa (Samtani ym. 2002) mainittuihin hyötyihin.

Toiminnalliset hyödyt:

- Dynaamiseen liiketoimintaan osallistuminen
 - o Nopea reagointi uusiin vaatimuksiin ja erilaisiin muutoksiin (IV, H+A+Y)
 - o Sovellusten parantunut liitettävyys (II, H+A+Y)
 - o Standardinmukaisuus parantaa hankintamahdollisuuksia (II, H+Y)
- Yhteisliiketoiminnassa toimiminen
 - o Toiminnallinen joustavuus (III, A+H)
 - o Nopea reagointi uusiin kumppanuuksiin (IV, A+H)
 - o Parantunut järjestelmäosien hankinta (II, H+A)
 - o Sovelluksiin hankittavien ominaisuuksien ja liitäntöjen alentuvat kustannukset (II, H+A+Y)
 - o Uusien toimijoiden helppo lisättävyys järjestelmään (III, A+H)
 - o Eri paikoissa olevien uusien toimintojen ja toimijoiden helppo lisääminen (III, A+H+Y)
 - o Eri järjestelmät ja palvelut helpommin liitettävissä toimiviksi kokonaisuuksiksi (II, A+H+Y)
- Parempi ja edullisempi asiakaspalvelu (III, Y+H)
- Parantunut tehokkuus toimintaprosessien etenemisessä ja automatisoinnissa
 - o Palvelut vastaavat osittain tai kokonaan todellisia prosesseja, tietokokonaisuuksia ja työnkulkuja (III-IV, Y+A)
 - o Toimintaprosessien määrittäminen ja muuttaminen keskitetysti (III-IV, A+H+Y)
 - o Tehtävien ja prosessien koostaminen ohjelmistopalveluista (III, H+A+Y)
 - o Prosessien määrittelyyn ja muokkaamiseen ei vaadita ohjelmointitaitoja (IV, H+A)
 - o Prosessien yhdenmukaistaminen, seuraaminen ja automatisointi (III, Y+H)
 - o Prosessien jakaminen eri yksiköille ja kumppaneille (III, A+Y+H)
 - o Järjestelmän kehittäminen asteittain organisaation toiminnan ja sen kehittämisen mukaan (II, A+H)
 - o Vähentää paikallista räätälöintiä (I, A+H+Y)
- Nopeampi uusien palveluiden ja piirteiden toteuttaminen
 - o Olemassa olevien järjestelmien ja verkkojen hyödyntäminen (I, A+H+Y)
 - o Olemassa olevien palveluiden ja sovellusten uudelleenkäyttö (I, A+H+Y)
 - o Palveluiden hyödyntäminen useissa tehtävissä ja prosesseissa (II, A+H+Y)

- Järjestelmien parantunut käytettävyys
 - o Käyttäjien päällekkäisen työn väheneminen (I, A+H+Y)
 - o Parantunut vastaavuus työn ja ohjelmiston välillä (IV, Y+H+A)
 - o Eri sovellusten tietojen ajanmukaisuus (I, Y+A+H)
 - o Yhdenmukaisten näkymien ja toiminnallisuuksien tarjoaminen (III, Y+H)
 - o Järjestelmien helpompi muokkaaminen ja personointi (III, A+H)
- Parantunut järjestelmäympäristön hallinta ja ylläpito
 - o Järjestelmä jaettavissa helposti hallittaviin osiin (III, H+Y+A)
 - o Järjestelmän kehittäminen ja seuraaminen voidaan tehdä osa kerrallaan (II, A+H)
 - o Tuotantoympäristön tekniseen seurantaan saatavilla hallintatyökaluja (II, Y+H)
 - o Tuotantoympäristön toiminnalliseen seurantaan ja hallintaan saatavilla hallintatyökaluja (III, Y+H)
 - o Päällekkäisten tietojen ja toimintojen ylläpidon väheneminen (I, A+H+Y)
- Muut toiminnalliset hyödyt

Hyödyt sovellustuotannon kannalta:

- Ohjelmistokehityksen automatisointi
 - o Kehitysvälineillä rajapintakuvauksista automaattisesti rungot toteutuksille (I, A+H)
 - o Kehitysvälineillä toteutuksista helposti tuotettavat rajapintakuvaukset (I, A+H)
 - o Palvelun käyttäjälle riittää rajapintakuvaus palvelun käyttämiseksi (I, A+H+Y)
- Uusien palvelujen ja sovellusten nopea toteuttaminen
 - o Uusiin vaatimuksiin reagoiminen luomalla uusia palveluja, muuttamalla prosessien määrittelyjä tai yhdistämällä palveluja uudella tavalla (III, A+H)
 - o Uusien sovellusten luominen olemassa olevista osista (II, H+A)
- Integroinnin tehostuminen kumppanijärjestelmiin (II, H+A+Y)
- Palveluiden ja komponenttien uudelleenkäyttö
 - o Sovelluspalvelujen hyödyntäminen eri prosesseissa ja tuotteissa (II, H+A)
 - o Kerran luotujen ominaisuuksien uudelleenkäyttö (I, A+H+Y)
 - o Palveluiden hankinta kumppaneilta tai palvelutoteutusten ulkoistaminen (II, Y+A+H)
 - o Toistuvien virheiden väheneminen ja virheiden korjaus yhteen paikkaan (I, A+H)
- Inkrementaalinen kehittäminen
 - o Järjestelmien asteittainen kehittäminen (II, H+Y)
 - o Ensin kehitetään järjestelmän ydinosat, jonka jälkeen voidaan toiminnallisuutta lisätä vähitellen (I, Y)
 - o Kehityksen ja riskien keskittyminen tiettyyn osaan kerrallaan (I, H+Y)
- Kehitysympäristöjen valinnanvara
 - o Yhteentoimivuus teknisiä standardeja tukevilla välineillä (I, H+Y)
 - o Myös edullisia ja ilmaisia kehitysvälineitä (I, H)
 - o Edistyneitä mallinnus-, seuranta- ja hallintaominaisuuksia (III, Y+H+A)
 - o Voi yhdistellä eri välineitä tai käyttää integroitua kehitysympäristöä (II, Y+H)
- Teknologian keveys ja helppo opittavuus
 - o Palvelujen toteutuksen ja liittämisen helppo opittavuus (II, A+H)
 - o Halvat ja edulliset kehitysvälineet helpottavat markkinoille pääsyä (II, H+A+Y)

Tekniset hyödyt:

- Väliohjelmistoteknologian virtaviivaistuminen (I, A)
- Standardiperustaisen integraation käyttö
 - o Laajasti tuetut alustat ja toteutusvälineet (II, Y+H)
 - o Eri tekniikoilla toteutettujen palvelujen yhdistäminen (II, H)
 - o Erilaiset alustaratkaisut mahdollisia (I, A+H)
 - o Mahdollistaa olemassa olevien verkkojen, palvelimien ja alustatuotteiden käytön (I, H)
 - o Eri ratkaisujen helppo liitettävyyys eri maissa (IV, Y+A+H)
 - o Laaja välinetuki (II, H+A)
- Sovellusten ja toimintaprosessien välinen integraatio
 - o Samat liitäntätekniikat sisäisille ja ulkoisille liitäntöille (II, Y+H)
 - o Rajapinnat voidaan tarvittaessa sitoa myös muihin kuin tavallisimmin käytettyihin protokolleihin (III, H)
 - o Palveluiden hakeminen saatavilla olevista varastoista helppoa (II-III, A+H)
 - o Palveluiden etsiminen mahdollista myös dynaamisesti ajon aikana (I, H)
 - o Mahdollistaa palveluiden itsenäisen kehittämisen ja päivittämisen ilman laajoja vaikutuksia järjestelmän muihin osiin (III, A+H)
 - o Osapuolien välillä ei tarvita samaa kehitystekniikkaa, komponenttimallia tai kehystuotetta (I, A+H)
 - o XML mahdollistaa ratkaisujen laajentamisen (I-III, A+H)
- Ei ohjelmistokoodin monistamista => lisää uudelleen käytettävyyttä (I, A+H+Y)
- Järjestelmien hajautus
 - o Eri palvelimilla olevien palvelujen käyttö verkon yli (I, H+A)
 - o Useiden asiakkaiden yhtäaikainen palvelu (I, H)

Web-sovelluspalveluilla ja palvelupohjaisella arkkitehtuurilla voidaan saavuttaa monia hyötyjä, joita on vaikea saavuttaa muilla tekniikoilla. Monien hyötyjen saavuttaminen vaatii kuitenkin aikaa ja pitkäjännitteistä kehitystyötä. Palvelupohjaisten investointien kannattavuuden arvioimiseksi luvussa viisi mainituilla kannattavuusmittareilla hyödyt tulisi pystyä arvioimaan rahallisesti. Tämä voi olla hankalaa, koska osa hyödyistä realisoituu vasta pitkällä aikavälillä ja silloinkin vain jos arkkitehtuurin kehittäminen on ollut riittävän pitkäaikaista ja oikeansuuntaista. Taloudellisissa laskelmissa on myös otettava huomioon, että ne kohdistetaan oikealle taholle, koska osa hyödyistä koskee ohjelmiston tuottajaa ja osa asiakasorganisaatiota tai vasta sen asiakkaita. Investoinnin kannattavuuden laskemiseksi pitäisikin pystyä arvioimaan mitä hyötyjä tullaan saavuttamaan, missä vaiheessa ne saavutetaan ja mikä on niiden arvo.

2.6 Yhteenveto tavoitelluista hyödyistä

Palveluarkkitehtuuri ja web services yhdistelevät uudella tavalla järjestelmäintegraation sekä hajautettujen ja komponenttipohjaisten sovellusten keskeisiä elementtejä. Monet tavoitelluista hyödyistä perustuvat web services-tekniikoiden teknisiin ominaisuuksiin ja täyttyvät, kun teknisiä standardeja noudatetaan yhdenmukaisesti. Osa hyödyistä kuitenkin nojautuu tietyn tyyppisiin oletuksiin toimintatavoista ohjelmistojen tuottamisessa, arkkitehtuurin määrittelyssä ja tietojärjestelmien kehittämis- ja hankintapäätöksissä. Alle on eritelty "normalisoitu" joukko seikkoja, jotka näyttävät olevan SOA- ja web services-lähestymistavan etuina järjestelmäintegraatiossa ja -kehityksessä.

- *Tekniikoiden avoimuus* on selvä web services-teknikoilla saavutettava hyöty. Laaja välinetuki sekä soveltuvuus uusiin ja vanhoihin sovelluksiin ovat selviä teknisiä edistysaskelia web services-teknikoissa aikaisempiin väliohjelmistoihin verrattuna.
- *Uudelleenkäyttö* on palvelujen hyödyntäjän kannalta entistä helpompaa käytettäessä web services-teknikoita. Se tukee järjestelmien vähittäistä kehitystä, jo tehtyjen sovellusten ja infrastruktuurin hyödyntämistä, sekä päällekkäisyyksien vähentämistä tietojen syötössä, järjestelmien toteuttamisessa ja ylläpidossa. Uudelleenkäyttö tulisi kuitenkin olla suunniteltua, mikä lisää ratkaisujen suunnittelun ja toteutuksen työmäärää ja edellyttää uudelleenkäyttöä mahdollistavia toimintatapoja sekä hankinnoissa että järjestelmäkehityksessä.
- *Abstraktiotason nostamiseen* voidaan pyrkiä kahdella tasolla: jo perustekniikoiden hyödyntämisessä toteutuskohtaiset seikat on piilotettu rajapinnan taakse. Tällöin palvelun hyödyntäjän ei tarvitse välittää toteutuksen yksityiskohdista, mikä yksinkertaistaa integrointia ja sovellusten koostamista sekä järjestelmäkokonaisuuden joustavuutta. Abstraktiotasoa ja joustavuutta voidaan pyrkiä nostamaan edelleen siten, että myös arkkitehtuurilliset ja sisällölliset seikat on eristetty palvelumäärittelyistä. Tämä kuitenkin vaatii lisäeristyskerrosten rakentamista järjestelmiin, sopimisen siirtämistä "meta-tason" mekanismeihin ja ylimääräistä sovitustyötä olemassa oleviin sovelluksiin.
- *Julkaistu rajapinta* on erityisen hyödyllinen, jos se voidaan esittää riittävän tarkalla tasolla ohjelmallista hyödyntämistä varten. Hakemistoja tarvitaan nykyisin lähinnä "omassa, hyväksytyssä käytössä" olevien palvelujen hallintaan ja vaihtoehtoisten palveluosoitteiden selvittämiseen, ei niinkään mahdollisten kumppanien löytämiseen tai dynaamisesti erilaisiin rajapintoihin mukautuvien sovellusten toteuttamiseen.
- *Formaali sopimus*, jonka tärkeä osa on rajapintamäärittely, ja joka sitoo palvelun toteuttajaa ja käyttäjää, muodostaa erittäin käyttökelpoisen pohjan järjestelmien liittämiseksi. Osa sovittavista seikoista on kuitenkin rajapinnan ulkopuolella, eikä niiden määrittelyyn ole saatavilla vastaavia globaaleja standardeja kuin rajapintoihin ja kommunikaatioon.
- Sovelluspalvelujen ja ratkaisujen läheinen *käsitteellinen yhtäläisyys tuettavan toiminnan ja sen käsitteiden kannalta* on SOA-lähestymistapaan ja abstraktiotasoon liittyvä toiminnan tavoite, joka ei ole suinkaan itsestäänselvää teknisissä ratkaisuisissa. Erityyppisiin vaatimuksiin (ml. käytettävyys, vuorovaikutteisuus) vastaaminen voi olla vaikeaa, jos palvelujen vuorovaikutukset rakennetaan aina vastaamaan todellisten prosessien vuorovaikutustilanteita tai palvelemaan kaikkia mahdollisia tilanteita.

Etujen saavuttaminen ei kuitenkaan ole automaattista, vaan vaatii soveltamistapojen tarkentamista, ja kuten tarkasta läpikäynnistä näkyy, kuhunkin liittyy myös haasteita. Kuitenkin erityisesti seuraavia tarpeita voidaan tunnistaa tämän tarkastelun perusteella:

- mahdollisten erilaisten soveltamistapojen määrän pitäminen riittävän pienenä opittavuutta ja tehokkuushyötyjä varten,
- kuitenkin riittävä määrä malleja, joilla voidaan vastata erityyppisiin tarpeisiin, riittävän tarkasti määritellyt soveltamistavat näille malleille ja lisäominaisuuksia tarjoaville laajennuksille,
- palvelupohjaisuuden edellyttämien muutosten selvittäminen hankinta- ja ohjelmistotuotantoprosesseihin,

- rajapintamäärittelyn lisäksi palvelukuvauksiin tarvittavien muiden seikkojen tarkentaminen sekä kuvaus, mitkä niistä ovat yleiskäyttöisiä, mitkä paikallisia,
- sisällöllisten ja toimialakohtaisten standardien sovittaminen palvelupohjaiseen lähestymistapaan, jotta yhteentoimivuus voidaan määritellä tarvittaessa myös semanttisella tasolla ja hyödyntää jo tehty standardointityö,
- menetelmät ja mittarit eri tavoitteiden toteutumisen arviointiin käytännön hankkeissa.

Nämä soveltamistapojen vaatimukset on pyritty huomioimaan tämän oppaan eri osissa ja SerAPI-hankkeen muissa tuotoksissa.

3 Ohjelmistokehityksen osatekijät ja kehitysstrategiat palvelupohjaisessa arkkitehtuurissa

Tässä luvussa käsitellään kuinka perinteisiä ohjelmistosuunnittelun osatekijöitä (attribuutit) voidaan hyödyntää palvelupohjaisen arkkitehtuurin toteutuksessa ja suunnittelussa sekä mitä hyötyjä ja etuja eri kehitysstrategioilla on. Osatekijät on jaoteltu projektiperusteisiin tekijöihin ja ohjelmist ominaisuuksiin. Projektiperusteisia tekijöitä ovat pääomakustannukset ja kehitystyö. Ohjelmist ominaisuuksia ovat sisäisen rakenteen ominaisuudet ja ulkoiset laatuominaisuudet. Tässä luvussa ohjelmistottribuuteista käsitellään lähinnä sisäisen rakenteen ominaisuuksia, jotka saavat aikaan ulkoisia laatuominaisuuksia. Luku perustuu pääosin lähteeseen (Perepletchikov ym. 2005).

3.1 Projektiperusteiset tekijät

Palvelupohjaisessa arkkitehtuurissa ohjelmiston koko ei sovellu hyvin projektin kokonaiskustannusten arviointiin, koska SOA mahdollistaa uusien sovellusten kokoamisen käyttämällä olemassa olevia palveluja tai ostamalla palveluja ulkopuolisilta ohjelmiston tuottajilta. Kehitystyö vähenee koostettaessa ohjelmistoa olemassa olevista palveluista. Myös palveluja ostamalla kehitystyö vähenee, mutta toisaalta palvelujen hankintahinnat nostavat pääomakustannuksia. Kustannusten arviointiin liittyen (ks. myös luku 5) projektiperusteisten tekijöiden kannalta tarkastellaan erikseen pääomakustannuksia ja kehitystyön kustannuksia. Pääomakustannukset sisältävät projektin etukäteiskustannukset, joihin kuuluvat laitteisto, kehitystyökalut sekä koulutuskustannukset. Kehitystyökustannuksilla puolestaan kuvataan ohjelmiston elinkaaren aikana aiheutuvia kustannuksia.

3.2 Sisäisen rakenteen ohjelmisto-ominaisuudet

Sisäisen rakenteen ohjelmisto-ominaisuuksia ovat kytkentä (coupling), koheesio (cohesion) ja monimutkaisuus (complexity). Valittavalla kehitysstrategialla sekä siihen liittyvillä toiminnoilla ja tehtävillä on merkittävä vaikutus sisäisen rakenteen ominaisuuksiin. Rakenteiset ohjelmisto-ominaisuudet eivät suoraan kuvaa tuotteen näkyvää laatua, mutta ne vaikuttavat merkittävästi ohjelmiston ulkoisiin laatuominaisuuksiin. Palvelupohjaisten ratkaisujen laadun parantamiseen pyritään kehittämällä ohjeita, joilla vähennetään monimutkaisuutta ja kytkentää sekä kasvatetaan palvelujen koheesiota.

3.3 Ulkoiset laatuattribuutit

ISO/IEC 9126-1 standardi määrittelee kuusi pääasiallista ohjelmiston laatuominaisuutta, jotka ovat toiminnallisuus (functionality), luotettavuus (reliability), tehokkuus (efficiency), käytettävyys (usability), ylläpidettävyys (maintenability) ja siirrettävyys (portability) (ISO/IEC 2001). Sisäisen rakenteen ohjelmisto-ominaisuuksissa yhdistyy eri laatuattribuuttien tekijöitä. Siten ulkoisilla laatuominaisuuksilla on yhteys palvelupohjaisten järjestelmien ja palvelujen rakenteellisiin ominaisuuksiin. Laatuominaisuuksiin vaikuttaa myös muita tekijöitä, joiden vaikutus on otettava huomioon laadun arvioimisessa sisäisten attribuuttien lisäksi.

3.4 Palvelupohjaisen arkkitehtuurin kehitysstrategiat

SOA-arkkitehtuurissa käytettävät sovellusten kehitysstrategiat voidaan luokitella top-down-, bottom-up- sekä meet-in-the-middle-lähestymistapoihin. Top-down-menetelmässä lähdetään liikkeelle vaatimusten määrittelystä sekä toimintaprosessien mallintamisesta ja edetään vaiheittain kohti ohjelmiston toteutusta. Menetelmässä ratkaisun kohdealue jaotellaan toiminnallisiin osiin ja alijärjestelmiin. Toimintaprosessien mallintaminen tuottaa suunnitelman tarvittavien palvelujen määrittämiselle.

Top-down-menetelmällä saavutetaan yleensä parempi yhteensopivuus kuin bottom-up-menetelmällä. Tämä johtuu siitä, että top-down-menetelmässä ratkaisuihin eivät vaikuta yksittäisten ohjelmistojen sisäiset ratkaisut tai ohjelmointikielet, ja siitä että määrittelyt aloitetaan rajapintojen ja viestien määrittelyllä. Top-down-menetelmän haittapuolena on se, että se ei sovellu tilanteisiin, joissa palveluja pitäisi rakentaa olemassa oleviin perinnejärjestelmiin. Sen sijaan top-down-menetelmä soveltuu kehitettäessä ohjelmisto kokonaan alusta alkaen, käytettäessä olemassa olevia palveluja uutta tarkoitusta varten tai ostettaessa palveluja kolmannelta osapuolelta.

Bottom-up-menetelmä on vastakkainen lähestymistapa top-down-menetelmälle: sitä käytetään rakennettaessa palveluja olemassa oleviin perinnejärjestelmiin. Perinnejärjestelmiä saatetaan joutua modularisoimaan uudelleen siten, että ne tukevat palvelujen toiminnallisuutta. Bottom-up-menetelmässä voidaan käyttää kahta eri tapaa. Toinen tapa on lisätä palvelukerros perinnejärjestelmiin toteuttamalla kääreitä (wrappers) ja sovittimia (adapters). Vaihtoehtoinen tapa on muokata perinnejärjestelmän sisäistä rakennetta (refaktoroida) siten, että sitä pystytään käyttämään palvelurajapintojen kautta kuitenkin järjestelmän ulkoista käyttäytymistä muuttamatta.

Meet-in-the-middle-strategia on yhdistelmä top-down- ja bottom-up-menetelmiä. Käyttämällä top-down menetelmään saadaan organisaation toiminnan lähtökohdista tunnistettua korkean tason toimintaprosessin toiminnallisuus, jotka ilmaistaan karkeajakoisina palveluina. Hienojakoisten palvelujen määrittämisessä käytetään bottom-up-menetelmää, jossa tutkitaan olemassa olevan perinnejärjestelmän toiminnallisuutta. Tämän perusteella päätetään kuinka sovittimet ja kääreet tehdään tai kuinka perinnejärjestelmää komponentisoidaan, siten että se tukee haluttuja toimillisuuksia. Tämän jälkeen käyttämällä goal-service-mallinnusta vähennetään jo tunnistettujen palveluehdokkaiden määrää ja sidotaan hienojakoiset palvelut karkeajakoisiin palveluihin.

3.5 Kehitysstrategian vaikutus projektikustannuksiin ja ohjelmist ominaisuuksiin

Seuraavissa taulukoissa on esitelty eri kehitysstrategioihin liittyviä toimintoja ja tehtäviä, joilla on vaikutusta projektitekijöihin ja sisäisen rakenteen ohjelmisto-ominaisuuksiin. Taulukoissa on plus ja miinusmerkeillä ilmaistu tehtävien vaikutus attribuutteihin. Plusmerkki (+) ilmaisee positiivista vaikutusta ja miinusmerkki (-) ilmaisee negatiivista vaikutusta attribuuttiin. Monimutkaisuuden ja kytkennän vähentäminen ja koheesion kasvattaminen ovat tavoitteita. Sulkeisiin on merkitty myös jos attribuutilla on vaikutusta myös muihin attribuutteihin tai pääoman tuottoon (ROI). ROI (Return on Investment) on sijoitetun pääoman tuotto, jolla tarkoitetaan hankkeeseen investoitujen nettohyötyjen suhdetta aiheutuneisiin kustannuksiin (ks. luku 5.1.3).

Tarkastelemalla taulukoissa esiteltyjä tehtävien vaikutuksia ohjelmistoattribuutteihin voidaan ohjeistaa palvelupohjaisen arkkitehtuurin suunnittelua ja toteutusta. Seuraavat kappaleet ja niissä

esiintyvät taulukot käsittelevät eri kehitysstrategioiden vaikutuksia ohjelmistosuunnittelun attribuuteihin. Taulukoiden 1.1–1.3 yksittäiset solut sisältävät suosituksia kullakin lähestymistavalla noudatettavista palvelujen suunnittelun ja tuottamisen käytännöistä. Suosituksissa on otettava huomioon, että joillakin attribuuteilla saattaa olla vastakkaisia vaikutuksia. Esimerkiksi palveluiden yksityisen rekisterin ylläpito nostaa kustannuksia ja siten alentaa pääoman tuottoa. Kuitenkin olemassa olevien palveluiden käyttö yksityisestä rekisteristä alentaa kehitystyökustannuksia ja näin ollen pidemmällä aikavälillä yksityisen palvelurekisterin käytöstä saadaan hyötyä.

3.5.1 Ohjelmistoattribuutit top-down-menetelmässä

Top-down-strategiaa voidaan käyttää eri tavoin tuotettujen palvelujen yhteydessä. Top-down-menetelmää voidaan käyttää rakennettaessa palveluja alusta alkaen, käyttämällä olemassa olevia palveluja uutta tarkoitusta varten tai ostettaessa palveluja kolmannelta osapuolelta.

Yksi tärkeimmistä tehtävistä rakennettaessa palveluja alusta alkaen, on pienimmän rakeisuuden (fine-grained) palvelukomponenttien määrittäminen siten, että ne ovat uudelleenkäytettävissä. Toimintaprosessit ja karkeajakoiset (coarse-grained) palvelut koostuvat näistä hienojakoisista palvelukomponenteista. Löyhästi kytketyillä palveluilla ja korkealla uudelleenkäytettävyydellä parannetaan SOA-investointien tuottavuutta, alennetaan ylläpitokustannuksia sekä parannetaan palvelujen uudelleenkäyttömahdollisuuksia tulevaisissa projekteissa.

Palveluväylän (Enterprise Service Bus, ESB) ja integroidun palveluympäristön hankinnasta aiheutuu pääomakustannuksia, mutta toisaalta niiden käyttäminen vähentää kehitystyötä. Palvelurekisterin käyttö alentaa kehitystyötä, koska sitä käyttämällä palvelujen kuvaukset ovat saatavilla keskiteysti. Käytettäessä palveluja uuteen käyttötarkoitukseen säästetään myös testauksessa, koska palvelun yksikkötestausta ei tarvitse tehdä uudelleen. Palvelun yksikkötestaus tehdään sen tuottamisen yhteydessä, ja uudelleenkäytön yhteydessä riittää integrointitestaus. Päätös siitä, toteutetaanko palvelut itse vai hankitaanko ne ulkopuoliselta ohjelmistotoimittajalta, perustuu palvelujen saatavuuteen ja kustannus-hyöty-analyysiin.

Taulukko 3.1. Top-down-menetelmän vaikutukset attributteihin

Toiminnot	Projektiattribuutit		Rakenteiset ohjelmistoattribuutit		
	Pääomakustannus (PO)	Kehitystyön kustannus (KT)	Monimutkaisuus (MM)	Kytkeä (KYT)	Koheesio (KOH)
Valmistettavat palvelut	<ul style="list-style-type: none"> - Olemassaoleva kehitystiimi – - Koulutuksen hankkiminen + (KT–) - Standardoidun väliohjelmiston ja kehitystyökalun hankinta + (KT–) - Standardoidun dokumentoinnin/viitemallin muodostaminen + (KT–) - Palveluiden yksityisen rekisterin ylläpito + (ROI–) 	<ul style="list-style-type: none"> - Iteratiivisesti rakentaminen – - Kypsiin ohjelmistokehitysprosessien käyttö – (PO+) - Kehitystiimien ryhmitys loogisten liiketoimintatehtävien mukaan – - Uudelleen käytettävyyden rakentaminen + (ROI+) (KYT–) 	<ul style="list-style-type: none"> - Mallipohjaisen arkkitehtuurin soveltaminen hienojakoisten palvelukomponenttien hajottamisessa toimintaprosesseihin – (KT–) - Palvelukomponenttien toteutus oliopohjaisesti – - Monimutkaisten komponenttien hajottaminen – - Globaalin datan kapselointi tietyn palvelun käyttöön – 	<ul style="list-style-type: none"> - Pienimpien uudelleen käytettävien ohjelmayksiköiden määrittäminen – - Palvelukomponenttien ja palveluiden liittäminen rajapintojen kautta – - Yksinkertaisten rajapintojen määrittely – - Erialaisten työkulkujen vaikutuksen eristäminen palvelujen toteutuksista – 	<ul style="list-style-type: none"> - Hienojakoisten palvelukomponenttien rakentaminen + - Palvelukomponenttien koostaminen palveluiksi vain jos palvelu kuvaa todellista (liike)toimintaa + - Sovelluksen toimintatapojen kuten turvallisuuden, palvelutasojen (SLA) ja palvelun laadun (QoS) vaikutusten eristäminen itse palvelusta +
Palveluiden uudelleen käyttö	<ul style="list-style-type: none"> - Toimintojen mallintamis- asiantunte muksen hankinta olemassa olevien uudelleen käytettävien palveluiden tunnistamista varten + (KT–) - Olemassa olevien väliohjelmistojen ja kehitystyökalujen hyödyntäminen – 	<ul style="list-style-type: none"> - Olemassa olevien palveluiden uudelleen käyttö sisäisestä rekisteristä – - Koostamiseen tarvittavan koodin sijoittaminen prosessitasolle, eikä palveluihin – 	Ei sovellettavissa (palvelujen sisäinen rakenne pysyy muuttumattomana)	Ei sovellettavissa (palvelujen sisäinen rakenne pysyy muuttumattomana)	Ei sovellettavissa (palvelujen sisäinen rakenne pysyy muuttumattomana)
Palvelujen hankinta ostamalla	<ul style="list-style-type: none"> - Palveluiden hankinta tunnetuilta myyjiltä + (KT–) - Riittävän analyysin teko palvelujen valmistamisen/hankinnan kustannuksista ja hyödyistä + (KT–) 	<ul style="list-style-type: none"> - Vain määriteltyjen palveluiden toteuttaminen itse ja loput ostamalla – - Hienojakoisten palvelujen hankinta ostamalla, mutta karkeajakoisten rakentaminen itse – - Käyttö uusiin tarkoituksiin, mikäli mahdollista – 	Ei sovellettavissa (ei voi vaikuttaa ostettaessa palveluja)	Ei sovellettavissa (ei voi vaikuttaa ostettaessa palveluja)	Ei sovellettavissa (ei voi vaikuttaa ostettaessa palveluja)

3.5.2 Ohjelmistoattribuutit bottom-up-menetelmässä

Bottom-up-menetelmässä on arvioitava perinnejärjestelmien rakenteellisia ominaisuuksia ja niiden vaikutusta laatuominaisuuksiin. Näiden perusteella voidaan tehdä päätökset siitä, onko järjestelmää parempi muuttaa sisäisesti tai rakenteellisesti vai voidaanko järjestelmään vain yksinkertaisesti lisä-

tä palvelukerros. Erittäin monimutkaisiin järjestelmiin on vaikeaa liittää palvelukerros. Toimintaprosessimalleja käytetään hyödyksi, kun päätetään mitä palveluja toteutetaan.

Olemassa olevan järjestelmän sisäisistä tai rakenteellisista muutoksista (refaktorointi) olisi hyvä tehdä ensin rajattu kokeilu, ja aloittaa vahvasti kytketyistä ja monimutkaisista ohjelmamoduuleista. Tällöin voidaan arvioida kustannuksia ja tuottoa ennen suurempien päätösten tekoa ja samalla saada kokemusta ennen siirtymistä isompiin ohjelmistomoduuleihin. Refaktorointityön tekemisessä tulisi käyttää hyväksi työntekijöitä, jotka ovat olleet mukana perinnejärjestelmän arkkitehtuuri- ja suunnittelutehtävissä. Mikäli perinnejärjestelmä on tuttu suunnittelijoille ja kehittäjille, kehityskustannuksissa saavutetaan säästöjä.

Taulukko 3.2 Bottom-up-menetelmän vaikutukset attribuutteihin

Toiminnot	Projektiattribuutit		Rakenteiset ohjelmistoattribuutit		
	Pääomakustannus (PO)	Kehitystyö (KT)	Monimutkaisuus (MM)	Kytkeä (KYT)	Koheesio (KOH)
Perinnejärjestelmien refaktorointi	<ul style="list-style-type: none"> - Perinnejärjestelmien arkkitehtuurin ja toteutuksen asiantuntijoiden käyttö + (KT-) - Yleiskäyttöisten (utility) palvelujen hankinta + (KT-) - Olemassa olevien resurssien (esim. tietokannat) maksimaalinen käyttö - 	<ul style="list-style-type: none"> - Iteratiivinen refaktorointi - - Keskittyminen tiukasti kytkettyihin ja monimutkaisiin järjestelmän osiin - (MM-) (KYT-) (KOH+) - Palvelusovittimien hankinta moduleille, jotka ovat löyhästi sidottuja ja joilla on korkea koheesio (ei tarvita refaktorointia) - (PO+) 	<ul style="list-style-type: none"> - Monimutkaisuuden jakaminen pienempiin, refaktoroituihin palvelukomponentteihin - 	<ul style="list-style-type: none"> - Kytken väheneminen varmistamalla, että eri tavoilla toteutetut moduulit kommunikoivat vain rajapintojen kautta - 	<ul style="list-style-type: none"> - Olemassa olevien moduulien refaktorointi hienojakoisiin palvelukomponentteihin +
Palvelukerroksen lisääminen perinnejärjestelmiin	<ul style="list-style-type: none"> - Perinnejärjestelmien arkkitehtuurin ja toteutuksen asiantuntijoiden käyttö + (KT-) - Valmiiden sovittimien hankinta + (KT-) - Olemassa olevien resurssien maksimaalinen käyttö - 	<ul style="list-style-type: none"> - Karkeajakoisten palvelujen rakentaminen - (MM+) (KYT+) (KOH-) - ESB-väylän käyttöönotto ja palveluiden vähittäinen lisääminen siihen - - Riippuvuuksien poistaminen järjestelmien väliltä, joilla on yhteistä infrastruktuuria + (MM-) (KYT-) 	<ul style="list-style-type: none"> - Perinnejärjestelmien ja moduulien tulisi olla yhteydessä ainoastaan palvelukerroksen läpi, ei suoraan tai yhteisen infrastruktuurin kautta - 	<ul style="list-style-type: none"> - Toimintoja eri perinnejärjestelmistä ei yhdistetä samaan palveluun - 	<ul style="list-style-type: none"> - Hienojakoisten palveluiden kehittäminen +

Lisättäessä palvelukerroksia perinnejärjestelmiin on joskus mahdollista hankkia ohjelmistopalvelujen kääreitä ja sovittimia ostamalla niitä kolmannelta osapuolelta, jolloin säästetään kehityskustannuksissa. Olemassa olevia resursseja tulisi hyödyntää palveluissa niin paljon kuin mahdollista. Palveluja kehitettäessä pyritään kehittämään hienojakoisia palveluja, jolloin palveluiden koheesio kasvaa sekä monimutkaisuus ja kytkentä vähenevät.

3.5.3 Ohjelmistoattribuutit Meet-in-the-middle-menetelmässä

Meet-in-the-middle-menetelmä on yhdistelmä top-down- ja bottom-up-menetelmiä. Bottom-up-menetelmässä toimintaa ei ohjaa toimintaprosessien näkökulma, vaan palvelujen suunnittelu perustuu olemassa olevaan IT-ympäristöön. Top-down-menetelmän ongelma taas on se että, sillä ei välttämättä pystytä määrittelemään ei-toiminnallisia vaatimuksia, ja menetelmällä syntyy helposti yhteensopimattomuutta karkeajakoisten palvelujen ja hienojakoisten komponenttien välille. Meet-in-the-middle-menetelmällä saavutetaankin usein parhaiten halutut toiminnot, ja voidaan ottaa huomioon sekä prosessien mallinnus että olemassa olevat järjestelmät. Meet-in-the-middle-menetelmän haittapuolena on kalleus, mutta sen avulla voitaneen aikaansaada parempaa tuottoa sijoitetulle pääomalle.

Taulukko 1.3 Meet-in-the-middle-menetelmän vaikutukset attribuutteihin

Toiminnot	Projektiattribuutit		Rakenteiset ohjelmistoattribuutit		
	Pääomakustannus (PO)	Kehitystyö (KT)	Monimutkaisuus (MM)	Kytkeä(KYT)	Koheesio (KOH)
Palvelukerroksen lisääminen perinnejärjestelmiin	<ul style="list-style-type: none"> - Perinnejärjestelmien arkkitehtuurin ja toteutuksen asiantuntijoiden käyttö + (KT-) - Standardoitujen dokumentointimallien tuottaminen + (KT-) - Olemassa olevien resurssien maksimaalinen käyttö - 	<ul style="list-style-type: none"> - Palvelujen kehittäminen etsimällä niitä perinnejärjestelmistä – - Mahdollisten palvelujen vähentäminen soveltamalla goal-service-modelling menetelmää – 	<ul style="list-style-type: none"> - Yhdistelmä top-down ja bottom-up menetelmiä 	<ul style="list-style-type: none"> - Yhdistelmä top-down ja bottom-up menetelmiä 	<ul style="list-style-type: none"> - Yhdistelmä top-down ja bottom-up menetelmiä

3.5.4 Ristiriittekijät

On useita tekijöitä, jotka vaikuttavat toisiin ominaisuuksiin ja tekijöihin positiivisesti ja toisiin negatiivisesti. Tällaisia ristiriittekijöitä on esimerkiksi projektin kustannusten ja ohjelmiston laadun välillä. Suurimmat ristiriitoja aiheuttavat tekijät ovat palvelujen rakentaminen uudelleenkäyttöä varten sekä palvelujen rakeisuuden (granularity) vaikutukset eri ominaisuuksiin.

Top-down-menetelmällä uudelleenkäytettäviksi kehitetyt palvelut aiheuttavat ristiriitatilanteita, koska ne lisäävät kehitystyön määrää, mutta toisaalta ne kasvattavat myös pääoman tuottoa ja parantavat palvelujen toteutustason kytkentää. Palvelujen rakentaminen uudelleenkäytettäviksi voi vaatia huomattavasti enemmän kehitystyötä verrattuna siihen, että palvelu rakennettaisiin vain tiettyä tarkoitusta varten. Toisaalta palvelun hyvä uudelleenkäytettävyys voi alentaa merkittävästi tulevien projektien kehityskustannuksia ja siten parantaa pääoman tuottoa. Uudelleenkäytettävien palveluiden kytkentä on hyvin alhainen, koska ne on toteutettu kuten itsenäiset ohjelmistoyksiköt.

Palvelujen rakeisuudella on vaikutusta useisiin ohjelmistoattribuutteihin. Karkeajakoisten palvelujen tuottaminen perinnejärjestelmiin vähentää kehitystyötä, koska kehittäjien on helpompaa yleistää olemassa olevaa toiminnallisuutta karkeajakoisiin palvelurajapintoihin. Karkeajakoiset palvelut usein myös vähentävät kommunikaation tarvetta ja siten parantavat verkon suorituskykyä. Toisaalta

karkeajakoisuus lisää kytkeä (coupling) sekä alentaa koheesiota (cohesion), mikä heikentää ylläpidettävyyttä, luotettavuutta ja tehokkuutta.

3.6 Yhteenveto ohjelmistokehityksen osatekijöistä ja kehitysstrategioista palvelupohjaisessa arkkitehtuurissa

Projektiperustaisia ohjelmisto-ominaisuuksia (pääomakustannuksia ja kehitystyön kustannuksia) sekä sisäisen rakenteen ohjelmisto-ominaisuuksia (kytkentä, koheesio ja monimutkaisuus) tarkastelemalla on tuotettu suosituksia palvelupohjaisen arkkitehtuurin suunnittelulle ja toteutukselle eri kehitysstrategioita käyttäen. Ohjelmistojen kehitysstrategiat vaikuttavat oleellisesti ohjelmistokehityksen kustannusrakenteeseen ja ohjelmiston laatuominaisuuksiin. Ohjelmistosuunnittelun eri tekijät kuvaavatkin hyvin kehitysstrategioiden vaikutuksia palvelupohjaisen arkkitehtuurin kehityskustannuksiin ja ohjelmistojen sisäisen rakenteeseen.

Tässä esitetyillä kehitysstrategioilla on suuri vaikutus palvelupohjaisen arkkitehtuurin suunnitteluun, palvelujen ja prosessien toteutuksiin ja hankintoihin. Oppaan osassa 2 kuvataan monia seikkoja esimerkiksi palvelujen tunnistamiseen ja määrittelyyn liittyen top-down ja bottom-up-näkökulmista sekä esitellään eri kehitysstrategioiden tukena käyttökelpoisia viitemalleja ja kehitysprosesseja.

4 SOA ja Web services-hyötyjen arviointi ja mittaus

Tämä luku käsittelee palveluarkkitehtuurin ja web-sovelluspalvelujen arviointia tietojärjestelmäprojekteissa. Luvussa esitellään tietojärjestelmien arviointiin kehitettyjä malleja ja sovelletaan niitä esimerkin omaisesti suhteessa Service Oriented Architecture (SOA)-lähestymistavalla ja web services-tekniikoilla tavoiteltuihin hyötyihin. Tuloksena tästä on joukko arvioinnin kohteita, mittareita ja toimintaohjeita arviointiin, joita voidaan käyttää projekteissa kehitettäessä ja arvioitaessa terveydenhuollon tietojärjestelmiä ja integrointiratkaisuja. Arviointimallien tavoitteena on tukea tuotteiden, sovellusten ja niiden osien (esim. sovelluspalvelut) arviointia suhteessa palveluarkkitehtuurilla ja web-sovelluspalveluilla (menetelminä, lähestymistapoina ja yleiskäyttöisinä teknisinä ratkaisuinä) tavoiteltuihin hyötyihin. Tällaisen arvioinnin tavoitteena on pyrkiä tuottamaan todistettavuutta asetettujen tavoitteiden saavuttamisesta, saavuttamattomuudesta tai etenemisestä haluttuun suuntaan. Tämän tyyppinen arviointi auttaa etenkin erottamaan, kuinka strategisissa linjauksissa tai "markkinointipuheissa" asetetut tavoitteet toteutuvat käytännössä. Mittareiden tunnistaminen myös auttaa määrittelemään tavoitteet mitattavalla tavalla. Niiden avulla voidaan pyrkiä perustelemaan esim. SOA-lähestymistavan kehittämisellä saavutettavia hyötyjä.

Luvussa 4.1 esitellään arvioinnin suosituksia ja tietojärjestelmien arviointimalleja, joita on mahdollista soveltaa myös palvelupohjaisten järjestelmien arvioinnissa. Luvussa 4.2 tarkastellaan palvelupohjaisten ratkaisujen laadun muodostumista. Edellä mainitut luvut sisältävät lähinnä taustaa-aineistoa ja kirjallisuudessa esitettyjä malleja. Lukuun 4.3 on koostettu palvelupohjaisten ratkaisujen arvioinnissa käytettäviä mittareita ja esimerkki mittareiden valinnasta yksilöidyn tavoitteen arviointiin. Luvussa 4.4 esitetään arviointiprosessin vaiheet palvelupohjaisten ratkaisujen arviointiin. Lukuun 4.3 ja 4.4 sisältö on tarkoitettu suoraan hyödynnettäväksi palvelupohjaisten ratkaisujen arviointia suoritettaessa.

4.1 Tietojärjestelmien arviointimallit

Arviointitutkimuksessa korostetaan arvioinnin eri puolia ja tuloksia. Arviointiin on valittava näkökulma, joka ilmaisee ne "silmälasit", jolla voidaan nähdä joitakin asioita paremmin kuin muutoin olisi mahdollista. Näkökulma myös sisältää oletuksen siitä, mikä on perusteltua tietoa (epistemologia), arvioitavat näkökannat, jotka on valittu ja menetelmät, kuinka tietoa muodostetaan (metodologia). Nämä oletukset vaikuttavat arvioinnin kohteena olevan kysymyksen/seikan käsitteellistämiseen ja siihen, kuinka arviointi suunnitellaan, aineisto kerätään ja millaisia analyysimenetelmiä käytetään. Esimerkiksi taloudellinen näkökulma tarkentaa kuinka paljon resursseja käytetään arvioinnissa ja mitä hyötyä saadaan. Seuraavassa on kuvattu arviointitutkimusta pääasiassa käyttäen Øvretveitin lähetymistapaa (Øvretveit 2003).

Arviointitutkimuksissa voidaan erottaa esimerkiksi seuraavat neljä eri näkökulmaa - kokeileva, taloudellinen, kehittävä ja johtamisen näkökulma. Jokaisella näkökulmalla voidaan tehdä arviointitutkimusta esimerkiksi hoidosta, palveluista, menettelytavoista ja organisaation interventioista (esimerkiksi tietojärjestelmän käyttöönoton vaikutuksista palveluprosessiin). Valittava näkökulma riippuu arvioinnin tarkoituksesta, arvioinnin luonteesta ja sen ennakoitavista vaikutuksista, mutta myös arvioijan harjaantuneisuudesta ja itsekurista.

Kokeilevan arvioinnin tavoitteena on löytää intervention (esim. tietojärjestelmän käyttöönotto) vaikutukset ja niiden syyt. Arvioinnilla testataan oletuksia vaikutuksista ja se toteutetaan tieteellisen tutkimuksen tapaan, pyrkien objektiiviseen mittaamiseen ja hallittuun toteutukseen. Taloudellisen

arvioinnin tavoitteena on löytää kuinka paljon resursseja käytetään intervention toteuttamiseen ja määrittellä sen seuraukset tai tulokset. Taloudellisessa arvioinnissa on samanlaisia lähtökohtia ja oletuksia kuin kokeellisessa arvioinnissa, ja ne toteutetaan usein samantyyppisesti. Kehittävä arviointi poikkeaa edellisistä etenkin oletuksineen intervention luonteesta ja arvioinnin tarkoituksesta. Kehittävä arviointi käyttää systemaattisia metodeja ja teorioita mahdollistaakseen palveluiden tai menettelytapojen kehittämisen ja parantamisen. Johtamisen arviointi on kehitetty johtajille ja hallinnolle palveluiden tai menettelytapojen seuraamiseksi tai parantamiseksi, tai tarkistamaan sovittujen muutosten tai projektien aiottu toteutuminen. Johtamisnäkökulmasta pyritään varmistamaan sovittujen vastuiden mukainen toiminta, vastine rahalle ja suorituskyvyn paraneminen

Erilaisten organisaation kehittämis- ja muutosprosessien kulkua voidaan kuvata seuraavasti: Ensin ovat tarpeet ja vaatimukset, joiden perusteella mietitään panokset ja prosessit (joita molempia arvioidaan ennen toteuttamista). Näistä puolestaan syntyvät prosessien tuotokset ja tulokset (outputs / outcomes). Panosten kohdalla arviointityyppi voi olla soveltuvuuden arviointia, ennustavaa arviointia, simulaatioarviointia, ennen-jälkeen arviointia, tai jopa arvioitavuuden arviointia. Prosessien kohdalla arviointi on prosessiarviointia, formatiivista arviointia tai vaihtoehtojen arviointia. Tuotos-ten kohdalla voidaan käyttää tulosten arviointia, vaikutusten arviointia tai summatiivista arviointia (ks. myös kuva 3).

Terveydenhuollon tietojärjestelmien arvioinnissa on ehdotettu käytettäväksi elinkaariajattelua. Tietojärjestelmien ja ohjelmistojen elinkaareissa voidaan löytää neljä vaihetta, joissa arviointia tehdään: 1) järjestelmän spesifikaatiot, 2) toteutetun järjestelmän validius ja oikeellisuus, 3) toimivuus ja käytettävyys käyttäjäympäristössä sekä 4) järjestelmän vaikutukset käyttäjään, organisaatioon, terveydenhuoltopalveluiden jakeluun, hoidon laatuun jne (ks. myös kuva 8). Kaksi ensimmäistä vaihetta ovat suosittuja asiantuntijajärjestelmien ja päätöksenteon tukijärjestelmien tutkimuksessa, kun taas kahta jälkimmäistä tulisi painottaa laajemmassa terveydenhuollon tietojärjestelmien arvioinnissa. Arvioinnissa on huomioitava myös ajanäkökulma, joka liittyy sopivien arviointistrategioiden ja metodien valintaan järjestelmän eri kehitysvaiheissa. Elinkaariajattelu on samantyylinen kuin DeLonen ja McLeanin tietojärjestelmien onnistumisen malli (ks. luku 4.1.3). (Turunen 2001)

4.1.1 Eri arviointitapoja

Chen (1990) mukaan normatiivinen arviointi (normien luonti arvioinnin avulla) antaa tietoa, joka auttaa tarkemmin selkeyttämään ja tunnistamaan kehittämisen kohteen rakennetta ja ominaisuuksia sekä teoriassa että käytännössä. Normatiivinen tulosten arviointi antaa sidosryhmille ajantasaista tietoa miten selventää ongelmia kohteen tavoitteiden osalta tai miten yhdistää tavoitteet ja varsinainen toiminta. Normatiivista tulosten arviointia voidaan käyttää myös terveydenhuollossa, sekä suunnittelun alkuvaiheessa että seuranta- ja toteutusvaiheissa. Kehittämis- ja muutosprosesseissa päämäärät kehittyvät tai muuttuvat esimerkiksi poliittisen tai organisaation ilmapiirin, menettelytapojen, henkilöstön, toteutettavan intervention rakenteen, asiakkaiden jne. muutosten vuoksi. Arviointi auttaa osapuolia selkeyttämään tulosten luonnetta ja tavoitteita suunnitelmien ja johtamisen tavoitteiden osalta. Tulokset koostuvat sekä odotetuista tuloksista että odottamattomista tuloksista, joita molempia arvioinnissa käsitellään. Tavoitteet ja tulokset ovat sidoksissa suunnitelmiin ja johtamisprosesseihin, ja normatiivinen arviointi tarjoavaa välineitä päätöksentekijöille, johtajille ja hallintoviranomaisille. Normatiivista arviointia käytetään mm. kehitystyön tai muutoksen olemassaolon oikeuttamiseen, poliittisten ryhmittymien sitomiseen, resurssien hankintaan, talousarvion kohdentamiseen, suorituksen ja tehokkuuden vertailuun sekä toiminnassa tarvittavan päätöksenteon perustaksi. Normatiivisessa tulosten arvioinnissa on kolme keskeistä kohtaa - tavoitteiden määrittely, tavoitteiden yksimielisyyden ja tavoitteiden toteuttamiskelpoisuuden arviointi. Erityisesti ta-

voitteiden määrittelyn arviointi ohjaa tunnistamaan ja kuvaamaan eri osapuolien käsitykset ja näkemykset pääasiallisten tavoitteiden suunnasta.

Samantapainen arvioinnin lähestymistapa kuin normatiivinen arviointi on myös Øvretveitin mainitsema arvioitavuuden arviointi, jossa arvioidaan suunniteltua interventiota (toiminnan muutosta, tietojärjestelmän käyt-töönottoa tms.) ja jossa päätetään arvioinnin toteutustavasta - tehdäänkö täysimittainen arviointi vai ei, ja millä tavoin. Tällainen arviointi selkeyttää intervention sisällön ja sen rajat, tärkeimmät arvioinnin kysymykset ja arvioitavien kohtien tärkeimmät osatekijät. Se antaa mahdollisuuden rajata tavoitteet ja sopia, mitä halutaan arvioida. Arvioitavuuden arviointi kattaa erilaiset arviointitavat, arvioinnin kohteet sekä erilaisten arviointisuunnitelmien kulut ja hyödyt.

Ennen-jälkeen, ex-ante(ennakoiva) ja soveltuvuuden arviointeja voidaan käyttää ennen aiotun suunnitelman toteuttamista. Tällöin usein halutaan arvioida mahdollisia seurauksia ja kustannuksia. Näitä tutkimuksia (arvioita) on joskus toteutettu ennen ehdotettua interventiota. Ne eroavat arvioitavuuden arviosta siten, että ne eivät tarkastele mahdollisia tapoja tehdä arviointia, vaan mahdollisia seurauksia. Ennen-jälkeen arvioinnissa arvioidaan intervention vaikutuksia vertaamalla tuloksia ennen ja jälkeen kehittämis- tai muutosprosessia ja toteuttamalla molemmat tutkimukset samalla tutkimusmenetelmällä.

Vaihtoehtojen arvioinnissa pyritään helpottamaan valintaa eri vaihtoehtojen välillä. Arviointi- ja analysointimenetelmät ovat usein simulointitekniikalla toteutettuja tapoja (=yleensä todellisten tapahtumien matemaattista tai loogista mallintamista). Niissä käytetään esimerkiksi ryhmän päätösten analysointia selventämään käytettyjä kriteereitä päätöksentekijöille ja vertaamaan systemaattisesti vaihtoehtoja näillä kriteereillä. Tätä lähestymistapaa on käytetty terveydenhuollossa ennen tietojärjestelmän hankintaa ja kehitettäessä laadun arviointia sekä arviomaan palvelun suoritusta. Tämä on käyttökelpoinen tapa erityisesti, kun ei ole aikaa kerätä aineistoa esim. uuden tietojärjestelmän vaikutuksista tai hyödyistä. Lähestymistapa myös auttaa päätöksentekijöitä, kun heillä on tarve selkiyttää subjektiiviset arvionsa päätettäessä toiminnan vaihtoehtoisista suunnista sekä kun on tarpeellista yhdistää monia näkökohtia.

Formatiivinen arviointi on eräs kehittävän arvioinnin tyyppi. Sen tarkoituksena on antaa tietoa ja tukea henkilöille, jotka tekevät parannuksia esim. kehitystyön kohteena olevaan prosessiin tai järjestelmään. Yleensä he ovat palvelun tarjoajia ja arvioijat työskentelevät läheisesti heidän kanssaan auttaen heitä tavoitteiden tarkentamisessa ja selkiyttämisessä ja antavat heille säännöllistä palautetta ja informaatiota. Informaatio voi koskea tuloksia, jotka on usein kuvattu prosessin näkökulmasta, tai palvelun toimintoja koskevaa tietoa tai eroavuuksia tavoitellun ja nykykäytännön välillä.

Prosessiarviointi keskittyy prosessin arviointiin. Sitä on toteutettu mm. palveluiden ja kehitystyön arvioinnissa sekä arvioitaessa menettelytapojen toteutumista. Prosessiarviointi tarkastelee vain prosessia, kun taas formatiivinen arviointi kokoaa tietoja myös tuloksista. Prosessiarvioinnin tavoitteena on antaa tietoa siitä, kuinka palvelu toimii ja miten palvelu tuottaa tuloksensa (ei siitä, mitä se tuottaa). Erityisen tärkeää prosessiarvioinnissa on oletettujen vahvuuksien ja heikkouksien käsittely sekä riittävän tarkan käsityksen muodostamisessa itse prosessista.

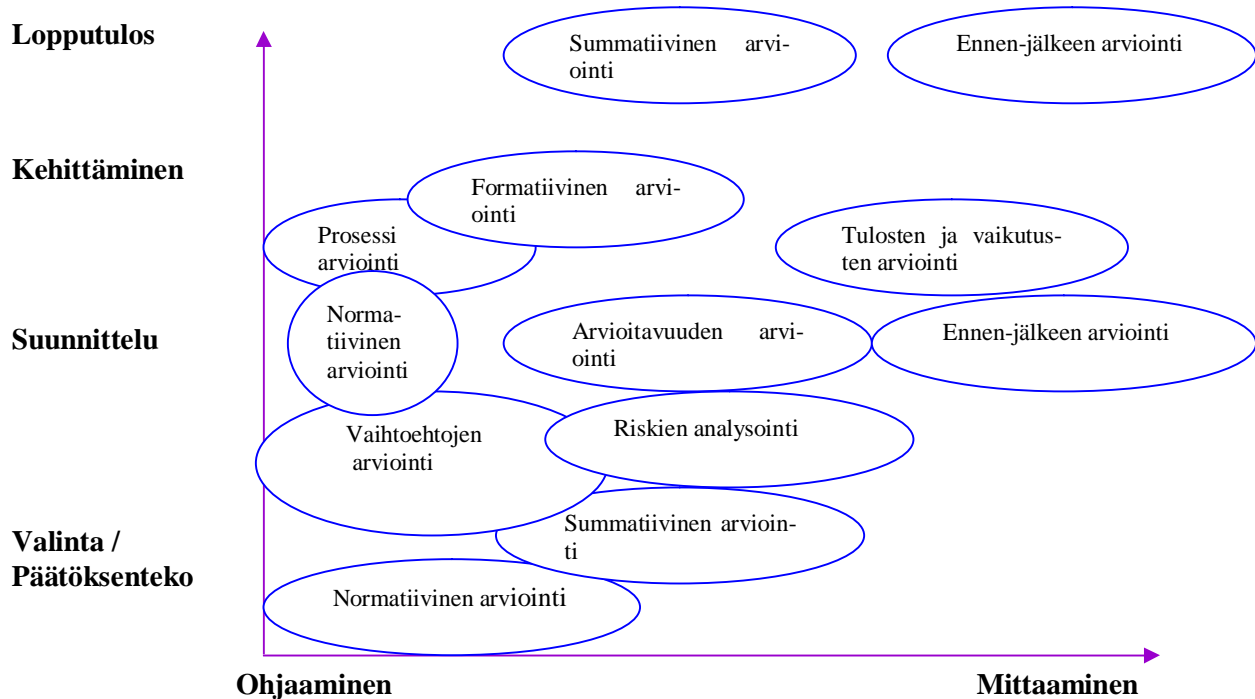
Summatiivisen arvioinnin tarkoituksena on tuoda apua päätöksentekijöille päätettäessä palvelun tai toimintamallin jatkamisesta. Tavoitteena on antaa heille arvio intervention vaikutuksista ja tehokkuudesta - tehdä niistä yhteenveto. Summatiivisen arvioinnin keskipisteessä ovat tuotokset ja tulokset sekä henkilökunnan resursointi. Yleensä summatiivista arviointia tekevät ulkopuoliset tahot, eikä juurikaan palvelun tarjoajat.

Tulosten tai vaikutusten arviointi termejä käytetään mistä tahansa arvioinnista, joka kohdistuu tulosten löytämiseen tai laajemmin vaikutuksiin, palveluun tai toimintamalliin. Tulosten arviointi voi olla osa summatiivista tai formatiivista arviointia, mutta se ei ole yleensä mukana prosessiarvioinnissa

Riskianalyysissä käsitellään todennäköisyyksiä sille, että jokin haitallinen asia tapahtuu (Juutilainen & Tuomisto 2002). Riskianalyysissä kootaan tietoa riskistä, tiedotetaan niistä, vertaillaan riskejä ja tehdään päätöksiä. Riskinarvioinnin tarkoituksena on tuottaa tietoa riskin todellisesta suuruudesta päätöksentekoa varten. Riskianalyysiä käytetään tuntemattomien riskien tunnistamiseen ja niiden vähentämiseen; esimerkiksi, kun halutaan varmistua uusien tuotteiden tai tekniikan turvallisuudesta. Koska tutkittavia riskejä on monenlaisia, riskinarviointiprosessit voivat olla hyvin erilaisia. Riskianalyysissä löydetään myös toiminnallisia huonoja puolia tai haavoittuvuuksia. Riskinarvioinnilla pyritään materiaalin kokoamiseen päätöksentekoa varten ja mahdollisesti päättäjille annettaviin suosituksiin päätöksen suunnasta.

Bernard ym. ja Lund ym. kuvaavat riskienhallinnan tarkoituksiksi tunnistaa ja dokumentoida mahdolliset ongelmat ennen kuin ne tulevat esille siten, että riskien hallintatoimet voidaan suunnitella. Hallintatoimet toteutetaan tarvittaessa tuotteen tai projektin elinkaaren aikana epäsuotuisten vaikutusten vähentämiseksi. Riskinhallinta on päätöksentekoa ja toimintaa, jolla yritetään vähentää riskinarvioinnissa esiin tullutta riskiä.

Erilaisia strategioita riskienhallinnan prosessissa ovat mm.: 1) kontekstimalli, 2) SWOT analyysi malli, 3) ei-haluttujen tapahtumien malli, 4) uhkatekijöiden malli, 5) riskimalli ja 6) hoitokeinojen malli. Turvallisuuden arviointi alkaa aina arvioinnin kontekstin tunnistamisella. (Bernard ym. 2005; Lund ym. 2004)



Kuva 3. Eri arviointitapojen vertailua

Erilaisia arviointitapoja ja lähestymistapoja on siis runsaasti. Olennaista arvioinnin kannalta on määrittellä arvioinnissa suoritettavan mittaamisen suhde ohjaamiseen, määrittellä se, kohdistuuko arviointi valintatilanteeseen, suunnitteluun, kehittämiseen vai lopputulokseen, sekä selkeä arvioinnin (ja arvioitavan kohteen) tavoitteiden määrittely.

4.1.2 VATAM-suositukset tietojärjestelmien arviointiin

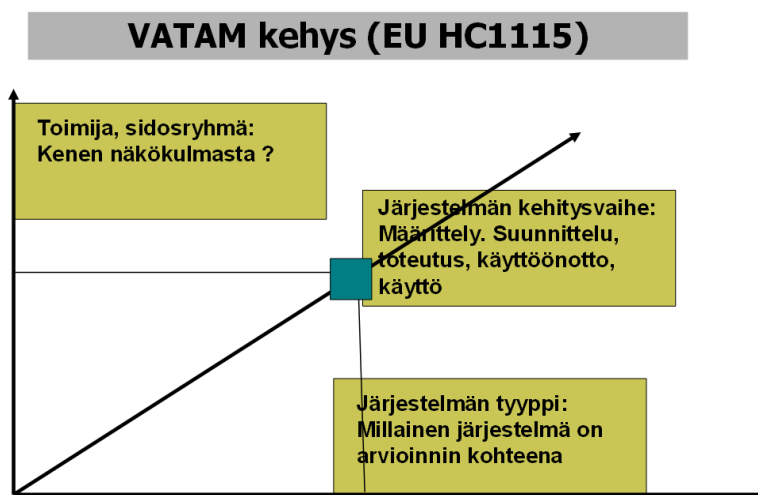
Tietojärjestelmien onnistumiseen vaikuttavat sovellusten ja ratkaisujen tekniset ominaisuudet, toimintaympäristön ja käyttäjäorganisaatioiden ominaisuudet sekä inhimilliset ja sosiaaliset tekijät (kuva 4). Järjestelmien arvioinnissa pyritään ottamaan näitä tekijöitä huomioon.



Kuva 4. Tietojärjestelmien onnistumisen osatekijät (Nykänen 2005)

Talmon ym. kuvaavat artikkelissaan (Talmon Jan, ym. 1999) VATAM -suositusta, jossa arviointiin yleensä voidaan sisällyttää erityyppisiä asioita. VATAM -suositus (Validation of Tele-matic Applications in Medicine project) sisällyttää summatiiviseen arviointiin myös järjestelmän vaikutukset, terveydenhuollon seuraukset ja taloudelliset näkökohdat. VATAM-suositukset on tarkoitettu esimerkiksi investoinneista päättävälle, asiantuntijoille, järjestelmän käyttäjille, johtajille, prosessin ylläpitäjille jne. Pääasiallinen tarkoitus on luoda "sopiva ajattelutapa" terveydenhuollon tietotekniikan arviointia varten

Pääpaino VATAM-suosituksissa on toimittajien ja palveluntarjoajan näkökulmassa, mutta myös hallinto on otettava huomioon. Toimittajat haluavat selvittää onko tuotteelle tai palvelulle kysyntää ja potentiaalisia asiakkaita, joiden ongelmia voidaan ratkaista. Ostajille on olennaista, kohtaako olemassa oleva tuote heidän tarpeensa ja mitä he voivat odottaa teknologian käytöltä, ja ottaako se huomioon riittävästi käytön vaikutuksia (taloudelliset, henkilökohtaiset, organisaationaliset jne.).



Kuva 5. VATAM-kehys (Nykänen 2005)

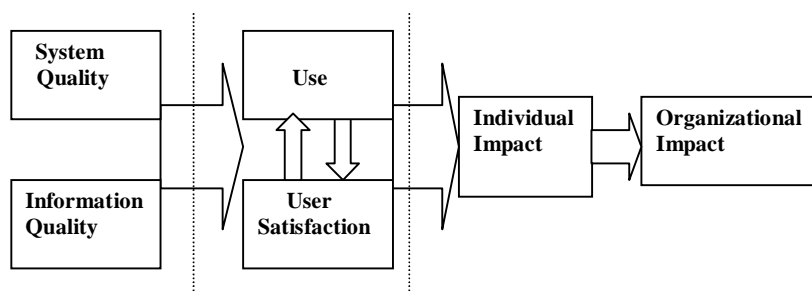
Ennen varsinaisen arviointisuunnitelman tekoa tarkennetaan, mitä osia suosituksesta käytetään, ja määritellään, missä tuotteen/palvelun elinkaaren vaiheessa tutkimus toteutetaan, millaisia aikaansaannoksia (esim. toimittajan ja asiakkaan yhteinen projekti, uusi kokeiluvaiheessa oleva tuote) tutkitaan sekä arvioidaanko sitä, millaisia vaikutuksia tuotteella on käyttöympäristössä. Tämä ohjaa päätöksiä arvioinnin laajuudesta ja syvyydestä. Keskeinen kysymys on pohtia miksi tuote on kehitetty/harkittu/hankittu. Myös prosessi, johon tuote on tarkoitettu sijoitettavaksi, on kuvattava niin tarkasti, että sovelluksen vaikutukset voidaan tunnistaa. Kun arvioinnin tarkoituksena on tarjota tietoa tuotteen ominaisuuksista, on kysymys summatiivisesta arviosta. Kun on kyse tiedosta, jota käytetään tuotteen kehittämisessä, puhutaan formatiivisesta arviosta. Formattiivinen arviointi tarjoaa yleensä tuloksia, jotka ovat voimassa vain kohtalaisen lyhyen aikaa.

Kattavien ja helppokäyttöisten suositusten antaminen tietojärjestelmien arvioimiseen on vaikeaa, koska arvioinnissa kohdataan hyvin vaihtelevia tilanteita ja erityyppisiä vastauksia vaativia kysymyksiä. VATAM -suosituksissa arvioinnissa käytetään seuraavia tietolähteitä: kirjallisuus, projektikuvaukset, menetelmät, kyselylomakkeet ja terminologia. Kirjallisuuslähteinä käytetään arviointitutkimuksista julkaistuja artikkeleita, tarkkoja tietoja järjestelmän kehittämisen vaiheesta, tutkimussuunnitelmaa, havaintojen määrää jne. Projektikuvaukset mahdollistavat toisten käyttäjien tunnistamisen, joilla on samantapaisia ongelmia tai sovelluksia. Menetelminä voidaan käyttää esimerkiksi prosessimallinnusta ja eri tiedonkeräysmenetelmiä, esim. kyselylomakkeita, jolloin on voitava yhdistää tulokset lähteisiin. Väärinymmärtämisen ehkäisemiseksi tarvitaan lisäksi käsitteiden yhtenäistä käyttöä ja määrittelyä.

Terveydenhuollon tietojärjestelmien arvioinnissa kohtaa kolme tieteenalaa: lääketiede, informatiikka ja teknologian arviointi. Arviointi voidaan jakaa kahteen osaan: 1) tiedon resurssien (ICT-sovellus) mittaamiseen ja havainnointiin eri kehitysvaiheissa ja 2) tiedon resursseja käyttävien roolien ja ihmisten käyttäytymisen mittaamiseen ja havainnointiin (summatiivinen/valmiin toimivan järjestelmän arviointi). Peruslähtökohta VATAM-suositusten arviointimallissa on summatiivinen, eli se käsittelee arvioitavia tuotteita tai sovelluspalveluita niiden vaikutusten ja käyttöarvon perusteella, eikä esimerkiksi formatiivisen arvioinnin tekniseen varmennukseen keskitytä. Tätä arviointia voi kuitenkin tehdä myös kehitystyön aikana.

4.1.3 DeLone ja McLean -mallit

DeLone ja McLean (DeLone & McLean 2003) ovat esittäneet lähestymistavan tietojärjestelmien arviointiin. Tietojärjestelmiin on luotu ominaisuuksia, joita voidaan kuvata eriasteisilla osoituksilla järjestelmän ja informaation laadusta. Käyttäjät ja johtajat voivat olla tyytyväisiä tai tyytymättömiä ominaisuuksien mukaiseen järjestelmään tai sen tuottamaan informaatioon. Arviointi perustuu järjestelmän laatuun, käyttöön ja vaikutuksiin heidän mukaansa seuraavasti: systeemin laatu ja informaation laatu vaikuttavat merkittävästi yhdessä käyttöön ja käyttäjätyytyväisyyteen. Lisäksi käyttö ja käyttäjätyytyväisyys vaikuttavat positiivisesti tai negatiivisesti toisiinsa. Käyttö ja käyttäjätyytyväisyys vaikuttavat suoraan henkilökohtaiseen suorituskäyttöön ja työtappoihin, ja eri käyttäjien vaikutukset saavat yhdessä aikaan organisaation vaikutuksen (Turunen 2001, 72; DeLone & McLean 2003, 10–11) (ks. kuva 6).



Kuva 6. IS Success Model 1992 mukailtu malli (DeLone & McLean 2003, 12)

Keskeisiä ajatuksia DeLonen ja McLeanin alkuperäisessä esityksessä olivat:

- 1) Tietojärjestelmien onnistumisen moniulotteisuus vaatii tarkkoja huomioita eri näkökulmista määriteltäessä ja arvioitaessa onnistumiseen vaikuttavia muuttujia. On tärkeää arvioida onnistumisen edellytyksiä ja vaikutuksia yhdestä tai useammasta onnistumisen näkökulmasta ja tunnistaa eri muuttujien vaikutukset.
- 2) Onnistumisen näkökulman valitsemisen ja arvioimisen on nojaututtava objektiiviseen ja empiiriseen tutkimukseen. Lisäksi luotettavuuden parantamiseksi on käytettävä onnistumisen tarkkaa testaamista ja todistettavaa mittaamista aina kun mahdollista.
- 3) Onnistumisen moniulotteisuudesta huolimatta tulee yrittää vähentää käytettyjä vertailuperusteita mitattaessa tietojärjestelmän onnistumista, jotta tutkimustulokset voivat olla vertailukelpoisia ja löydökset perusteltuja.

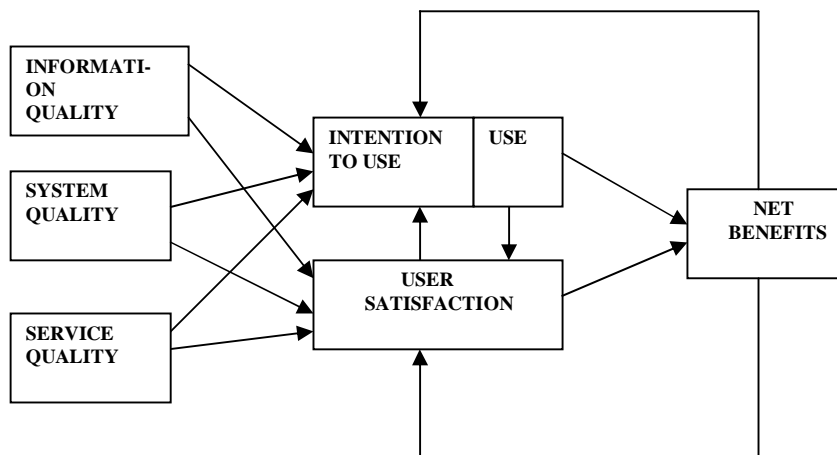
Lisäksi kirjoittajat korostivat, että organisaation vaikutusten arviointiin tarvitaan enemmän tutkimusta, ja että heidän alkuperäistä malliaan on kehitettävä edelleen.

Tähän työhön liittyen onnistumisen näkökulmiksi voidaan valita joitakin palveluarkkitehtuurilla ja web-sovelluspalveluilla tavoitelluista hyödyistä. Tavoitteiden saavuttamisen arviointiperusteita esitetään seuraavissa luvuissa. Organisaation vaikutusten arviointiin kiinnitetään huomiota varsinkin asiakasorganisaation ja käyttäjien tavoitteiden toteutumisen arvioinnissa.

Jokainen laadun ulottuvuus saa erilaisen painoarvon riippuen analyysin tasosta. Arvioitaessa yhden tietojärjestelmän onnistumista informaation laatu ja järjestelmän laatu ovat tärkeimpiä osatekijöitä.

Arvioitaessa onnistumista tietojärjestelmäkokonaisuudessa, toisin kuin yhtä järjestelmää, voi palvelun laatu tulla tärkeimmäksi osatekijäksi. Lisäksi määrittelyn tarkkuus ja mallin soveltamistapa riippuvat käyttökontekstista.

Kuvassa 7 on esitetty laatu-käsitteen kolme ulottuvuutta: "informaation laatu", "järjestelmän laatu" ja "palvelun laatu". Jokaista tulee arvioida erikseen, koska yksin tai yhdessä ne vaikuttavat myöhempään "käyttöön" (käyttäytymistä) ja "käyttäjätyytyväisyyteen". D&M:n mukaan joskus kannattaa arvioida myös "aikomusta käyttää". Tätä asennetta ja sen yhteyttä käyttäytymiseen on vaikea arvioida ja siksi monet tutkimukset keskittyvät "käyttöön". Käyttö edeltää käyttäjätyytyväisyyttä, ja positiiviset kokemukset käytöstä lisäävät sitä. Lisääntyvä käyttäjätyytyväisyys johtaa lisääntyvään "aikomukseen käyttää" ja siten käyttöön. Tuloksena käytöstä ja käyttäjätyytyväisyydestä syntyy lopullinen hyöty (nettohyöty). Käytöstä saatava palaute voi olla myös negatiivista, mutta se on silti yhtä pätevää. Lisäksi käyttäjätyytyväisyyttä ja hyötyjä voidaan tarkastella eri sidosryhmien kannalta: esimerkiksi potilaan, terveydenhuollossa toimivan ammattilaisen tai organisaation tai sen johdon näkökulmista.



Kuva 7. Updated IS Success Model mukailtu malli (DeLone & McLean 2003, 24)

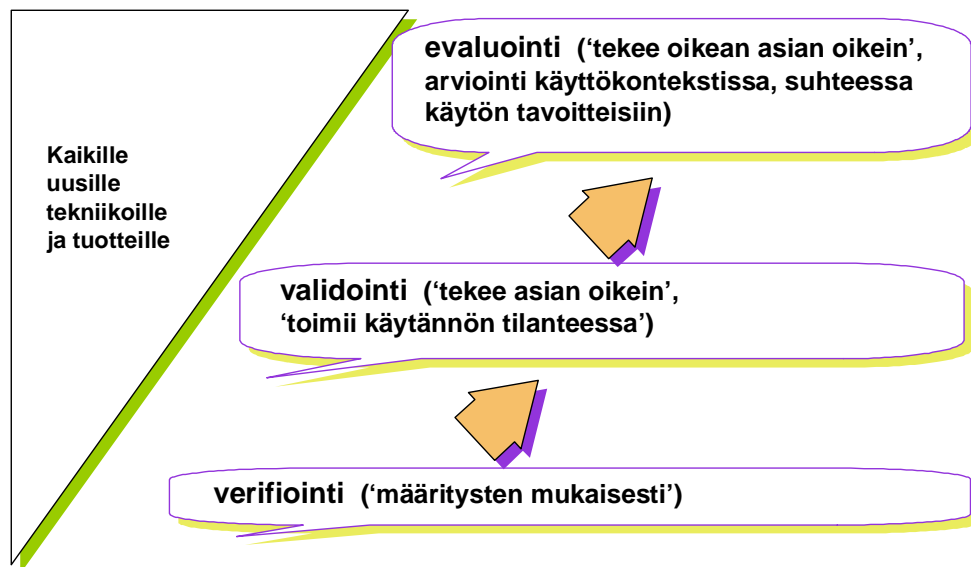
Uuden teknologian vaikutuksia tyytyväisyyteen sekä sosiaalisen ja työelämän laatuun on tutkittu mm. toimistotyöhön liittyen, jolloin on havaittu lisääntynyttä itsenäisyyttä, enemmän vapautta työhön ja lisääntyvää vastuuta työntuloksista; käyttäjistä tuntuu, että heidän työnsä on rikastunut. (Igarria, Tan 1997). Käyttäjien työn laadun paranemista, työn helpottumista, ajansäästöjä sekä työn tarpeiden ja vaatimusten toteuttamisen helpottumista on myös saatu aikaan, mutta tekijöidenkin mukaan tuloksiin on suhtauduttava varauksellisesti (Etezadi-Amoli, Farhoomand 1996, 66 ja 72). Tämä tutkimus oli toteutettu EUCS -mittaria (Doll ym. 1988) soveltaen.

Tietojärjestelmien välittömässä käytössä tapahtuvan vaikutusten arvioinnin lisäksi on tarpeen arvioida myös muita tietojärjestelmien vaikutuksia. Tällaisia ovat esimerkiksi ryhmätyön vaikutukset, organisaation sisäiset vaikutukset, teolliset ja tuotannolliset vaikutukset, asiakasvaikutukset ja yhteiskunnalliset vaikutukset. Tietojärjestelmien toiminta vaikuttaa monin tavoin eri tasoilla yksilötaloudesta kansantalouteen. Se, missä vaikutuksia arvioidaan, tulisi riippua järjestelmästä tai järjestelmän tarkoituksesta. Lopullisten hyötyjen arviointi on kaikkein tärkeintä, mutta niitä ei voida analysoida ja ymmärtää ilman järjestelmän laadun ja informaation laadun arviointia. Lopullisia hyötyjä arvioitaessa on otettava huomioon erityisesti miten lopullinen hyöty määritellään ("mikään tulos ei ole täysin positiivinen ilman negatiivisia seurauksia"), kenen (suunnittelijan, rahoittajan, käyttäjän,

joku muu) hyödystä on kysymys ja minkä tasoisella analyysillä (yksilö, työnantaja, teollisuus/tuotanto) hyötyä mitataan.

4.1.4 Muita arviointimalleja

Tietojärjestelmien arvioinnissa ja "oikeellisuuden" tutkimisessa voidaan tunnistaa useita tasoja (ks. kuva 8., Saranummi 2003). Järjestelmää tai ratkaisua voidaan tarkastella määritysten mukaisuuden suhteen, esim. toteuttaako se määritellyt vaatimukset tai rajapinnat, esim. testaamalla (verification, todentaminen). Kattavammassa validoinnissa (validation) tarkastellaan sitä, toimiiko ratkaisu oikein käyttöympäristössään suorittaen haluttuja toimintoja. Varsinainen arviointi (evaluation) tutkii, saavutetaanko oikein toimivan ratkaisun käytöllä haluttuja vaikutuksia toiminnassa. Todentamista ja validointia on yleensä mahdollista tehdä rajatusti kehitystyön ja käyttöönoton aikana, kun taas kattava arviointi on usein mahdollista vasta vakiintuneesta käytöstä saatujen kokemusten analysoinnin avulla. Todentamisessa voidaan yleensä rajautua tarkastelemaan lähinnä tuotteiden toiminnan oikeellisuutta suhteessa määriteltyihin tarkkoihin vaatimuksiin tai järjestelmän tai rajapintojen määrittelydokumentteihin. Validoinnissa arvioidaan ratkaisujen toimivuutta siten, että saadaan tietoja myös siitä, onko käyttöönotto onnistunut ja että järjestelmä toimii myös lopullisessa käyttöympäristössä. Arvioinnissa tarkastellaan laajemmin järjestelmän vaikutuksia organisaation tavoitteisiin, käyttäjien toimintaan ja myös välillisiä vaikutuksia.



Kuva 8. Arvioinnin tasot (Saranummi 2003)

Myös Turunen on väitöskirjassaan kuvannut tietojärjestelmien vaikutusten arviointia erityisesti terveydenhuollon tietojärjestelmien näkökulmasta. Hänen jaottelunsa koostuu:

1. teknisen laadun arvioinnista, jossa arvioidaan mm. tarkkuutta, luotettavuutta, vastausaikaa ja joustavuutta esim. käyttäjätyytyväisyysmittareilla.
2. henkilökohtaiseen suorituskykyyn kohdistuvien vaikutusten arvioinnista, mm. vaikutukset diagnoosiin, päätöksentekoon, hoitoon, ajankäyttöön. Myös näitä voidaan arvioida esim. käyttäjätyytyväisyysmittauksilla.
3. organisaatio- ja potilasvaikutusten arvioinnista, joka kohdistuu mm. terveysvaikutuksiin, palvelujen määrään, henkilökunnan määrään, prosessin mittaamiseen, sijoitetun pääoman tuottoon jne.

4. yhteiskunnallisten vaikutusten arvioinnista, jossa kohteina voisivat olla mm. väestön työkyky, kansallinen hyvinvointi ja palvelujen vienti.

Arviointitasojen väliset rajat eivät ole absoluuttisia, vaan esimerkiksi osa käyttäjätyytyväisyyskysymyksistä voidaan ajatella kuuluvaksi tekniseen osaan ja osa henkilökohtaiseen suorituskykyyn. Mitattaessa vaikutuksia organisaatioon täytyy käyttää kokonaisvaltaisesti asioita tarkastelevia mittareita, esim. prosessien mittaamista tai kustannus-hyöty analyysia (Turunen 2001)

Tässä työssä arvioitavat tavoitteet ja hyödyt määritellään myöhemmissä kappaleissa ja niitä arvioidaan etenkin terveydenhuollon organisaatioiden keskeisten sidosryhmien (käyttäjät, jotka ovat useimmiten terveydenhuollon ammattilaisia, tietohallinto, johto) sekä sovellusten kehittäjien (sovelluksia ja palveluja tuottavat yritykset) näkökulmasta. Arviointi keskittyy yksilöiden ja organisaatioiden toimintaan eikä kata laajempia välillisiä tai yhteiskunnallisia vaikutuksia. Arvioinnin eri tasojen suhteen tässä työssä pyritään tukemaan "validation" ja "evaluation" -tasoja, eli esittämään malleja sille, kuinka ratkaisujen hyötyjä arvioidaan niiden oikeellisuuden ja vaikutusten kautta.

4.2 Palvelupohjaisten ratkaisujen laadun muodostuminen

Tässä luvussa sovelletaan DeLonen ja McLeanin uudempaa (DeLone & McLean 2003, 24) arviointimallia suhteessa palvelupohjaisiin arkkitehtuureihin ja ratkaisuihin.

Informaation laatu

Tutkimuksissa on havaittu merkittävä yhteys informaation laadun ja yksilöllisten vaikutusten välillä (mm. Etezadi-Amoli, Farhoomand 1996; Teo and Wong 1998). Sen on myös osoitettu olevan vahvasti yhteydessä järjestelmän käytön ja lopullisten hyötyjen kanssa varsinkin verkkokaupankäynnissä. Informaation laatua on arvioitu virheettömyyden, oikea-aikaisuuden, täydellisyyden, johdonmukaisuuden ja tilanteeseen sopivuuden (relevanssin) kannalta. Esimerkiksi verkkokaupassa informaation laadussa tärkeitä ovat tiedon täydellisyys, helppo ymmärrettävyys, henkilökohtaisuus, tärkeys ja turvallisuus. Yksilöllisiä vaikutuksia on arvioitu päätöksenteon, tehtävän tuloksellisuuden ja työn laadun kannalta (DeLone & McLean 2003.) Myös tiedon vertailtavuus ja mu-kautuvuus vaikuttavat informaation laatuun.

Saundersin ja Jonesin tekemässä tutkimuksessa on todettu, että usein yritykset aloittavat tietojärjestelmien käyttöönoton saadakseen parempaa informaatiota päätöksentekoon. Informaation laadun ajatellaan parantuvan, koska tietojärjestelmien avulla tietoa on helpompi päivittää, käyttää ja muokata ja sitä kautta voidaan tarjota merkityksellistä informaatiota päätöksentekoon.(Teo and Wong 1998)

Teon ja Wongin mukaan informaation laatuun kuuluu sen hyöty, virheettömyys, oikea-aikaisuus ja merkityksellisyys, ja sillä on näin merkittävä yhteys käytettävyyteen. Virheellisellä, epätäsmällisellä, vaikeaselkoisella tai ajankohdan kannalta merkityksettömällä tiedolla voi olla vahingollisia seurauksia johdon seurannalle, päätöksenteolle kuten myös kustannussäästöjen tai tuotannollisen kehityksen suhteen. Yhteystietotekniikkainvestointien ja informaation laadun, työympäristön kohenemisen sekä organisaation vaikutusten välillä on vaikeaa todistaa - suurempi panostus IT-investointeihin ei välttämättä johda parannuksiin. Lisäksi monet muut tekijät vaikuttavat todennäköisesti arviointiin, kuten johdon tuki, johtajuus ja ympäristön muutos. Informaation laadulla näyttää kuitenkin olevan yhteys johdon tyytyväisyyteen, työympäristön parantumiseen ja organisaatio-

naalisiin vaikutuksiin. Loppukäyttäjien lisäksi tyytyväisyyden arviointi johdon näkökulmasta on tärkeää.

Palvelupohjaisissa ratkaisuissa informaation laatu muodostuu yksittäisten sovelluspalvelujen lisäksi siitä, kuinka erityyppisten palvelujen tarjoamia ja käsittelemiä tietoja on yhdistetty eri prosesseissa ja sovelluksissa. Parempaa informaatiota päätöksentekoon voidaan palvelupohjaisissa ratkaisuissa tuottaa esim. prosessien keskitetyn ohjauksen ja seurannan kautta. Informaation virheettömyys paranee esimerkiksi vähentyneen tietojen päällekkäisyyden kautta. Oikea-aikaisuutta ja tilanteeseen sopivuutta edistää eri tietojen saatavuus osaksi uusia ratkaisuja entistä vähemmällä kehitystyöllä ja nopeammalla kehityssyklillä. Tietojen täydellisyys ja johdonmukaisuus eivät pääosin ole yleisiä arkkitehtuuriin kohdistuvia vaatimuksia vaan riippuvat soveltamiskohteesta. Informaation hyödyn, virheettömyyden, oikea-aikaisuuden ja merkityksellisuuden muodostumiseen liittyvät eniten järjestelmien sisältöön ja käytettävyyteen liittyvät ratkaisut, joille SOA -lähestymistapa vain rakentaa edellytyksiä.

Järjestelmän laatu

Tutkimuksissa (mm. Etezadi-Amoli, Farhoomand 1996; Teo and Wong 1998) on löydetty tilastollisesti merkittävä yhteys järjestelmän laadun ja yksilöllisten vaikutusten välillä. Järjestelmän laatua arvioitiin helppokäyttöisyyden, toiminnallisuuden (järjestelmällä on positiivisia vaikutuksia suoritukseen/suorituskykyyn vain, kun on olemassa vastaavuus niiden toiminnallisuuden ja käyttäjän työn vaatimusten välillä), luotettavuuden, joustavuuden, yhdisteltävyyden (integroitavuuden) ja merkityksen (järjestelmän käytön merkitys organisaatiossa/kuinka tärkeä järjestelmä on organisaatiolle/organisaation tavoitteille) kannalta. Esimerkiksi verkkokaupankäynnin osalta tärkeitä piirteitä ovat järjestelmän sovellettavuus, saatavuus, luotettavuus, reaktionopeus ja käytettävyyden. Yksilöllisinä vaikutuksina on tarkasteltu työympäristön laatua ja suorituskykyä. Tietojärjestelmän yksilövaikutusten arviointiin on käytetty esimerkiksi neljästä päätekijästä koostuvaa mittaria. Yksilöllisiä vaikutuksia mm. ovat: 1) työn tuottavuus - kuinka paljon sovellus parantaa "tuotosta"/aikayksikkö, 2) työn uudistuminen - minkä verran sovellus auttaa käyttäjää luomaan ja kokeilemaan uusia ideoita työssään, 3) asiakastytyväisyys - minkä verran sovellus auttaa käyttäjää luomaan merkitystä yrityksen sisäisille tai ulkoisille asiakkaille ja 4) johdon seuranta - kuinka sovellus auttaa ohjaamaan työprosesseja ja suorituskykyä (DeLone & McLean 2003)

Palveluarkkitehtuurin ja web-sovelluspalvelujen monet keskeiset lupaukset liittyvät järjestelmän laatutekijöihin. Helppo päivitettävyyden, käyttö ja muokkaus sekä kokonaisjärjestelmän muutosherkkyys ovat keskeisimpien tavoiteltujen hyötyjen joukossa. Tietojärjestelmän toiminnallisuuden joustavuudella on huomattavaa merkitystä organisaation työn tuottavuuteen, työn uudistumiseen ja asiakastytyväisyyteen. Lisäksi parannetaan johdon seurantamahdollisuuksia. SOA:an liittyvillä käyttöliittymäratkaisuilla (esim. samojen palveluiden käyttö erilaisilla päätelaitteilla ja käyttöliittymillä) pyritään parantamaan tietojärjestelmien helppokäyttöisyyttä, käytettävyyttä ja tiedon saatavuutta. SOA -lähestymistavan keskeinen piirre on järjestelmäkokonaisuuden vähittäinen ja inkrementaalinen kehitys, jolla pyritään vastaamaan joustavuus- ja yhdisteltävyyden vaatimuksiin ja varmistamaan ratkaisujen sovellettavuus. Palveluratkaisujen ja toiminnan käsitteiden ja abstraktiotason lähentäminen voi tukea työn uudistumista ja tuottavuutta, vähentäen kehittämistyön ja työtoiminnan välimatkaa.

Palvelun laatu

Tietojärjestelmän tehokkuuden arviointiin on sisällytettävä sen palvelun laadun arviointi. Palvelun laadun arvioinnissa on käytetty esimerkiksi alun perin markkinointitutkimuksen yleistä palveluiden laatua arvioivaa SERVQUAL arviointia, jota on sovellettu tietojärjestelmäympäristöön sopivaksi.

22 -kohtaisessa mittarissa on viisi ulottuvuutta. Nämä ulottuvuudet ovat alun perin 1) palvelun konkreettisuus, jossa kuvataan ulkoiset ja aineelliset ominaisuudet, välineet ja henkilökunta, 2) luotettavuus, joka kuvaa kykyä suorittaa luvattu palvelu luotettavasti ja virheettömästi, 3) herkkyys/reagoivuus on valmiutta auttaa asiakasta ja antaa täsmällistä palvelua, 4) varmuuteen puolestaan kuuluu työntekijöiden tietämys ja huomaavaisuus sekä heidän kykynsä herättää luottamusta ja varmuutta sekä 5) empatia, josta kuvastuu välittäminen ja asiakkaan yksilöllinen huomioiminen (tietojärjestelmien osalta esim. käyttäjäystävällisyys ja kuinka vastaa reaali maailman toimintaa). Muun muassa verkkokaupan palveluiden laatua on arvioitu varmistettavuudella, luottamuksellisuu-della, käyttäjäystävällisyydellä ja palvelun käyttäjälle tarjoaman avun määrällä. (DeLone & McLean 2003)

Servqual-mittariston palvelun laatumittarit kuvaavat lähinnä ihmisten tarjoamia palveluita. Niitä ei voi suoraan soveltaa tietoteknisiin ratkaisuihin. Palvelun laatua kuvataan palvelun luotettavuudella ja varmuudella, jotka ovat myös uudelleen käytön keskeisiä tavoitteita.

Nettohyöty

(DeLone ja McLeanin artikkelin ((DeLone & McLean 2003) mukaan organisaationaalaisia vaikutuksia on arvioitu erilaisissa tutkimuksissa. Esimerkiksi Miranin ja Lede-rin kehittämällä 33 -kohtaisella mittaristolla on arvioitu tietojärjestelmäprojekteista saatavaa hyötyä. Mittaristo kuvaa organisaation hyötyjen koostuvan kolmesta pääosasta - strateginen hyöty, tiedollinen hyöty ja liiketoiminnallinen hyöty. Strategisiin hyötyihin kuuluvat mm. kilpailuetu, yhdistymiseen liittyvät ja asiakassuhteisiin liittyvät hyödyt. Tiedollisiin hyötyihin sisältyvät informaation reitit, informaation laatu ja informaation joustavuus. Liiketoiminnallisiin hyötyihin sisältyy tiedonvälityksen tehokkuus, järjestelmän kehittämisen tehokkuus ja liiketoiminnan tehokkuus. Kaplanin ja Nortontin Balanced Scorecard (BSC) lähestymistapaa on myös käytetty arvioitaessa organisaationaalista suorituskykyä. Tähän lähestymistapaan kuuluvat liiketoiminnan näkökulma, asiakasnäkökulma, sisäiset liiketoimintaprosessit, oppimisen ja kasvun näkökulmat. Näitä näkökulmia on sovellettu myös tietojärjestelmäympäristöön. Tällöin arviointi sisältää liiketoiminnan laadun arvioinnin näkökulman, käyttäjälähtöisen näkökulman, sisäisten prosessien näkökulman ja tulevaisuuteen suuntautumisen näkökulman ja joista jokaiselle osa-alueelle suositellaan omaa spesifiä arviointia. Tietojärjestelmän organisaationaalaisia vaikutuksia voidaan tarkastella myös teoreettiselta pohjalta mm. suorituskyvyn, tehokkuuden ja ymmärtämisen näkökulmista. Arviointia voisi kohdistaa myös esimerkiksi ohjelmistojen arviointiin, organisaationaaliseen käyttäytymiseen, sosiologian ja/tai kognitiivisen psykologian alueille. Esimerkiksi verkkokaupassa lopulliseksi hyödyiksi on nähty kustannussäästöt, laajenevat markkinat, kasvava myynti, supistuvat etsintäkustannukset ja ajan säästö.

Palveluarkkitehtuurin tarjoaman toiminnallisen joustavuuden avulla saavutettavaksi organisaation nettohyödyksi ajatellaan yleensä tiedonvälityksen tehokkuutta, liiketoiminnan tehokkuutta ja kilpailuetua. Myös organisaatiomuutosten ja yhdistymisten tapahtuessa modulaarinen järjestelmä on helpommin muokattavissa vastaamaan muuttunutta tilannetta kuin monoliittiset sovellukset. SOA -ratkaisulla pyritään lisäämään informaation reittien ja tiedonvälityksen sekä järjestelmien kehittämisen tehokkuutta. SOA -lähestymistapaan liitetty prosessien määrittely ja tukeminen pyrkii vastaamaan sisäisten prosessien kehittämistarpeisiin. Arkkitehtuurikehitystä pyritään aina tekemään tulevaisuuteen suuntautuen, ja järjestelmän kehittämisen tehokkuus (etenkin nopean muutoksiin reagoivuuden kautta) on keskeinen SOA -tavoite. Palveluarkkitehtuuri ja sen yhteydessä tehtävä prosessien määrittely tarjoavat myös mahdollisuuksia seurata automaattisesti haluttujen toiminnan mittarien kehittymistä tietojärjestelmäympäristössä. Usein palveluarkkitehtuurin kehittämiseen liittykin BAM (Business Activity Monitoring) -ratkaisujen kehittäminen (Webmethods 2006).

IT -palveluiden hallinnassa ja niiden arvioinnissa voidaan hyödyntää esim. IT Infrastructure Library:n (ITIL) mittareita. ITIL:in perusajatuksen mukaan IT -operaatioiden lähtökohtana ovat liiketoiminnan tarpeet ja IT -palvelun laatua mitataan käyttäjiä havainnoimalla. Kuitenkin useat organisaatiot raportoivat ainoastaan teknisiä mittaustuloksia (esim. hävikki), jotka ovat sinällään tärkeitä, mutta joista ei välttämättä voida päätellä, kuinka palvelu vastaa käyttäjän tarpeisiin sekä millainen palvelun laatu on. IT -palvelun laatua voidaan mitata esim. IUM -mittarilla (Impacted User Minutes), joka sisältää seuraavat näkökulmat: palvelun saatavuus, kapasiteetti, turvallisuus ja tiedon eheys (Availability, Capacity, Security, and Data Integrity). (Marquis, 2006).

Lisäksi ITIL tarjoaa paljon muita yleisiä mittareita, joita voidaan käyttää apuna IT -palveluiden arvioinnissa, kuten kehittämis- ja muutospyyntöjen määrä, virhetilanteiden määrä, virheistä toipumisaika sekä työmäärä ja kustannukset (ks. myös alla oleva taulukko) (Central Computer & Telecommunications Agency 2002; OGC 2002; OGC 2004).

Esimerkkejä ITIL:issä käytetyistä mittareista ovat mm.:

- MTBF (Mean Time Between Failures) = keskimääräinen aika, siitä kun palvelu on käytettävissä ja esiintyy seuraava virhe. (Käytössä-----Virhe)
- MTBSI (Mean Time Between System Incidents) = keskimääräinen aika virheiden esiintymisen välillä. (Virhe-----Virhe)
- MTTR (Mean Time To Repair) = keskimääräinen aika virheiden esiintymisestä sen korjaukseen. (Virhe-----Käytössä).

4.3 Palveluarkkitehtuurin ja Web-sovelluspalvelujen tavoiteltujen hyötyjen arviointi

Luvussa 2 on käsitelty tarkemmin palveluarkkitehtuurin ja Web-sovelluspalvelujen tavoiteltuja hyötyjä, hyötyjen arviointia ja arvioinnissa käytettäviä mittareita. Erityisesti kappaleiden 2.1–2.3 loppuun on koottu tietojärjestelmien arviointiin vaikuttavia seikkoja ja mittareita, jotka pohjautuvat tunnistettuihin tutkimuksissa käytettyihin informaation laadun, järjestelmän laadun, palvelun laadun ja nettohyödyn tai organisaation hyödyn attribuutteihin. Seuraavissa luvuissa on taulukoihin koostetussa muodossa arviointimittareita tunnistettuihin hyötyihin ja esimerkki mittarien valinnasta valitun kohteen arviointiin.

4.3.1 Kooste arviointimittareista

Tässä luvussa esitetään kooste arviointimittareista, joista osa on kuvattu luvussa 2. Mittarit on ryhmitelty seitsemään taulukkoon mitattavan kohteen perusteella:

- käyttäjä- ja asiakastyytyväisyyden mittarit,
- käytettävyys- ja saatavuusmittarit, jotka vaikuttavat usein keskeisesti myös käyttäjä- ja asiakastyytyväisyyteen,
- toimintaprosessien mittarit, joilla tarkastellaan toimintojen seuranta ja muuttumista ratkaisujen käyttöympäristössä,
- tietojen mittarit, joilla tarkastellaan tietomääriä, soveltuvuutta, riittävyyttä, vastaavuuksia sekä tietoihin liittyvää työtä,
- kehitysprosessin mittarit 1) toimittajan ja 2) asiakkaan tai integraattorin näkökulmasta,
- tekniset mittarit.

Koosteeseen on valittu mittareita ja suureita, joita voidaan hyödyntää palveluarkkitehtuuriin kehittämisen ja sillä tavoiteltujen tavoitteiden mittaamisessa. Eri osa-alueista ei ole yritetty koota kaikenkattavia mittaristoja.

Kussakin taulukossa mittarit on ryhmitelty mittarin tyyppin perusteella:

- lukumäärämittarit, jotka kuvaavat laskettavia suureita, jotka voivat olla usein myös automaattisesti saatavissa;
- työmäärä- ja ajalliset mittarit, jotka sisältävät toisaalta henkilötyöhön liittyviä, toisaalta järjestelmän toimintaan ajallisesti liittyviä mittareita;
- laadulliset mittarit, jotka ovat usein ennen/jälkeen tilanteeseen liittyviä muutoksen mittareita, tai subjektiivisia kokemuksia ja mielipiteitä mittaavia,
- taloudelliset mittarit.

Osa taulukoissa esitetyistä asioista on yksinkertaisia suureita, osa suhteellisen monimutkaisesti muodostuvia mittareita. Osa mittareista on selvästi ja helposti mitattavissa, osa nojautuu esim. asiointijoiden tai käyttäjien tekemiin arvioihin. Tyypillisesti pyritään löytämään ainakin muutamia selvästi määrällisiä mittareita, jotka mittaavat suoraan haluttujen tavoitteiden toteutumista, jotta saavutetaan luotettavaa mittaustietoa. Laadullisia ja välillisiä mittareita käytetään usein tukemaan määrällisiä mittareita tai mittaamaan vaikeammin todettavissa olevia tekijöitä tai tunnistamaan eri tekijöiden välisiä suhteita. Yleisimpiä ja laajalti käytettyjä mittareita on kursivoitu taulukoihin.

Teknisillä ja lukumäärämittareilla ei välttämättä ole suoraan yhteyttä tai näkyvyyttä käyttäjään, esimerkiksi virhelokeihin tulevat tapahtumat tai taustalla olevien palvelujen lukumäärät eivät näy suoraan käyttäjille.

Työmäärämittareihin voi lisätä "muodostetaan tai mitataan eri tehtäviin tarkalleen tai keskimäärin kuluva aika" monissa tilanteissa esim. lukumäärämittarien pohjalta.

Taloudellisia mittareita voi muodostaa määrittelemällä hinta tai tuotto lukumäärä- ja työmäärämitta-reiden yksiköille. Toisaalta taloudellisissa mittareissa on usein löydettävissä helpommin säästöjä tai tuottoja mittaavia seikkoja. Lisäksi taloudellisissa vertailuissa voidaan myös käyttää käännteistä arviointia, esim. "paljonko kustannuksia aiheuttaa, kun tietty palvelu tai tieto ei olekaan saatavilla". Yleiset taloudelliset mittarit on sijoitettu taulukkoon "toimintaprosessit".

Taulukko 4.1. Käyttäjä- ja asiakastyytyvyyden mittareita.

Mitattava kokonaisuus: Käyttäjä- ja asiakastyytyvyys	Lukumäärämittarit	Työmäärä- ja ajalliset mittarit	Laadulliset mittarit	Taloudelliset mittarit
	sovellusten ja palvelujen käytön määrä ja sen muutos	käyttäjien ajankäytön seuranta (opettelu, käyttö)	käyttäjien kokemat työnkuvan ja työmäärän muutokset	<i>liiketoiminnan mittarit</i>
	kehittämisen- ja muutospyyntöjen määrä	palvelun saatavuuden aikataulu vaatimuksesta käyttöönottoon	muutokset työnkulkuun ja kommunikaatioon	
	kuinka moneen kertaan samoja tietoja on syötettävä eri järjestelmiin	samojen tietojen päällekkäiseen syöttöön käytettävä aika	<i>asiakas- ja käyttäjätyytyväisyyskyselyjen avoimet kysymykset</i>	
			EUCS mm. saatavan informaation laatu, selkeys, tulosteiden käyttökelpoisuus, tiedon oikea-aikaisuus	

Taulukko 4.2. Käytettävyys- ja saatavuusmittareita.

Mitattava kokonaisuus: käytettävyys ja saatavuus	Lukumäärämittarit	Työmäärä- ja ajalliset mittarit	Laadulliset mittarit	Taloudelliset mittarit
	käyttäjien tekemien virheiden määrä	järjestelmän käyttöön työajasta / tapausajasta käytetty osuus	luotettavuus ja skaalattavuusmittarit	IUM - impacted user minutes (ITIL)
	ks. myös käyttäjä- ja asiakastyytyvyys	<i>järjestelmän / palvelun saatavuus (availability) %</i>	eri käyttötilanteiden ja käytön edellytysten kuvaus	CoD - cost of downtime
		palvelujen ja sovellusten vasteajat	koulutuksen ja tiedon saatavuus	
		virheistä toipumisai- ka (MTTR)	EUCS - mm. kokemukset tietojärjestelmän käytöstä ja käytettävyydestä	
			palvelun soveltuvuus (miltä osin soveltuu käyttöön, miltä osin ei)	

Taulukko 4.3. Toimintaprosesseihin liittyviä mittareita.

Mitattava kokonaisuus: toimintaprosessit	Lukumäärämittarit	Työmäärä- ja ajalliset mittarit	Laadulliset mittarit	Taloudelliset mittarit
	<i>palveltujen asiakkaiden lukumäärä / aikayksikkö</i>	<i>prosessin tai sen jonkin vaiheen läpimenoaika</i>	käyttäjäkysymykset prosessien ja työn muutoksista	palvelun kustannukset ja tuotot
	prosessin käynnistysten, läpimenojen, tapahtumien tai vaiheiden suorituksen lukumäärä	uusien osapuolien mukaan ottamisen nopeus prosesseissa	palvelun koko ja vastavuus toiminnan kanssa (esim. sisältöanalyysi)	kassavirta-analyysit: nettonykyarvo ja sisäinen korkokanta
	prosessimäärittelyjen lukumäärä	ilman järjestelmämuutoksia (keskitettyihin prosessimäärittelyihin) tehtävien muutosten osuus	asiantuntemuksen laatu ja määrä prosessimuutosten tekemiseksi	<i>takaisinmaksuaika</i>
	prosessivaiheiden lukumäärä	prosessin tehollisen työajan määrä ja osuus suhteessa läpimenoaikaan	prosessien seurantatietojen saatavuus	<i>sijoitetun pääoman tuotto</i>
	katettujen toimintojen lukumäärä	odotusaika	tietojärjestelmämuutosten määrä ja laatu	toiminnan virheiden lukumäärä, niiden muutos ja kustannukset
	prosessiin määriteltävien poikkeusten (tapahtumien) lukumäärä		ratkaisun eri osien riippuvuudet	prosessin tuotto (process revenue)
	kuinka monessa eri tilanteessa/prosessissa tai sovelluksessa tiettyä ratkaisua on käytetty		onko prosessi määritelty (ja millä tasolla)	
	kuinka suuri osa prosesseista menee läpi (/ menee läpi samalla tavalla)			

Taulukko 4.4. Järjestelmissä käsiteltyihin tietoihin liittyviä mittareita.

Mitattava kokonaisuus: tiedot	Lukumäärämittarit	Työmäärä- ja ajalliset mittarit	Laadulliset mittarit	Taloudelliset mittarit
	palvelun kautta saatavien tietokokonaisuuksien lukumäärä (record count)	tietojen syöttöön käytetty aika	EUCS: ovatko kaikki tarvittavat tiedot mukana	
	yksittäisten tietoelementtien / kenttien lukumäärä (schema complexity)	tietojen päällekkäiseen / toistuvaan syöttöön käytetty aika	EUCS: onko tiedon muoto käyttöön soveltuva	
	kuinka monessa paikassa säilytetään päällekkäisiä tietoja		tiedon eheys (luotettavuus, oikeellisuus, EUCS: ajantasaisuus)	
	eri lähteistä tulevien tietojen vastaavuusmäärittelyjen (mapping) lukumäärä		käytetäänkö pelkkiä tunnisteita	
			onko tietotyypit määritelty ja yhdenmukaisia	
			EUCS: tarkkuusaste oikea	

Taulukko 4.5. Palvelujen tarjoajan / toteuttajan kehitysprosessin mittareita.

Mitattava kokonaisuus: kehitysprosessi (toimittaja)	Lukumäärämittarit	Työmäärä- ja ajalliset mittarit	Laadulliset mittarit	Taloudelliset mittarit
	kokonaisjärjestelmän osien ja sovelluspalvelujen lukumäärä ja koko	kehitysprosessin ja sen eri vaiheiden työmäärä	kehittäjäkyselyjen avoimet kysymykset	toteutuksen kustannukset
	rajapintojen lukumäärä + yhden rajapinnan eri käyttötilanteiden määrä (tuotteessa tai ympäristössä)	virheiden korjaukseen kuluva aika (MTTR)	käsiteanalyysi käyttäjien ja järjestelmäratkaisujen välillä	sovelluksen uusien osien suunnittelun, kehittämisen ja käyttöönoton kustannukset
	tietojärjestelmämuutosten määrä (ja kuvaus = laadullinen)	testaukseen kuluva aika	kehittämisperiaatteiden saatavuus ja niiden noudattamisen seuranta	saatavilla olevien kehitysympäristöjen hankinta- ja ylläpitokustannukset
	tunnistettujen käyttötilanteiden ja järjestelmien lukumäärä, joihin ratkaisulla saavutetaan liitettävyys	kuinka suuri osa tarvittavista muutoksista saadaan tehtyä keskitettyihin prosessimäärittelyihin	saatavilla olevien kehitysympäristöjen ja välineiden ominaisuudet	versiohallinnan kustannukset
	kuinka monessa eri tilanteessa/prosessissa tai sovelluksessa tiettyä ratkaisua on käytetty	ylläpidon työmäärä/versiohallinnan työmäärä	ovatko palvelut tilallisia (ja mitä tilaa ne säilyttävät)	
	sovelluksen ja palvelun koko (muu kuin osien lukumäärä)	opiskeltavan ja tuotettavan dokumentaation määrä	toimivatko eri välineillä tehdyt palvelut yhdessä	
	ratkaisun teknisten osien määrä	toteutukseen kuluva aika	onko kutsusuhteet tiukasti määritelty	
	ratkaisun eri osien riippuvuuksien lukumäärä (myös erityyppiset riippuvuudet - liitännät, ajalliset riippuvuudet, tietoriippuvuudet)	uusien piirteiden toteuttamisen vaatimuksista käyttöönoton nopeus		
	saatavilla olevien kehitysympäristöjen ja välineiden lukumäärä sekä saatavilla olevien välineiden lukumäärä	käyttöönottoon kuluva aika		
	katettujen käyttötapausten lukumäärä	järjestelmämuutoksiin kuluva aika		
	kuormitustestaus	kuormitustestaus		
	muutospyyntöjen lukumäärä tietyllä aikavälillä			
	toteutettujen muutosten lukumäärä ja suhteellinen määrä muutospyyntöihin			
	järjestelmästä löydettyjen tai korjattavien virheiden määrä			

Taulukko 4.6. Palvelujen hyödyntäjän, asiakkaan ja integraattorin kehitysprosessin mittareita.

Mitattava kokonaisuus: kehitysprosessi (asiakas / integraattori)	Lukumäärämittarit	Työmäärä- ja ajalliset mittarit	Laadulliset mittarit	Taloudelliset mittarit
	uudelleenkäytöstä ja integroinnista aiheutuvien virheiden lukumäärä	käyttöönottoon ja paikalliseen sovitamiseen kuluva aika ja työmäärä	kehittämisperiaatteiden saatavuus ja niiden noudattamisen seuranta	uudelleenkäytöstä aiheutuvat kustannukset ja säästöt
	tietojärjestelmämuutosten määrä ja luonne	vaatimuksista käyttöönottoon kuluva aika	oletetaanko suoritussympäristössä yhteisiä teknisiä ratkaisuja	integroinnista aiheutuvat kustannukset ja säästöt
	integroinnista johtuvien virheiden lukumäärä	käyttökatkot (ja niiden eri luokkien) määrä / saatavuus ym.	käsiteanalyysi käyttäjien ja järjestelmäratkaisujen välillä	IT-hankintainvestoinnit
			kehittäjäkyselyjen / tietohallinto avoimet kysymykset	hankinta- ja integraatiokustannusten muutos ja suhde toteutettujen uusien piirteiden lukumäärään ja laajuuteen
			paljastavatko rajapinnat järjestelmien sisäisiä teknisiä yksityiskohtia	ylläpidon kustannukset / versiohallinnan kustannukset
			miten eri käyttötilanteet ja käytön edellytykset on kuvattu	sovelluksen uusien osien käyttöönoton ja kehittämisen kustannukset
			millä tasolla hallinta- ja ylläpitotehtävät ja vasteajat on määritetty	suoritusympäristöjen hankinta- ja ylläpito-kustannukset
			tukevatko käytettävät välineet ja alustat useita vaihtoehtoisia liikennöintiä ym. ratkaisuja	
			EUCS mm. järjestelmän tarkkuus, virheetömyys	

Taulukko 4.7. Sovelluspalvelujen ja rajapintojen teknisten seikkojen mittareita.

Mitattava kokonaisuus: tekniset	Lukumäärämittarit	Työmäärä- ja ajalliset mittarit	Laadulliset mittarit	Taloudelliset mittarit
	yhteistyökumppanien / järjestelmien määrä, joilla samat liitäntätekniikat käytössä	palvelujen, sovellusten ja verkon vastajat, latenssi	teknisten kehittämisperiaatteiden mukaisuus ratkaisun eri osissa (esim. WS-I)	transaktion tuotto (transaction revenue)
	teknisten virhetilanteiden lukumäärä (esim. virhelokista)	saatavuus		
	eri rajapintatekniikoiden lukumäärä	virheiden esiintyvyyshävyysfrekvenssi		
	teknisten järjestelmäadapterien lukumäärä	MTBF		
	viestien vastaanottajien (osoitteiden) lukumäärä	MTBSI		
	tiedon hävikki	MTTR		
	transaktioiden lukumäärä	jonotusaika (esim. viestit jonossa)		
	tiedon muutosten lukumäärä			
	palvelukutsujen lukumäärä			
	virheiden lukumäärä			

4.3.2 Esimerkki arviointimittareiden valinnasta: järjestelmäympäristön tehostunut hallinta ja ylläpito

Kun on valittu, minkä tavoitteiden toteutumista halutaan seurata, voidaan käydä läpi arviointimittarit ja valita ne, joilla pystytään suoraan tai välillisesti mittaamaan ko. tavoitetta. Tämän jälkeen on vielä järkevää karsia mahdollisia mittareita siten, että saadaan valittua avainmittarit, jotka eivät ole päällekkäisiä ja jotka mittaavat mahdollisimman suoraan (muiden tekijöiden vaikutukset minimoiden) haluttua tavoitetta.

Alla esimerkki, jossa "Tehostunut hallinta ja ylläpito" -tavoitteeseen on koottu taulukossa olevia mittareita. : (Havainto: tällä päästään siihen, että on linkki tavoitteiden ja mittareiden välillä).

Lukumäärämittarit: Muutospyyntöjen ja muutosten lukumäärä, käyttäjien tekemien virheiden määrä, kuinka monessa paikassa säilytetään päällekkäisiä tietoja, kokonaisjärjestelmän osien ja sovelluspalvelujen lukumäärä ja koko, rajapintojen lukumäärä + yhden rajapinnan eri käyttötilanteiden määrä (tuotteessa tai ympäristössä), uudelleenkäytöstä ja integroinnista aiheutuvien virheiden lukumäärä, yhteistyökumppanien / järjestelmien määrä, joilla samat liitäntätekniikat käytössä, teknisten virhetilanteiden lukumäärä (esim. virhelokista).

Työmäärä ja ajalliset mittarit: ylläpidon työmäärä, kuinka moneen kertaan samoja tietoja (päällekkäisten rekisterien ylläpito, useat saman käyttäjän käyttäjätunnukset tms.) on syötettävä eri järjestelmiin ja kuinka kauan siihen käytetään aikaa, MTBF, MTBSI, MTTR, järjestelmän / palvelun saatavuus, uusien osapuolien mukaan ottamisen nopeus prosesseissa, ilman järjestelmämuutoksia (keskitettyihin prosessimäärittelyihin) tehtävien muutosten osuus, testaukseen kuluva aika, käyttöönottoon ja paikalliseen sovittamiseen kuluva aika ja työmäärä, käyttökatkot (ja niiden eri luokien) määrä / saatavuus ym., palvelujen, sovellusten ja verkon vastajat, latenssi.

Laadulliset mittarit: ylläpidon kokemat työnkuvan ja -määrän muutokset, luotettavuus ja skaalattavuusmittarit, tietojärjestelmämuutosten määrä ja laatu, prosessien seurantatietojen saatavuus, ratkaisun eri osien riippuvuudet, tiedon eheys (luotettavuus, oikeellisuus, EUCS: ajantasaisuus), onko kutsusuhteet tiukasti määritelty (mitkä kriteerit), tukevatko käytettävät välineet ja alustat useita vaihtoehtoisia liikennöinti ym. ratkaisuja, kehittämisperiaatteiden saatavuus ja niiden noudattamisen seuranta, oletetaanko suoritusympäristössä yhteisiä teknisiä ratkaisuja, millä tasolla hallinta- ja ylläpitotehtävät ja vasteajat on määritelty.

Taloudelliset mittarit: ylläpidon kustannukset, IUM, CoD, sovelluksen uusien osien suunnittelun, kehittämisen ja käyttöönoton kustannukset, versiohallinnan kustannukset, uudelleenkäytöstä ja integroinnista aiheutuvat kustannukset ja säästöt, IT -hankintainvestoinnit, hankinta- ja integraatiokustannusten muutos ja suhde toteutettujen uusien piirteiden lukumäärään ja laajuuteen, ylläpidon kustannukset, sovelluksen uusien osien käyttöönoton ja kehittämisen kustannukset, suoritusympäristöjen hankinta- ja ylläpito-kustannukset.

Edellä kuvatusta listasta avainmittareiksi voidaan valita (organisaation / hankkeen tavoitteista ja mittausmahdollisuuksista riippuen) esimerkiksi:

- kuinka monessa paikassa säilytetään päällekkäisiä tietoja (↓), kokonaisjärjestelmän osien ja sovelluspalvelujen lukumäärä (↑), muutospyyntöjen ja muutosten lukumäärä, teknisten virhetilanteiden lukumäärä (↓)
- MTTR (↓), kunkin sovelluspalvelun saatavuus (↑), palvelujen vasteajat (↓), ilman järjestelmämuutoksia (keskitettyihin prosessimäärittelyihin) tehtävien muutosten osuus (↑)
- ylläpidon kokemat työnkuvan ja -määrän muutokset, tiedon eheys (↑), prosessien seurantatietojen saatavuus (↑)
- ylläpidon kustannukset ja niiden muuttuminen (↓), IUM (↓), hankinta- ja integraatiokustannusten kehitys (↓).

Yllä mainittuihin mittareihin liittyvät kehityksen tavoitesuunnat on merkitty mittareiden yhteyteen. On kuitenkin huomattava, että mittareiden kuvaamien seikkojen kehittymistä tähän suuntaan ei tavoitella loputtomiin, vaan on määriteltävä tavoitetaso, jolle kunkin mittarin suhteen pyritään. Valittujen mittarien avulla voidaan seurata toisaalta ylläpito- ja hallintatyön "tehoa", toisaalta myös palveluarkkitehtuurin kehittymistä ja toimivuutta.

4.4 Arviointiprosessin vaiheet

Arviointiprosessin vaiheet voivat olla erilaisia riippuen siitä, mitä luvun 2 arviointitapaa käytetään. Eroja voi syntyä esim. siitä, onko kyseessä valintatilanne, arvioidaanko, missä määrin jotain saavutetaan vai käytetäänkö ennen/jälkeen tutkimusta. Seuraavassa on esitetty yleisen arviointiprosessin vaiheet (mukailtu Cronholm & Goldkuhl, 2003).

1) Tavoitteiden valinta ja arvioinnin suunnittelu

Aluksi päätetään arvioinnin tavoitteet. Arvioinnin tavoitteena voi olla tukea esim. tietojärjestelmän hankkimista, työtoiminnan kehittämistä tai nykytilan selvittämistä. Sen jälkeen valitaan arviointikriteerit, joiden mukaan arviointi tehdään. Lisäksi päätetään arvioinnin laajuus ja yksityiskohtaisuus, arviointiin käytettävä aika sekä resurssit (esim. kuka/etkät arvioivat). Lisäksi määrätään arvioinnin

vastuuhenkilöt sekä kenelle arvioinnin tulokset luovutetaan. Tässä vaiheessa tehdään myös varsinaiset etukäteisvalmistelut arviointia varten, esim. varmistetaan arvioijien tietämys kontekstista.

2) Mittareiden valinta

Tässä vaiheessa valitaan arviointikriteereihin ja arviointinäkökulmaan pohjautuvat mittarit. Mittarit voivat olla sekä laadullisia (esim. avoimet kysymykset) että määrällisiä (esim. virhemäärä, kustannukset, kulunut aika). Mittareiden valinnassa kannattaa kiinnittää huomiota myös siihen, kuinka varmoja tulokset ovat, kuinka kalliiksi mittauksen järjestäminen tulee sekä kuinka paljon tietyn mitaustuloksen saamiseen menee aikaa.

3) Tiedon keruu

Tiedon keruu voidaan tehdä monella tavalla kerättävän tiedon laadusta riippuen (joko osana toimintaa tai sen ulkopuolella). Voidaan esim. kokeilla/käyttää tarkasteltavaa järjestelmää, tutustua järjestelmädokumentteihin (esim. käyttöohjeet, järjestelmäkuvaukset) ja lokeihin, sekä havainnoida tai haastatella käyttäjiä, ylläpitäjiä ja kehittäjiä. Myös erilaiset ryhmäpohjaiset tapaamiset (esim. työpaikat, kokoukset) sekä kyselyt ovat yleisesti käytettyjä tiedonkeruumenetelmiä.

4) Tiedon analysointi

Arvioinnista saatu tieto analysoidaan tilanteesta ja mittareista riippuen joko määrällisesti tai laadullisesti. Ennen analysointia kannattaa kuitenkin tehdä tulosten järkevyydestä tarkastelu, jossa arvioidaan voivatko saadut mitaustulokset pitää ylipäänsä paikkaansa. Määrällisessä analysoinnissa tarkastellaan mitattavia ominaisuuksia ja analysoidaan niitä tilastollisia menetelmiä käyttäen. Laadullisessa analysoinnissa tavoitteena on todellisen elämän kuvaaminen, jonkin ilmiön seikkaperäinen kuvailminen, tai jonkin asian ymmärrettäväksi tekeminen. Voidaan esim. tarkastella käyttäjien vaatimuksia/mielityksiä jonkin tietyn ominaisuuden suhteen ja verrata niitä arvioinnin tuloksiin eli kuinka hyvin ominaisuus vastaa käyttäjien vaatimuksiin.

5) Johtopäätösten teko

Arviointiprosessin lopuksi tehdään tulosten pohjalta johtopäätökset. Johtopäätöksissä koostetaan havaitut ongelmat ja löydetty vahvuudet. Lisäksi huomattavimmat ongelmat priorisoidaan. Johtopäätösten tekemisessä pitää huomioida, kuinka yleistettäviä tulokset ovat, vai voidaanko tuloksia hyödyntää ainoastaan tässä tilanteessa. Tuloksia kannattaa tarkastella myös käytännöllisyyden kannalta (esim. tulos voi olla tilastollisesti merkittävä, mutta sillä ei ole merkitystä käytännössä). Arvioinnin tuloksena voi myös paljastua, että tarvitaan tarkennuksia tavoitteisiin tai käytettyihin arviointimenetelmiin, tai että tarvitaan lisätietojen keräämistä.

Tämän yleisen arviointiprosessin lisäksi on kehitetty tarkempia arviointiprosesseja erityyppisiin arviointeihin. Esimerkiksi Software Acquisition Capability Maturity Model (SA-CMM) (Cooper & Fisher, 2002) sisältää tietojärjestelmien hankintaan liittyvän arviointiprosessin kuvauksen. Mallin mukaan arviointi alkaa tuotevaatimusten kehittämisellä ja loppuu, kun hankintaprosessi on saatu päätökseen. Sen tavoitteena on kehittää arviointikriteerit yhdessä hankintasopimukseen liittyvien vaatimusten kanssa ja ylläpitää niitä koko hankintaprosessin ajan. Arvioinnin tuloksena saadaan objektiivinen näkemys hankintapäätöksen tekemiseen.

4.5 Yhteenveto hyötyjen arvioinnista

Tietojärjestelmien ja niihin liittyvien hankkeiden arviointiin on runsaasti erilaisia lähestymistapoja ja malleja. Arvioinnissa on erityisen tärkeää määritellä, mitkä ovat arvioinnin tavoitteet ja mitä seikkoja arvioinnissa halutaan mitata tai selvittää.

Aiemmin eri tutkimuksissa on arvioitu tietojärjestelmien havaittuja hyötyjä, yleistä hyväksyntää/leviämistä, virheettömyyttä, suhteellista hyötyä, mutta jatkossa on myös tulospainotteinen tutkimus tarpeen. Tietojärjestelmien arviointia tekevilla tutkijoilla on kiinnostus lisääntynyt tutkia myös informaatioteknologian vaikutuksia: kilpailuetua/hyötyä, organisationaalisia strategioita, tietojärjestelmien hyödyntämisaikaa, keskitason hallintoa ja teollisuuden/tuotannon tasoa (Tordzadeh, Doll 1999).

Tietojärjestelmien arkkitehtuuriin liittyvien seikkojen arviointi ei ole yleisimmin eri arviointihankkeissa esiin nostettujen seikkojen joukossa. SOA -lähestymistavalla ja web-sovelluspalvelutekniikoilla tavoiteltuja hyötyjä voidaan kuitenkin arvioida soveltaen erilaisia arviointimalleja, joita on esitetty kirjallisuudessa ja eri hankkeissa. Lisäksi tavoiteltujen hyötyjen arvioimiseksi voidaan tunnistaa monia suureita ja mittareita, joilla voidaan toisaalta seurata organisaation tietojärjestelmäarkkitehtuurin ja kokonaisuuden "palveluarkkitehtuurimaisuutta", toisaalta palvelupohjaisella lähestymistavalla tavoiteltuja yleisiä tai paikallisia tavoitteita.

Hyvin monissa tietojärjestelmien arvioinneissa ja arviointimalleissa korostuu loppukäyttäjän näkökulma, mahdollisesti johtuen siitä, että käyttöliittymien ja käytettävyyden muodostumista voidaan havainnoida ja liittää helposti käyttäjien mielipiteisiin. Monet arkkitehtuurilliset ja palveluarkkitehtuurilla tavoitellut seikat ovat kuitenkin vain epäsuorasti käyttäjälle näkyviä ja kohdistuvat ensisijaisesti tietohallintoon ja edellytysten luomiseen varsinaisille toiminnallisille sovelluksille. Tämä hankaloittaa tai estää monien arviointimallien käyttöä palveluarkkitehtuurilla tavoiteltujen hyötyjen arvioinnissa. Samasta syystä arvioinnissa on aina tekijöitä, jotka eivät ole johdettavissa arkkitehtuurillisiin ratkaisuihin: arkkitehtuuri tukee sitä, MITEN ratkaisuja toteutetaan, ja suurin osa käyttäjien kokemuksista liittyy kuitenkin siihen, MITÄ sisältöä eli tietoja, toimintoja ja suoraan käyttäjälle näkyviä ominaisuuksia sovelluksista löytyy. Käyttöliittymien suuri merkitys käytön miellyttävyydessä ei myöskään ole suoraan johdettavissa palveluarkkitehtuurin tai web-sovelluspalvelujen tarjoamiin mahdollisuuksiin. Samat rajoitteet koskevat tietenkin kaikkia tietojärjestelmä- ja arkkitehtuuriratkaisuja, jotka eivät ole suoraan näkyvissä loppukäyttäjälle. Lisäksi esimerkiksi järjestelmien käytettävyyteen liittyvät seikat ovat muuttuneet joidenkin mittaristojen kehittymisen jälkeen, kuten EUCS -mittaristossa (Doll ym. 1988).

Toinen ongelmallinen tekijä on monien tietojärjestelmien arviointiin tarkoitettujen mittareiden ja lähestymistapojen epämääräisyys ja usein lähes täysin laadullinen luonne. Kun mitattavana ei ole suoraan käyttäjien kokema subjektiivinen hyöty tai miellyttävyys, tarvitaan suhteellisen tarkasti määriteltyjä ja määrällisiä mittareita. Tällaisia mittareita vaaditaan yhä enemmän myös tietojärjestelmäkehityshankkeilta ja -lähestymistavoilta. Esimerkiksi tässä luvussa pohjana käytetyt DeLonen ja McLeanin arviointimallit eivät sovellu lainkaan taloudellisten hyötyjen arviointiin, eivätkä kiinnitä juurikaan huomiota tuotannollisten tai teollisten vaikutusten arviointiin, ja niiden sisältämien näkökulmien linkittäminen tiettyihin tavoiteltuihin hyötyihin tai yksittäisiin mittareihin tai suureisiin on vaikeaa. Tämä johtuu pääosin edellä kuvatusta ristiriidasta yleisluontoisten "laatuojen" ja toisenlaisen yleistyksen kautta saatujen tavoitteiden (lähestymistavalla yleisesti tavoitellut hyödyt) välillä. Malleja palvelupohjaisten ratkaisujen taloudellisten hyötyjen ja kustannusvaikuttavuuden arviointiin on kuvattu taloudelliseen arviointiin liittyvissä luvuissa .

Käyttäjänäkökulman ja vaikeasti vertailtavien sanallisten kuvausten lisäksi tietojärjestelmien saataavuus näyttää ylikorostuneelta monissa mittaristoissa, esimerkiksi ITIL -mittareissa sitä käsitellään monien eri näkökulmien kautta. Näyttääkin siltä, että loppukäyttäjän arviointien sijaan erityisen hyödyllisiä tässä käsitellyn arvioinnin tyyppisissä tapauksissa ovat johdon ja tietohallinnon sekä organisaation toiminnan näkökulmasta tehtävät arvioinnit ja niihin soveltuvat mittarit, joissa tieteenkin voidaan myös huomioida eri käyttäjäryhmiä, kumppaneita ja toimittajia. Lisäksi strategisimmat palvelupohjaisuuden tavoitteet kuten nopea reagointi muutospyyntöihin ja uusiin tilanteisiin saavutetaan tyyppillisesti vasta suhteellisen pitkälle kehitetyssä palvelupohjaisessa lähestymistavassa, ja kehityksen alkuvaiheessa tarvitaan "lähempänä kehitystyötä" olevia mittareita ja menetelmiä. Ylläpidon luonne muuttuu yleensä hitaasti, järjestelmien modulaarisuuden ja muutosten "lähellä tekemisen" kautta, joten näiden toimintatapojen mittaaminen auttaa määrittelemään, onko kehityssuunta johtamassa kokonaisuuden joustavuuden paranemiseen.

On järkevää yhdistää erityyppisiä mittareita tiettyjen tavoitteiden seurantaan, mutta on myös tärkeää pyrkiä rajaamaan käytettävät avainmittarit riittävän suppeaksi joukoksi, jossa on mahdollisimman vähän päällekkäisyyksiä ja välillisesti mitattavia seikkoja. Kunkin mittarin "hyvä" taso ja tavoiteltu muutossuunta on syytä määritellä ja tarkistaa aika ajoin. Kun tietty tavoitetaso on saavutettu, voidaan myös määritellä poikkeusten hallintaa. Mittaus- ja arviointimenetelmät vaikuttavat luonnollisesti siihen, kuinka usein ja toistuvasti arviointia tehdään. Erityisesti on pyrittävä erottamaan kehitystyön aikainen ja siihen liittyvä arviointi toistuvasta "tuotannon" seurannasta: kehitysarvioinnissa käytettävien tietojen seurannan jatkaminen varsinaisen kehityksen päätyttyä ei useinkaan ole järkevää. Jos halutut tavoitteet saadaan määriteltyä mitattavasti, voidaan myös niiden toteutumisessa siirtyä suuntaan, jossa esim. palveluarkkitehtuurissa voidaan saada ajantasaista tietoa toiminnasta. Tämä voi myös mahdollistaa välittömien analyysien tekemisen seikoista, joita ennen ei voinut suoraan mitata lainkaan, tai joiden arviointi olisi vaatinut esim. työlään manuaalisen tiedonkeruun tai erillisen arviointitutkimuksen.

On siis mahdollista käyttää yleisiä tietojärjestelmäarvioinnin malleja palvelupohjaisten ratkaisujen arviointiin. Mallien valinta sekä konkretisointi paikallisiin tavoitteisiin ja lähtökohtiin perustuen on kuitenkin erittäin tärkeää. Arvioinnissa on:

- ensin valittava tavoitteet (luku 2), joita kehittämisessä tavoitellaan,
- valittava arviointitapa (luku 4.1), joka soveltuu kehitystyön sekä tavoitteiden arviointiin,
- kehitettävä / valittava mittaristo ja suureet (luku 4.3.1), jotka sopivat valittuihin tavoitteisiin ja arviointitapaan,
- asetettava tavoitetasot valittujen mittareiden suhteen, koottava tieto valittuja mittareita käyttäen, ja käytettävä kerättyä tietoa toiminnan parantamiseksi ja ohjaamiseksi (luku 4.4).

Tässä luvussa kuvatut mallit ja esimerkit antavat pohjaa monille näistä toimenpiteistä. Eri tavoitteiden painottuminen käytännön kehittämisessä ohjaa menetelmien ja mittareiden valintaa tapauskohtaisesti.

5 Web Services ja SOA: taloudellinen arviointi

Tässä luvussa käsitellään Web Services-palvelujen käyttöönotosta sekä palvelupohjaiseen arkkitehtuuriin (Service Oriented Architecture, SOA) siirtymisestä aiheutuvia kustannuksia ja tavoiteltuja hyötyjä. Lisäksi kuvataan, miten web-sovelluspalvelujen ja palvelupohjaisen arkkitehtuurin kustannuksia ja hyötyjä voidaan arvioida.

Luvun alussa käsitellään mittareita, joita voidaan käyttää arvioitaessa Web Service ja SOA investointien kannattavuutta sekä minkälaisia tunnuslukuja niille voidaan laskea. Kannattavuuden arvioimisessa on oleellista kustannusten ja hyötyjen arviointi, joten seuraavaksi luvussa selvitetään, mistä kaikista tekijöistä aiheutuu kustannuksia ja hyötyjä web-sovelluspalvelujen ja SOA:n käyttöönotossa ja käytössä. Kustannuksia käsitellessä käydään ensin läpi palvelukeskeisen arkkitehtuurin taloudellinen malli sekä se, miten kustannusrakenne eroaa perinteisestä ohjelmistokehityksen kustannusten taloudellisesta mallista. Tämän jälkeen tarkastellaan tarkemmin mistä kulut ja kustannukset koostuvat. Palvelupohjaisesta arkkitehtuurista saatavia hyötyjä on kuvattu luvuissa 2.4 ja 2.5 arkkitehtuurin kypsyysasteiden näkökulmasta. Lopuksi käsitellään ohjelmistosuunnittelun attribuutteja palvelupohjaisessa arkkitehtuurissa sekä eri kehitysstrategioiden vaikutusta kustannuksiin ja hyötyihin. Ohjelmistosuunnittelun attribuuteilla on pääasiassa vaikutusta ohjelmistotuotantoon ja siten niiden suorat kustannus- ja hyötyvaikutukset koskevat lähinnä ohjelmistotuottajia, ja vain välillisesti asiakasorganisaatioita.

5.1 Web Services ja SOA: kannattavuuden arvioinnin mittarit

Web-sovelluspalvelujen ja palvelupohjaisen arkkitehtuurin investointien kannattavuutta voidaan arvioida yleisillä investoinnin kannattavuusmittareilla. Tässä esitellään yleisiä kannattavuusmittareita, joita voidaan käyttää Web Service ja SOA -investointien kannattavuuden arviointiin. Lisäksi esitellään joitain apukäsitteitä, joita kannattavuuslaskelmissa tarvitaan. Tässä käsiteltävät kannattavuusmittarit ovat kassavirta-analyysin (cash flow analysis) nettonykyarvo (Net Present Value, NPV) ja sisäinen korkokanta (Internal Rate of Return, IRR), takaisinmaksuaika (Payback Period) sekä sijoitetun pääoman tuotto (Return on Investment, ROI).

5.1.1 Kassavirta-analyysi

Kassavirta-analyysissä (cash flow analysis) lasketaan oletettujen kassavirtojen suhde käytettyihin investointeihin. Analyysi sopii hyvin pitkän aikavälin IT-projekteilte ja soveltuu siten hyvin myös käytettäväksi SOA:n ja Web Servicen yhteydessä. Kassavirta-analyysissä määritetään asianmukaista diskonttokorkoa (Discount Rate) käyttämällä nettonykyarvo ja/tai sisäinen korkokanta (Samtani ym. 2002). Tulevien kassavirtojen arvioinnissa tulee ottaa huomioon minkälaisia kuluja ja kustannuksia ja mitä hyötyjä investoinnilla saavutetaan. Palvelupohjaisen arkkitehtuurista ja web-sovelluspalveluista aiheutuvia kuluja ja kustannuksia sekä hyötyjä käsitellään tarkemmin luvussa 5.2. Investointiin liittyvien kulujen ja kustannusten sekä investoinnilla saavutettavien hyötyjen toteutumiseen liittyvät riskitekijät huomioidaan kassavirta-analyysin diskonttokorossa.

5.1.1.1 Diskonttokorko

Diskonttokorolla (diskonttauskorko) (discount rate) tarkoitetaan laskentakorkoa (tuottovaatimusta), jonka avulla tulevaisuuden kassavirrat voidaan muuttaa nykyarvoksi (Present Value, PV). Diskont-

tokorko määräytyy yleisen korkotason (interest rate), inflaation ja projektin tuottojen epävarmuustekijöiden mukaan (Rissa 2004). Diskonttokorko määrittelee halutun tuottovaatimuksen. Investoinnista aiheutuvien kulujen ja kustannusten sekä etenkin investoinnista saatavien hyötyjen toteutumiseen liittyy kuitenkin epävarmuustekijöitä. Nämä epävarmuustekijät aiheuttavat riskejä investoinnille, jonka vuoksi investoinnille tulisikin vaatia suurempaa tuottoa. Tämä tuottovaatimus voidaan ottaa huomioon käyttämällä suurempaa diskonttokorkoa arvioitaessa investoinnin kannattavuutta. Tämän vuoksi kassavirtoihin liittyvät epävarmuustekijät voivat vaikuttaa käytettävään diskonttokorkoon huomattavasti. Mitä suuremmat riskit ovat, sitä suuremmat ovat tuottovaatimukset ja sitä suurempi diskonttokorko.

5.1.1.2 Nykyarvo

Nykyarvon (Present Value, PV) laskemisessa investoinnista aiheutuvat nettokassavirrat (kassaan- ja kassastamaksut) siirretään diskonttokorkoa käyttäen haluttuun laskentahetkeen (KTI). Näin laskettu nykyarvo ilmaisee tulevaisuuden kassavirtojen arvon laskentapäivän arvona (Rissa 2004). Nykyarvossa ei siis oteta huomioon investointikuluja, vaan ne otetaan huomioon vasta laskettaessa nettonykyarvo. Luvussa 5.2 käsitellään mistä web sovelluspalvelujen ja palvelupohjaisen arkkitehtuurin kulut ja kustannukset, hyödyt sekä riskit muodostuvat. Investoinnista aiheutuvat kulut ja kustannukset aiheuttavat kassastamaksuja. Investoinnilla saavutettavilla hyödyillä saavutetaan kassaanmaksuja ja toisaalta niillä voidaan pienentää kassastamaksuja (saavuttaa säästöjä). Kuitenkin kulujen, kustannusten ja etenkin saavutettavien hyötyjen oikein arvioiminen voi olla vaikeaa. Tästä johtuen investoinnille vaadittava tuottovaatimus tulisi olla suurempi kuin yleisen korkokannan ja inflaation yhteisvaikutus. Tämä isompi tuottovaatimus huomioidaan laskuissa suurempana diskonttokorkona. Nykyarvo saadaan laskettua seuraavasta kaavasta:

$$\text{nykyarvo} = [1. \text{ vuoden nettokassavirta} / (1 + \text{diskonttokorko})] + [2. \text{ vuoden nettokassavirta} / (1 + \text{diskonttokorko})^2] + \dots + [\text{vuoden N nettokassavirta} / (1 + \text{diskonttokorko})^N]$$

5.1.1.3 Esimerkki nykyarvon laskemisesta

Otetaan esimerkki nykyarvon laskemisesta, jonka tarkoituksena on esittää kuinka laskentakaavaa käytetään. Laskuesimerkissä olevat investoinnista saatavat nettohyödyt eivät perustu mihinkään todellisiin saavuttaviin nettohyötyihin. Käytettävät nettohyödyt ovat täysin keksittyjä ja niiden ainoa tarkoitus on esittää kuinka laskentakaavaa käytetään. Luvussa 2.4 käsitellään, minkälaisia hyötyjä saavutetaan palvelupohjaisen arkkitehtuurin eri kehitysvaiheissa ja luvussa 2.5 eritellään tarkemmin, mitä toiminnallisia, sovelluskehityksen ja teknisiä hyötyjä SOA:lla voidaan saavuttaa. Laskussa käytettävää diskonttokorkoa sen sijaan voidaan pitää IT-projekteissa suuntaa antavana. Diskonttokorkossa otetaan kuitenkin huomioon investointiin liittyvät riskit, joten sen suuruus voi vaihdella tapauskohtaisesti. Myös yleinen korkokanta ja inflaatio vaikuttavat diskonttokoron suuruuteen.

Oletetaan että laskentahetken yleisen korkokannan ja sen hetkisen inflaation mukainen diskonttokorko on 4 %. Organisaatio on investoimassa palvelupohjaiseen arkkitehtuuriin ja web-sovelluspalveluihin. Investoinnin epävarmuustekijöiden vuoksi sijoitukselle vaaditaan kuitenkin korkeampaa tuottovaatimusta ja siksi laskussa käytettävä diskonttokorko on 10 %. Oletetaan että ensimmäisenä vuonna nettohyötyä saadaan 20 000 €, toisena vuonna 50 000 € ja kolmantena vuonna 70 000 €. Tästä saadaan laskettua kolmen ensimmäisen vuoden nykyarvo:

$$\text{nykyarvo} = [20\,000 \text{ €}(1 + 0.1)] + [50\,000 \text{ €}(1 + 0.1)^2] + [70\,000 \text{ €}(1 + 0.1)^3] = 18182 + 41322 + 52592 = 112\,096 \text{ €}$$

5.1.1.4 Nettonykyarvo

Nettonykyarvo (Net Present Value, NPV) saadaan kun nykyarvosta vähennetään alkuperäiset investoinnit (KTI). Nettonykyarvo osoittaa saatavan euromääräisen tuoton lasketulta aikaväliltä. Jos nettonykyarvo on positiivinen lasketulla aikavälillä, silloin projektin nykyarvo saavuttaa sen vaaditut kassakulut ja hankkeeseen kannattaa ryhtyä. Nettonykyarvo siis saadaan kaavasta:

nettonykyarvo = nykyarvo – alkuperäinen investointi

5.1.1.5 Sisäinen korkokanta

Sisäinen korkokanta (Internal Rate of Return, IRR) määrittelee diskonttokoron, jolla nettonykyarvo saa arvon nolla (Samtani ym. 2002). Sisäinen korko toisin sanoen ilmaisee investoinnin tuoton saatuna (tai menetettynä) korkona. Mikäli sisäinen korkokanta on suurempi kuin hankkeen pääomakustannukset, niin tällöin kassatuotot kattavat hankkeen poistot ja diskonttokoron. Investointi on sitä kannattavampi, mitä isompi sisäinen korkokanta on. Aiemmin luvussa 3 on käsitelty, minkälaisia pääomakustannuksia palvelupohjaisesta arkkitehtuurista ja web-sovelluspalveluista aiheutuu. Sisäinen korkokanta saadaan ratkaistua kaavasta:

1. vuoden nettohyöty / (1+IRR) + 2. vuoden nettohyöty / (1+IRR)² + ... + vuoden N nettohyöty / (1+IRR)^N = investointi

Sisäinen korkokanta kannattaa laskea taulukkolaskentaohjelmalla, jonka avulla laskenta onnistuu vaivattomasti. Esim. Excel-tilukkolaskentaohjelmasta löytyy IRR-funktio (suomenkielisessä SI-SAINEN KORKKO) sisäisen korkokannan laskemiseksi.

Mikäli yritys investoisi palvelupohjaiseen arkkitehtuuriin 100 000€ ja arvioitaisiin, että se tuottaa hyötyä ensimmäisenä vuotena 20 000€, toisena 50 000€ ja kolmantena vuotena 70 000€, niin tällöin kolmen ensimmäisen vuoden sisäiseksi korkokannaksi saataisiin 16 %.

5.1.2 Takaisinmaksuaika-analyysi

Takaisinmaksuaika-analyysissä (Payback Period Analysis) lasketaan, missä ajassa investointi maksaa itsensä takaisin ja alkaa tuottaa tulosta (Samtani ym. 2002). Takaisinmaksuajan siis tulisi olla lyhyempi kuin investoinnin oletettu käyttöikä, että investointi olisi taloudellisesti kannattava. Takaisinmaksuaika lasketaan jakamalla investoinnit vuotuisella nettohyödyllä. Palvelupohjaisen arkkitehtuurin ja web-sovelluspalvelujen tapauksessa hyödyt kuitenkin jakaantuvat usean vuoden ajanjaksolle ja siksi hyödyt tulee esittää nykyarvossa. Nettokassavirta ei kuitenkaan tavallisesti ole tasaista pitkällä aikavälillä ja tästä johtuen takaisinmaksuaika joudutaan laskemaan kumulatiivisesti. Takaisinmaksuajan laskeminen selvitetään seuraavan esimerkin avulla.

5.1.2.1 Esimerkki takaisinmaksuajan määrittämisestä

Oletetaan, että organisaatio investoi palvelupohjaiseen arkkitehtuuriin liittyviin palveluihin 100 000 € ja on arvioitu että hyötyä saadaan ensimmäisenä vuonna 20 000 € toisena vuonna 50 000 € ja kolmantena vuonna 70 000 €. Diskonttauskoroksi on määritelty 10 % yleisen korkotason, inflaation ja investointiin liittyvien epävarmuustekijöiden perusteella. Tästä saadaan laskettua takaisinmaksuaika seuraavasti:

investointiajankohta: (investointikulut) -100 000 € => kumulatiivinen tulos = -100 000 €

1. vuoden kumulatiivinen tulos: $[20\,000\text{ €} / (1 + 0.1)] - 100\,000\text{ €} = -81\,818\text{ €}$

2. vuoden kumulatiivinen tulos: $[50\,000\text{ €} / (1 + 0.1)^2] - 81\,818\text{ €} = -40\,496\text{ €}$

3. vuoden kumulatiivinen tulos: $[70\,000\text{ €} / (1 + 0.1)^3] - 40\,496\text{ €} = 12\,096\text{ €}$

Koska kolmannen vuoden kumulatiivinen tulos on positiivinen se tarkoittaa sitä, että investointi on maksaa itsensä takaisin ennen kolmannen vuoden päättymistä. Nyt jos oletetaan, että kolmannen vuoden kustannukset jakautuvat tasaisesti kyseiselle vuodelle, niin saadaan määriteltyä, missä vaiheessa vuotta investointi maksaa itsensä takaisin. Jos kolmannen vuoden kustannukset Web Servicestä on ollut 20 000 €, niin tällöin kolmannen vuoden bruttohyöty ennen kustannuksia on 90 000 € (3. vuoden nettohyöty + kustannukset). Kun toisen vuoden kumulatiiviseen tulokseen lisätään kolmannen vuoden kulut ja tämä jaetaan kolmannen vuoden bruttohyödyllä ja edelleen tämä osamäärä kerrotaan kahdellatoista, niin saadaan määriteltyä missä vaiheessa vuotta investointi maksaa itsensä takaisin:

$$[(40\,496\text{ €} + 20\,000\text{ €}) / 90\,000\text{ €}] \times 12\text{ kk} = 8\text{ kk}$$

Joten tässä tapauksessa Web Service -investoinnin takaisinmaksuaika olisi 2 vuotta ja 8 kuukautta.

5.1.3 Sijoitetun pääoman tuotto (ROI) web-sovelluspalveluissa

5.1.3.1 Kaava web-sovelluspalvelujen pääoman tuoton määrittämiseksi

Sijoitetun pääoman tuotolla (Return on Investment, ROI) tarkoitetaan hankkeeseen investoitujen nettohyötyjen suhdetta aiheutuneisiin kustannuksiin. Mikäli kyseessä on lyhyen aikavälin Web Service investointi, saadaan sijoitetun pääoman tuotto laskettua kaavasta:

$$\text{ROI} = [\text{Rahalliset hyödyt (aineelliset ja aineettomat)} - \text{Web Service teknologiaan käytetyt kustannukset}] / \text{Web Service teknologiaan käytetyt kustannukset} \times 100$$

Pääoman tuotto voidaan laskea vain lyhyelle aikavälille käyttäen edellä mainittua kaavaa, jossa jaetaan projektin nettohyödyt projektin kustannuksilla. Web Service -investoinnit ovat kuitenkin usein pitkän aikavälin investointeja, jolloin kustannukset ja hyödyt jakautuvat usealle vuodelle. Mikäli tehdään pidemmän aikavälin investointeja, tulee pääoman tuoton oikein arvioimiseksi tulevaisuuden kassavirrat muuttaa laskentahetkeä vastaavaksi arvoksi. Kassavirtojen arvon muuttaminen on tärkeää siksi, että tämän päivän rahan arvo ei ole sama kuin sama rahamäärä esimerkiksi vuoden tai kahden kuluttua. Tämän vuoksi pääoman tuoton laskemisessa tulee nettohyödyt muuttaa nettonykyarvoksi, jolloin laskettava tuotto voidaan ilmaista laskentahetken arvona (Rissa 2004). Tällöin pitkän aikavälin investoinnin pääoman tuottoprosentin kaavaksi saadaan seuraava:

$$\text{ROI} = \{[1.\text{ vuoden nettohyöty} / (1 + \text{diskonttokorko}) + 2.\text{ vuoden nettohyöty} / (1 + \text{diskonttokorko})^2 + \dots \dots \dots + \text{vuoden N nettohyöty} / (1 + \text{diskonttokorko})^N]\} / \text{investointikustannukset} \times 100$$

Tällä laskentakaavalla laskettu ROI ilmaisee sijoitetun pääoman tuoton lasketulta aikaväliltä laskentahetken arvona. Pääoman tuotto voidaan laskea investoinnin oletetun käyttöiän mukaan. Mikäli sijoituksen tuotto jää alle 100 %, ei kyseinen investointi ole kannattava. Kaavassa olevalla nettohyödyllä tarkoitetaan kyseisen vuoden aikana investoinnista aiheutuvien hyötyjen ja kustannusten erotusta.

5.1.3.2 ROI esimerkki

Oletetaan että palvelupohjaiseen arkkitehtuuriin on sijoitettu 100 000 € ja arvioidut nettohyödyt ovat samat kuin edellisissä laskuissa eli 20 000 €, 50 000 € ja 70 000 €. Aikaisemman laskun mukaan nettohyötyjen nykyarvoksi saatiin 112 096 €. Kun tämä jaetaan investointikustannuksilla ja kerrotaan sadalla, niin saadaan sijoitetun pääoman tuotto kolmelle vuodelle.

$$\text{ROI} = (112\,096 \text{ €} / 100\,000 \text{ €}) \times 100 = 112 \%$$

Pääoman tuotto siis ilmaisee tuoton lasketulle aikavälille, tämän vuoksi erisuuruisille investoinneille tai eri aikajaksoille lasketut ROI:t eivät ole keskenään vertailukelpoisia. Jos esimerkiksi edellinen lasku olisi laskettu neljän vuoden ajanjaksolle ja oletettaisiin, että neljännen vuoden nettohyödyt olisivat 80 000 €, niin ROI olisi jo 192 %.

5.1.4 Kannattavuusmittareiden soveltuvuus

Nettonykyarvo on eräs käyttökelpoinen mittari arvioitaessa investoinnin kannattavuutta. Kassavirtojen tarkastelussa arvioidaan mitä kuluja ja kustannuksia investoinnista aiheutuu sekä mitä hyötyjä investoinnilla voidaan saavuttaa. Käyttämällä suurempaa tuottovaatimusta voidaan huomioda investointiin liittyvät riskitekijöitä. Suurempi tuottovaatimus näkyy laskuissa suurempana diskonttokorkona. Mikäli nettonykyarvo on investoinnin elinkaaren päätyttyä positiivinen tai nolla, niin silloin investointi on kannattava. Jos nettonykyarvo on nolla, niin siinä tapauksessa investointi on tuottanut laskennassa käytetyn koron sijoitetulle pääomalle eli laskennassa käytetty tuottovaatimus on saavutettu.

Sisäinen korkokanta ilmaisee kuinka suuri tuotto saavutetaan niillä oletuksilla, joilla laskeminen on tehty. Sisäisellä korkokannalla voidaan arvioida saavutetaanko investoinnille asetetut tuottovaatimukset. Mikäli sisäinen korkokanta tarkasteluajana on suurempi kuin investoinnin tuottovaatimus, niin silloin investointi on kannattava.

Sijoitetun pääoman tuotto (ROI) on yleisesti käytetty tapa mitata, investoinnin kannattavuutta. ROI ilmaisee prosenteissa, paljonko pääomalle kertyy tuottoa tarkasteltavalta aikaväliltä. Arvioitaessa kannattavuutta ROI:n perusteella tarvitaan usein muutakin tietoa. Koska ROI ilmaisee vain tuoton koko aikaväliltä, niin esim. eripituisien ajanjaksojen ja erisuuruisien investointien ROI:t eivät ole vertailukelpoisia. Lisäksi ROI:n käyttö terveydenhuollon tietotekniikan arvioinnissa on vaikeampaa kuin monilla muilla teollisuudenaloilla, koska IT toimii lähinnä välineenä hoidon sekä toiminnan parantamisessa, ja suuri osa toiminnasta on ihmisten välistä. Kaikkien tavoitteiden toteutuminen ei näin näy suoraan esim. kassavirroissa.

Myös takaisinmaksuaika on eräs hyödyllinen kannattavuusmittari, koska se kertoo millä ajankellällä investointi maksaa itsensä takaisin ja alkaa tuottaa tulosta.

Sisäinen korkokanta, takaisinmaksuaika ja sijoitetun pääoman tuotto ovat kaikki hyviä tunnuslukuja vertailtaessa erisuuruisia investointeja. Nettonykyarvoa sen sijaan voi käyttää vain samansuuruisien investointien vertailuun. Kannattavuuden mittaamisessa suurimmat ongelmat ovat kuitenkin siinä miten hyvin onnistutaan arvioimaan investoinnin kustannukset, hyödyt ja riskit. Etenkin aineettomien hyötyjen arvioiminen taloudellisina etuina voi olla joskus hankalaa. Tämän dokumentin muissa osioissa käsitellään, miten kustannuksia ja hyötyjä voidaan arvioida. Kun nämä tekijät on määriteltä, niin voidaan kannattavuutta tarkastella edellä mainituilla kannattavuusmittareilla. Web-

sovelluspalveluihin ja palveluarkkitehtuuriin investoimalla pyritään yleisesti esim. sähköisessä kaupankäynnissä saavuttamaan seuraavia tuloksia:

- Kasvaneet myyntituotot
- Alentuneet kustannukset
- Parantunut tehokkuus
- Korkeampi kannattavuus
- Lyhyempi takaisinmaksuaika
- Korkeampi sisäinen korkokanta (sisäinen korko)
- Vähäisemmät riskit

5.2 Kustannusten, hyötyjen ja riskien määrittäminen

Kun web-sovelluspalveluihin ja palvelupohjaiseen arkkitehtuuriin investoidaan, tulee ensiksi selvittää investoinnista aiheutuvat kulut ja kustannukset, investoinnista saatavat hyödyt sekä investointiin liittyvät riskit. Näiden selvitysten perusteella voidaan tehdä kannattavuuslaskelmat ja arvioida investoinnin kannattavuutta.

Kustannuksia määritettäessä on otettava huomioon palvelujen toteutusvaiheeseen liittyvät kustannustekijät sekä palvelujen aiheuttamat jatkuvat kustannukset. Näihin kustannuksiin sisältyvät ohjelmistot, laitteistot, järjestelmäintegraatio sekä tuotannon tukikulut. Laskemisessa ei kuitenkaan riitä teknisten asioiden tarkasteleminen, vaan lisäksi on otettava huomioon myös joukko erilaisia toimintaan liittyviä tekijöitä, jotka vaikuttavat sijoitettavan pääoman tuottoon (Samtani ym. 2002). Tällaisia tekijöitä on lueteltu luvussa 5.2.2.

Web-sovelluspalvelujen pääoman tuottoon vaikuttavat tekijät vaihtelevat paljon yrityksestä, sovelluksesta ja sen toteutuksesta riippuen. Pääoman tuottoa saadaan kasvavasta operatiivisen toiminnan tehostumisesta, jolla saavutetaan toimintaprosessien virtaviivaistumista ja automatisoitumista. Toisaalta pääoman tuottoa saadaan myös alenevista kustannuksista, joita tulee mm. sovelluksien kehitysajan lyhenemisestä ja paranevasta palvelujen uudelleenkäytöstä. Web-sovelluspalveluille voidaan kuitenkin määritellä melko hyvin pääoman tuotto käyttämällä luvussa kaksi esiteltyä web-sovelluspalvelujen ROI mallia. Tässä mallissa tulee arvioida web-sovelluspalveluilla saavutettavia hyötyjä sekä kuluja ja kustannuksia, joiden perusteella voidaan laskea tulevien vuosien nettohyödyt sekä niiden perusteella laskea tuotto sijoitetulle pääomalle. (Samtani ym. 2002)

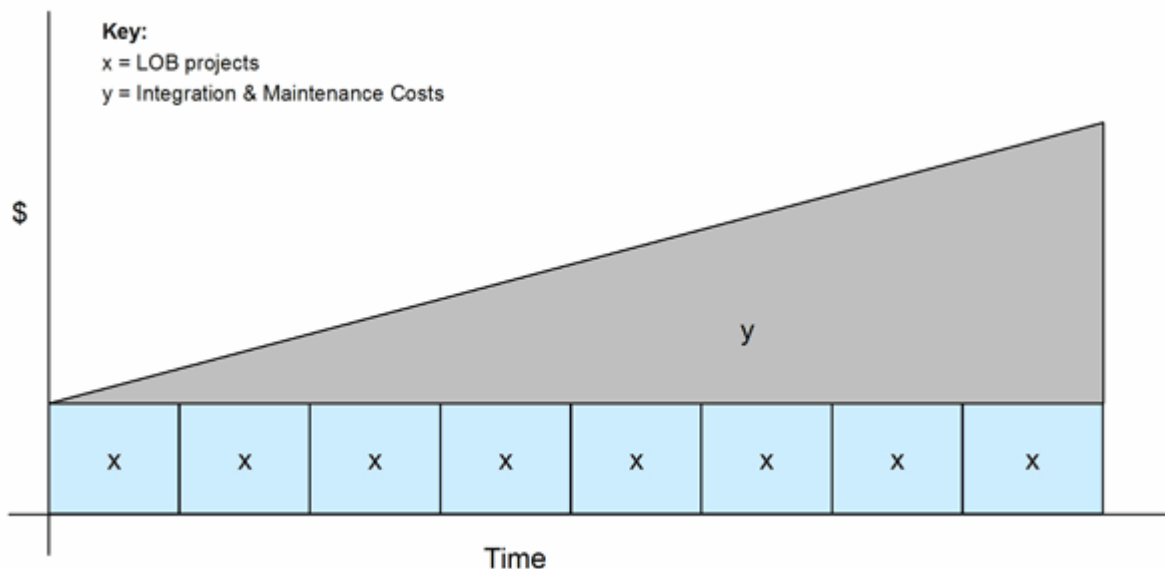
Seuraavassa käsitellään palvelupohjaisen arkkitehtuurin kustannusten taloudellista mallia ja selvitetään miten SOA:n kustannusrakenne eroaa perinteisen ohjelmistokehityksen kustannusrakenteesta. Tämän jälkeen kuvataan, mistä web-sovelluspalvelujen kulut ja kustannukset koostuvat. Kulujen ja kustannusten jälkeen käydään läpi palveluarkkitehtuurin eri kypsyysvaiheet, mitä hyötyjä SOA:lla voidaan saavuttaa sekä missä kypsyysvaiheissa hyödyt realisoituvat. Kypsyysvaiheiden jälkeen käydään läpi mitä eri teknisiä, toiminnallisia ja sovellustuotannon hyötyjä Web Servicellä on mahdollista saavuttaa. Hyötyjä asiakasorganisaation ja ohjelmistotuottajan kannalta käsitellään tarkemmin luvussa "SOA ja Web Services - tavoitellut hyödyt". Luvussa 5.2.3 käsitellään web-

sovelluspalvelujen käyttöönotosta ja käytöstä aiheutuvia riskejä. Web-sovelluspalvelujen kokonais-kustannusten ja hyötyjen arvioinnista sekä niihin liittyvien riskien arvioinnista saatuja tunnuslukuja voidaan hyödyntää luvussa 5.1 esiteltyn investoinnin kannattavuusmittareiden laskemisessa ja siten investoinnin kannattavuuden arvioinnissa. (Samtani ym. 2002).

Luvussa 5.3 selvitetään tarkemmin, minkälaisia kustannus- ja hyötyvaikutuksia palvelupohjaisen arkkitehtuurin eri ominaisuuksilla saavutetaan. .

5.2.1 Kustannusten taloudellinen malli

Perinteisen ohjelmistokehityksen kustannusten taloudellinen malli perustuu eri liiketoiminta-alueiden (line of business) vaatimukseen, joita ohjaavat kysynnän ja tarjonnan lait sekä omistus- ja rahoitusmallit. Tällaiset toiminta-aluekohtaiset projektit ja niiden erillisjärjestelmät eivät ole kuitenkaan suunniteltu alun perin integroitumaan toisten projektien ja järjestelmien kanssa ja tämän vuoksi integrointikustannukset kasvavat nopeasti integroituvien projektien lisääntyessä. Siksi tämä taloudellinen malli ei sovellu palvelupohjaiselle arkkitehtuurille. Kuvassa 9 on esitetty toiminta-aluekohtainen malli yksinkertaistettuna kaaviona, jossa x:llä on merkitty liiketoiminta-alueiden projektit (LOB projects) ja y:llä integraatio- ja ylläpitokustannukset. (Sprott 2006)

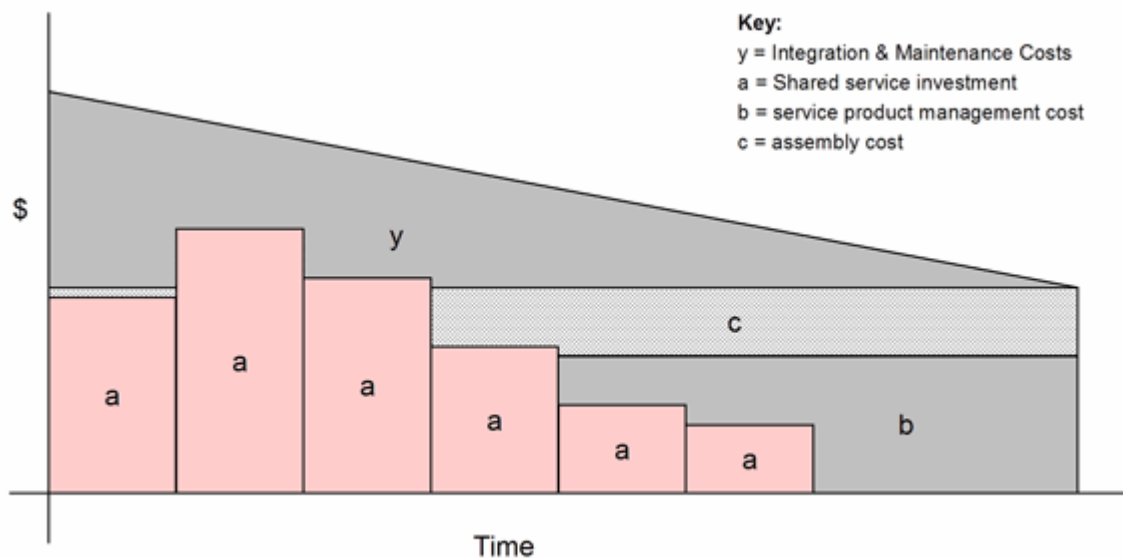


Kuva 9. Perinteisen ohjelmistokehityksen kustannusten taloudellinen malli.

Palvelupohjaiselle arkkitehtuurille sopiva malli on eräänlainen "valmista ja kokoa" -malli. SOA:ssa on pyrkimyksenä saada aikaan palvelukerros, jonka abstrakteja ja yleistettyjä palveluja voidaan käyttää eri tilanteissa ja eri tarkoituksiin. Näistä palveluista kokoamalla ja orkestroimalla saadaan aikaan halutut vaatimusten mukaiset toiminnot ja prosessit. Tämä mahdollistaa uusien vaatimusten mukaisten prosessien nopean koostamisen sekä standardipalvelujen uudelleenkäytön. (Sprott 2006)

Tällaisessa SOA-mallissa standardoituja palveluita hallitaan kuten ohjelmistotuotteita, tuottamalla ennustettujen vaatimusten mukaisia monen käyttäjän ohjelmistopalveluja ja hallitsemalla niiden eri piirteiden kehittämistä ja muutoksia. Näin ollen tuotehallinnan kustannukset jakautuvat laajalti kaikkien palveluiden käyttäjien kesken. Tässä mallissa päinvastoin kuin perinteisessä mallissa integraatio- ja ylläpitokustannukset alenevat asteittain palvelujen määrän lisääntyessä. Mitä enemmän organisaatio saa laajennettua tarvitsemiensa palvelujen kokoelmaa, sitä vähemmän se tarvitsee kah-

denvälistä integrointia. Kuvassa 10 on esitelty palvelupohjaisen arkkitehtuurin kustannusten taloudellinen malli. Mallissa kirjain y kuvaa alenevia integrointikustannuksia, a kuvaa jaettujen palvelujen investointikustannuksia, b kuvaa tuotehallinnan kustannuksia ja c kuvaa koostamisesta ja orkestroinnista aiheutuvia kustannuksia. (Spratt 2006)



Kuva 10. Palvelupohjaisen arkkitehtuurin kustannusten taloudellinen malli.

5.2.2 Kulut ja kustannukset

Palvelupohjaisen arkkitehtuurin ja web-sovelluspalvelujen kustannukset jakautuvat laajalti tietotekniikan ja organisaation toiminnan eri alueille. Seuraavassa listassa on esitelty kuluja ja kustannuksia, jotka tulee ottaa huomioon laskettaessa pääoman tuottoa Web Servicelle. Listassa on käsitelty kuluja ja kustannuksia soveltaen lähteitä (Samtani ym. 2002; Spratt ym. 2003):

- Laitteistovaatimukset ja -hankinnat
- Ohjelmistovaatimukset
 - Middleware-ohjelmistojen hankinnat ja päivitykset Web Service-protokollien tukemiseksi.
 - Sovelluspalvelujen hallintaohjelmistot sekä valvontatyökalut.
 - Organisaation omaan käyttöön tarkoitettu sovelluspalveluhakemisto (esim. yksityinen UDDI-rekisteri). Hakemisto-ohjelmistot ovat usein ilmaisia, mutta vaativat hallintatyötä ja voivat tarvita erillisen palvelimen.
 - Turvallisuusinfrastruktuuri, johon sisältyvät mm. yleiskäyttöiset turvallisuuspalvelut, käyttäjien ja sovellusten tunnistaminen, XML-palomuri, viestitason turvallisuuden toteuttava infrastruktuuri (esim. tarvittavat sertifikaatit) jne.

- Organisaation kattavan palveluarkkitehtuurin suunnittelu-, toteutus- ja ylläpitokustannukset, jos palveluiden oltava sovellettavissa koko organisaation laajuisesti.
- Palveluiden suunnittelu- ja toteuttamiskustannukset (palvelut pyritään suunnittelemaan siten, että niiden abstraktiotaso vastaa toimialan asiantuntijoiden ymmärtämiä prosesseja ja käsitteitä, eikä paljasta toteutuksen tai teknisiä yksityiskohtia).
- Palveluiden suunnitteluun ja hyväksymiseen liittyvät kustannukset sekä palveluiden tuottajien että niiden käyttäjien osalta.
- Ohjelmistosuunnittelun ja -kehityksen elinkaarikustannukset - toimialueeseen tutustuminen, vaatimusmäärittelyt, ratkaisumäärittelyt (toiminnallinen & tekninen), toteutuskustannukset, testauskustannukset.
- Käyttöönottokustannukset - ohjelmistoasennukset, asiakasorganisaation käyttöönotto-projekti ja toiminnan muutokset.
- Suunnittelijoiden ja kehittäjien kustannukset sekä projektinhallinta.
- Toiminnallisten (mukana olevien toimintojen, tietojen ja kokonaisuuksien = piirteiden hinnat) ja laatuvaatimusten (suorituskyky, käytettävyys, turvallisuus jne.) kustannukset.
- Koulutusvaatimukset
 - SOA-osaamisen hankinta ja kouluttaminen.
 - Web Service -teknologian opiskelun ja käyttöönoton kustannukset (optimitilanteessa näiden tekniikoiden käyttö on pitkälti automatisoitua ja piilossa kehittäjiltä).
- Organisaation toiminnan ja roolitusten muutokset.
- Sovelluspalvelujen kokoelman suunnittelu ja jatkuva hallinta.
- Tukipalvelujen investoinnit, esim. palvelujen ja sovellusten käyttötuki (päivystys, helpdesk, lähituki, etähallinta), teknisen infrastruktuurin ylläpito.
- Organisaation tarkan ja yksiselitteisen toimintapolitiikan muodostaminen ja hallinta, palveluihin liittyvän laatu järjestelmän kehitys.
- Verkon kaistanleveysvaatimukset ja tarvittavat uusimiset.
- Käyttökustannukset (ylläpitokustannukset, päivityskustannukset jne.)
- Toiminnan kehittämisen kustannukset.
 - Pitkän aikavälin palveluvaatimusten suunnittelu.
 - Palvelutuotteiden hallinta.
 - Sisäisten ja ulkoisten käsitteiden ja terminologioiden (semanttinen) kehittäminen ja yhtenäistäminen.

- Toimintaprosessien määrittely, kehittäminen ja yhtenäistäminen.
- Suunnittelu- ja mittausjärjestelmät, joilla voidaan mm. seurata muutoksiin reagoinnin nopeutta toiminnassa.
- Palvelujen hallinta- ja valvontajärjestelmät.
- Konsultointipalkkiot liittyen koulutuskustannuksiin, käyttöönottokustannuksiin, suunnittelukustannuksiin tai muihin yllä oleviin kustannuksiin.

Palvelupohjaisen arkkitehtuurin suunnittelun ja toteutuksen kustannusten arvioinnissa eri lähestymistavoilla voidaan hyödyntää ohjelmistosuunnittelun osatekijöitä, joita on käsitelty luvussa 3.

5.2.3 Riskinhallintaan vaikuttavia tekijöitä

Web-sovelluspalveluihin ja SOA:an liittyy myös epävarmuustekijöitä. Taloudellinen epävarmuus liittyy pitkälti siihen, miten hyvin kustannukset osataan arvioida ja etenkin siihen miten hyvin tavoitellut hyödyt pystytään saavuttamaan ja arvioimaan oikein. Palvelupohjaiseen arkkitehtuuriin investoiminen edellyttää pitkäaikaista sitoutumista arkkitehtuurin kehittämiseen, jotta sillä saavutettaisiin riittävästi taloudellisia hyötyjä ja että investointi olisi kannattava. Suurimmat mahdolliset hyödyt alentaen kustannuksia ja riskejä saavutetaan vasta edistyneen kulttuurillisen integraation vaiheessa. Omistussuhdeasiat ovat myös eräs riskejä aiheuttava tekijä, joka on otettava huomioon riskienhallinnassa.

Palvelukeskeisen arkkitehtuurin vaiheittaisesta kehittämisestä on hyötyä myös kehitystyön riskienhallinnassa. Kun järjestelmiä kehitetään asteittain, voidaan keskittyä kehittämään vain tiettyä osaa kerrallaan ja näin myös riskit kehitystyössä rajoittuvat pienemmälle alueelle.

Samtani ym. (2002) ovat esitelleet riskitekijöitä, jotka voidaan jaotella seuraaviin tekijöihin, jotka on yhdistetty arkkitehtuurikehityksen vaiheisiin ja hyötytekijöihin (ks. luku 2.4 ja 2.5).

- Uusi teknologia (I, A+H+Y)
- Standardien kypsyys (Standardit eivät kypsyä tai valmiita), (I-II, Y+A+H)
- Web Service työkalut ja palvelimet - vakaus, kehittyvyys jne. (II, H+A)
- Ulkoisten Web Service -palvelujen laatu, luottamus ja sopimukset (II-IV, Y+H+A)
- Turvallisuus (II, A+Y+H)

Nykyisin teknisimmät näistä riskeistä alkavat olla melko hyvin hallittavissa. Web Service -teknologia on ollut käytössä jonkin aikaa, ja perusteknologiaan kuuluvat Web Service standardit (WSDL, SOAP ja UDDI) ovat jo siten melko kypsiä. Näiden lisäksi on saatavilla WS-I:n (Web Services Interoperability Organization) profiileja standardien yhdenmukaiseen käyttöön ja tulkitaan (WS-I 2006a) sekä turvallisuuspiirteiden toteuttamiseen (WS-I 2006b). Web Service työkalut ovat myös kehittyneet, ja nykyisin ne mahdollistavat nopean ja suhteellisen helpon tavan toteuttaa yksinkertaisia Web Service -palveluja. Myös palvelimet tukevat hyvin Web Service teknologiaa, joten sekään ei ole nykyisin merkittävä riski.

5.3 Palvelupohjainen arkkitehtuuri (SOA) ja ROI

Palvelupohjaisen arkkitehtuurin kehittämiseen ja muihin IT-projekteihin liittyvät päätökset ovat usein ristiriidassa muiden organisaation toiminnassa ja johtamisessa tehtävien päätösten kanssa. Tämä johtuu siitä, että useimmat päätökset tehdään lyhyelle aikavälille, kun taas IT-projektien hyödyt saavutetaan pitkällä aikavälillä. Tästä johtuen SOA-investoinnin kannattavuutta ei voida arvioida lyhyen aikavälin investoinnin kannattavuusmittareilla.

Edellisessä luvussa käsiteltiin web-sovelluspalvelujen ja SOA:n kustannuksia ja hyötyjä. Hyötyjen kohdistuminen eri organisaatioille selviää tarkemmin luvussa 2 (tavoitellut hyödyt) ja 4 (arviointi ja mittaus). On hyödyllistä määritellä myös, mistä palvelupohjaisen arkkitehtuurin ominaisuuksista eri kustannukset ja hyödyt aiheutuvat. Seuraavassa taulukossa onkin esitetty joitain tärkeitä kustannuksia ja hyötyjä, joita palvelupohjaisen arkkitehtuurin eri ominaisuudet aiheuttavat (Sprott 2006). Taulukossa 5.1 on kuvattu SOA:n vaikutuksia kustannuksiin ja hyötyihin.

Taulukko 5,1.SOA:n vaikutuksia kustannuksiin ja hyötyihin.

SOA:n seuraus	Kustannus	Hyöty
Jaetut palvelut	Omistussuhdeasiat	Yhtenäiset toteutustavat
	Projektinhallinnan kustannukset Projektien väliset riippuvuudet	Suurempi valinnan mahdollisuus
	Yleiskäyttöisten ja uudelleenkäytettävien palvelujen kehityksen lisäkustannukset	Yksittäisen toteutuksen tuottavuuden paraneminen
Palvelu itsenäisesti hallittavana ja uudelleenkäytettävänä yksikkönä (kokonaisen järjestelmän sijaan)	Koordinointi	Nopeampi reagoiminen toiminnan muutoksiin
	Käytettävissä olevien resurssien hallinta	Parantunut prosessien tuottavuus
	Prosessien uudelleensuunnittelu palvelupohjaiseen toimintaan siirryttäessä	Muutosten vaikutusten rajaaminen pienempiin vaikutusalueisiin Hyödynnettävien palvelujen valvonta ja hallinta toiminnan näkökulmasta (Palvelut vastaavat käsitteellisesti todellisia prosesseja)
Toimintamahdollisuuksien suoraviivainen kuvaaminen		Palvelun tarjoajan ja asiakkaan toimintamallien parempi vastaavuus ilman erillistä välitystä
Palvelujen toimittaminen sopimus pohjaisesti	Sopimusten hallinta	Parantunut toiminnallinen tehokkuus
	Muodolliset palvelutaso-sopimukset (SLA)	Vastuiden selkeä määrittely ja erikoistuminen omalle osaamisalueelle.
Resurssien virtualisointi ja ulkoistaminen	Käyttöönottojen kokonaisuuden hallittavuus (etenkin pitkällä aikavälillä)	Suurempi valinnan mahdollisuus
		Mahdollisuus käyttää yleisiä työvälineitä, joiden perusominaisuuksiin rakentuu tuki käytettäville tekniikoille
SOA:n seuraus	Kustannus	Hyöty
Jaettu infrastruktuuri	Enterprise Service Bus ja palvelun hallintainfrastruktuuri	Reaaliaikainen palvelujen suorituksen hallinta
		Dynaaminen palvelujen valinta
		Toiminnan seuranta (esim. BAM)
Sääntöihin perustuva palvelujen suoritus	Eri järjestelmiin hajautuneiden toimintasääntöjen tiukempi kontrolli	Prosessien automatisointi
		Palvelujen itsenäisyys, toiminnan seuranta
Palvelupohjaiset prosessit	Siirtyminen Web Service-pohjaiseen prosessiteknologiaan	Web Service-palvelujen joustavuuden ja infrastruktuurin hallinnan hyödyntäminen palveluissa
Siirtymäpolkujen tukeminen olemassa olevien sovellusten pohjalta	Palvelujen rakentaminen ja käyttö olemassa olevista järjestelmistä	Yksinkertainen rajapinta olemassa oleviin sovelluksiin yksinkertaistaa sovellusintegraatiota ja alentaa ylläpitokustannuksia
		Rajapinnat ja selkeästi määritellyt palvelut tukevat perinnejärjestelmien korvaamista

5.4 Yhteenveto kustannusten ja hyötyjen muodostumisesta

Web-sovelluspalvelujen ja palvelupohjaisen arkkitehtuurin investointien kannattavuuden arvioimiseksi löytyy käyttökelpoisia mittareita. Mittareiden soveltaminen vaatii taloudellisen osaamisen yhdistämistä toiminnassa ja kehittämisessä tavoiteltaviin hyötyihin. Kannattavuuden arviointi vaatii oikeanlaisen kustannus-hyöty-analyysin tekemistä. Kustannus- ja hyötytekijöitä on helposti löydettävissä, ja etenkin kustannusten arviointiin löytyy myös taloudellisia malleja. Hyötytekijöille ei sen sijaan löydy helposti sopivia tarkkoja malleja, joten hyötyjen arviointi sekä oikea kohdistaminen voi olla hankalaa. Hyötyjä saavat sekä sovellusten tuottajat että asiakasorganisaatiot, mutta aina ei ole selvää, mitkä hyödyt todellisuudessa realisoituvat ja mitkä näistä kohdistuvat kullekin osapuolille. Hyötyjen arvioiminen rahallisesti oikein voi olla myös ongelmallista, koska osaa hyödyistä on vaikea rahallistaa. Palvelupohjaiseen arkkitehtuuriin tehtävät investoinnit ovat suhteellisen pitkän aikavälin investointeja. Tämä johtuu siitä, että palvelupohjaisen arkkitehtuurin hyödyt realisoituvat vähitellen ja kasvavat arkkitehtuurin kehittyessä. Tämän vuoksi arkkitehtuurin ja organisaation toiminnan kehittäminen vaatii pitkäjänteistä työtä. Arkkitehtuurin alkuvaiheessa hyödyt ovat projektitason hyötyjä ja arkkitehtuurin kehittyessä aletaan hyötyjä enenevässä määrin saada myös palvelujen uudelleenkäytöstä sekä alenevista integraatiokustannuksista. Myöhemmässä vaiheessa hyötyjä saavutetaan prosessien paremmasta muokattavuudesta ja yhtenäisyydestä. Vasta pitkälle kehittyneessä palveluarkkitehtuurissa saavutetaan suurimmat strategiset hyödyt. Mikäli palvelupohjaisen arkkitehtuurin ja organisaation toiminnan kehittäminen ei ole riittävän pitkäjänteistä ja oikeansuuntaista, osa hyödyistä jää saavuttamatta.

6 Sovelluspalvelut ja rajapinnat ohjelmistohankinnoissa

Hankintakäytäntöjen määrittelyllä ja kehittämisellä pyritään parantamaan hankintojen yhdenmukaisuutta sekä tarjoamaan valmiita malleja hankintojen suunnitteluun, toimittajien ja ohjelmistojen arviointiin sekä valintojen tekemiseen (IEEE 1998). Nämä tavoitteet ovat myös tämän luvun keskeisiä lähtökohtia. Luku kokoaa tietojärjestelmähankintoihin liittyvää materiaalia ja soveltaa sitä suhteessa palveluarkkitehtuuriin ja sovelluspalveluihin. Luvussa käsitellään hankintaprosessin, vaatimusmäärittelyjen, hankintoihin liittyvien arviointikriteerien, tarjouspyyntöjen ja sopimusten merkitystä suhteessa palvelu- ja rajapintapohjaiseen lähestymistapaan. Erityisesti käsitellään sitä, kuinka palvelupohjaisuus vaikuttaa hankintoihin ja mitä seikkoja hankinnoissa on huomioitava kiinnitetäessä huomiota ratkaisujen joustavuuteen ja liitettävyyteen, jotka ovat palvelupohjaisen lähestymistavan keskeisiä tavoitteita.

Tämän luvun alussa käydään läpi yleisesti eri hankintatapoja ja hankintaprosessin vaiheita. Eri vaiheita kuvataan aliluvuissa tarkemmin, ja näissä kuvauksissa huomioidaan rajapintojen ja sovelluspalveluiden merkitys hankinnoissa. Lisäksi luvussa kuvataan julkisten hankintojen erityispiirteitä ja vedetään yhteen hankintoihin liittyviä seikkoja ja suosituksia sekä esitetään tarkistuslistoja hankintoja varten.

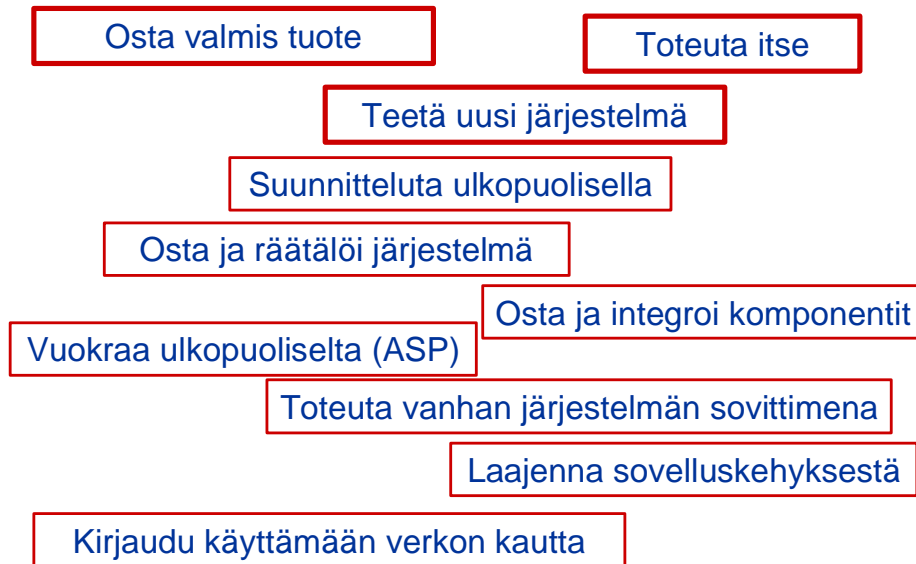
6.1 Hankintatavat ja hankintaprosessi yleisesti

6.1.1 Hankintatapojen luokittelua

Tietojärjestelmähankinnoissa on nykyisin usein kysymys valmissovellusten hankinnasta, jotka eivät sellaisenaan vastaa käyttäjien odotuksia tai kaikkia paikallisia tarpeita. Sovelluksia tulee varsinaisten ohjelmistohankintojen lisäksi myös erilaisten laitehankintojen ja kehityshankkeiden yhteydessä. Uusien valmissovellusten sovittaminen muihin organisaatioissa käytettyihin sovelluksiin muodostaa yhtä suuremman osuuden hankinta- ja käyttöönottohankeista.

Yksinkertainen hankintatapojen perusluokittelu voidaan tehdä tarkastelemalla, kohdistuuko hankinta valmistuotteeseen (COTS), paikallisesti räätälöitävään tuotteeseen (MOTS) vai tilanne- tai asiakaskohtaisesti kehitettävään tietojärjestelmään (IEEE 1998). Sovellusten muokkaaminen omia tarpeita vastaavaksi voi johtaa yhteensopivuusongelmia erityisesti organisaatioiden välisessä yhteistyössä, ja hankinnoissa pyritään yleensä suosimaan sopivia kaupallisia tuotteita.

Ajan mittaan myös vaihtoehdot uuden toiminnallisuuden hankkimiseksi tietojärjestelmiin ovat monipuolistuneet. Kuvassa 11 on lueteltu joukko vaihtoehtoja, joilla uusia järjestelmiä ja komponentteja voidaan hankkia osaksi tietojärjestelmäkokonaisuutta. Eri vaihtoehdot sisältävät omia erityispiirteitään, ja niissä esimerkiksi käytetään vaihtelevasti hankkivan organisaation omia ja ulkopuolisia resursseja. Vaihtoehtojen runsaus vaatii selkeää hankintastrategioiden ja -tapojen määrittelyä organisaatiokohtaisesti. Kuvassa esitettyjen hankintavaihtoehtojen lisäksi on joissakin tapauksissa mahdollista ulkoistaa tarvittavia tietojärjestelmäpalveluja tai toimintoja, käyttää ei-tietoteknisiä ratkaisuja esitettyjen vaatimusten toteuttamiseen, tai muuttaa toimintatapoja, prosesseja tai olemassa olevien sovellusten käyttöä siten, ettei varsinaisia ohjelmistohankintoja tarvitakaan.



Kuva 11. Tietojärjestelmäkomponenttien hankintavaihtoehtoja.

Eräs hankintoihin olennaisesti liittyvä tekijä on, tulevatko palvelut "itse hallittaviksi", ja millaista palvelin- tai muuta infrastruktuuria tällöin tarvitaan. Toiminnan kannalta kriittisissä järjestelmissä on suosittu toimintatapaa, jossa voidaan paikallisesti ylläpitää ja hallinnoida järjestelmiä, joskin myös tällöin varsinainen työ voidaan usein hankkia asiakasorganisaation ulkopuolelta. Toinen olennainen hankintatapoja toisistaan erottava tekijä on, kuinka syväälle ratkaisut vaikuttavat olemassa oleviin järjestelmiin (invasiivisuus). Joissakin tapauksissa jo käytössä olevista järjestelmistä voidaan yhdistelemällä tai sovittimilla kehittämällä tuottaa uusiin vaatimuksiin vastaavia tietoja ja toimintoja. Erityisesti ulkopuolisten sovellus- ja käyttöpalvelujen hyödyntämisessä sekä ylläpitopalveluissa palvelutasosopimukset (SLA) ovat keskeisiä.

Rajapinnoilla on suuri merkitys lähes kaikissa yllä kuvatuissa hankintatavoissa. Monissa tilanteissa ulkoa hankittu tai vuokrattu valmisohjelmisto voi sisältää hyvät mahdollisuudet sisältämiensä toimintojen ja tietojen tukemiseen, mutta sikäli kuin samoja tietoja tarvitaan myös muissa järjestelmissä, ei löydykään valmiita rajapintoja tai mahdollisuuksia niiden siirtoon tai yhteiskäyttöön.

6.1.2 Palveluarkkitehtuurin hallinta ja hankinnat

Siirryttäessä kohti palvelupohjaista ajattelua on ensi vaiheessa kiinnitettävä huomiota jo hankittujen ja hankittavien tietojärjestelmien rajapintoihin tai mahdollisuuksiin rajapintojen toteuttamiseksi (esim. sovittimet). Pidemmälle edenneessä palvelupohjaisessa ajattelussa pyritään hahmottamaan tietojärjestelmäkokonaisuus erillisten sovellusten sijasta joukkona komponentteja, joista osa voi olla myös verkon yli organisaation ulkopuolelta käytettäviä. Useissa organisaatioissa yhdistellään jo nykyisin eri hankintatapoja, mutta niiden yhteensovittaminen käyttäjän ja tietohallinnon kannalta on haasteellista.

Palvelupohjaiseen arkkitehtuuriin perustuvassa järjestelmäkokonaisuudessa rajapintojen määrittelyt muodostavat yhden tärkeimmistä hankintoja ohjaavista tekijöistä. Rajapintamäärittelyjen avulla kuvataan, mitä tietoja ja toimintoja yksittäiset komponentit sisältävät tietojärjestelmäkokonaisuudessa. Määrittelyt voivat perustua yleisiin ja avoimiin standardeihin, tai ne voivat olla paikallisia tai tuotekohtaisia. Hankinnasta vastaavien tehtävänä on selvittää, millaisia rajapintoja tarvitaan tai kuinka valmiiden komponenttien tarjoamat rajapinnat sovitetaan järjestelmäkokonaisuuteen. Tämä

vaatii hankintojen ohjaamista "arkkitehtuuripelisääntöjen" avulla, ja esimerkiksi rajapintamäärittelyjen käyttämistä tarjouspyyntöjen liitteinä.

Rajapintojen lisäksi palvelupohjaiseen lähestymistapaan liittyen on yhä selvemmin nousemassa esiin tarve hallita järjestelmällisesti ja mitattavasti kokonaisarkkitehtuurin kehittämistä ja siihen liittyviä hankintoja. Käsitteenä on nostettu esiin palveluarkkitehtuurin hallintaa (SOA governance). Hallinnassa (governance) on yleisesti kaksi osaa (Holley ym. 2006). Ensinnäkin hallinta edellyttää, että organisaatio määrittelee prosessit joiden avulla saadaan selville, kuka vastaa erityyppisistä päätöksistä. Toiseksi tarvitaan mekanismit ja toimintatavat tehtyjen päätösten toimeenpanoa ja hallintaa varten. Näistä kahdesta tekijästä muodostuu hallintamalli (governance framework). Tietotekniikan hallinta on osa koko organisaation hallintaa, ja sisältää tietotekniikkainvestointeihin liittyvien päätöksenteko-oikeuksien määrittelyt sekä toimintatavat ja prosessit, joilla mitataan ja ohjataan tietotekniikkapäätösten priorisointia ja toteuttamista.

Palveluarkkitehtuurin kehittämisessä nousee esiin monia seikkoja päätösvaltuuksiin, mittaukseen ja hallintaan liittyen. Palveluarkkitehtuurin hallinta (SOA governance) on tietotekniikan hallintaa, jossa keskitytään sovelluspalvelujen ja niistä koostettujen sovellusten elinkaareen organisaation palvelupohjaisessa arkkitehtuurissa (Holley ym. 2006). Hallintamalli sisältää suuren joukon vastuuta ja menettelytapakysymyksiä liittyen arkkitehtuurin ja sovelluspalvelujen omistajuuteen, kehittämiseen ja hankintoihin sekä seurantaan ja hyötyjen mittaukseen. Hallintamallissa on otettava huomioon, miltä osin palvelupohjaisuuden kehittäminen perustuu käytössä olevien järjestelmien uudelleenkäyttöön ja kapselointiin, miltä osin taas uusien sovelluspalvelujen ja uuden sovelluslogiikan kehittämiseen. Palveluarkkitehtuurin hallintamalliin kuuluu olennaisena osana päätöksenteko-oikeuksien määrittely, tarvittavien palvelujen tunnistamisen ja määrittelyn ohjaus, voimavarojen ja resurssien ohjaus (asset management) sekä toiminnan tehokkuuden seuranta ja mittaus. Hallintamallin avulla voidaan pyrkiä yhdenmukaisuuden lisäämiseen tietotekniikka-arkkitehtuurissa, prosesseissa, infrastruktuurissa ja käyttökokemusten osalta. Uudelleenkäytön lisäksi on muistettava huomioida myös sovelluspalvelujen versiointi, päivitysmekanismit ja käytön seuranta, jotta voidaan välttyä virhetilanteilta esimerkiksi silloin, kun paljon käytetyn sovelluspalvelun toimintaan tehdään muutoksia. Koska uudelleenkäyttö voi olla ennakoimatonta, on tärkeää seurata myös suorituskykyä ja kuormitusta sekä sitä, kuinka käyttö jakaantuu. Palvelutasosopimukset (SLA) ovat keskeinen väline monien näiden seikkojen määrittelyssä.

Toistaiseksi ohjelmistopalveluja on käytetty enimmäkseen organisaation sisäisessä integraatiossa, mutta niiden kehittämisessä on alusta lähtien korostettu järjestelmien yhteentoimivuutta yli organisaatorajojen ulottuvissa prosesseissa etenkin sähköisen kaupankäynnin alueella. Kun järjestelmiä yhdistetään paikallisesti, voidaan yleensä nojautua paikallisten ohjeiden ja yhtenäisen arkkitehtuurin (enterprise architecture) ja hallintamallin käyttöön. Sen sijaan eri organisaatioiden välisissä liitännöissä on kiinnitettävä enemmän huomiota erilaisten paikallisten käytäntöjen yhteensovitukseen, keskitetyn hallinnan puuttumiseen ja vastuiden tarkkaan määrittelyyn sopimusten avulla. Palveluarkkitehtuurin hallintamallin kehittämiseen on esitetty yleistettyjä malleja (esim. Holley ym. 2006), mutta mallin kehittämisessä on syytä huomioida myös hankintoihin liittyvät ohjeet ja mallit.

6.1.3 Hankintaprosessien mallit

Järjestelmähankintoja voidaan toteuttaa erilaisia vaiheita noudattaen. Avaimet käteen -periaatteella tehtävissä hankinnoissa ostaja esittää tarjouspyynnössä haluamansa lopputuloksen ja käytettävissä olevan budjetin. Vaiheittaisessa hankinnassa asiakas suorittaa vaatimusten määrittelyn tai osallistuu siihen ja hankkii vaatimusten mukaisen toteutuksen joko yhdeltä tai useammalta toimittajalta (Kavén, Hartikainen 2001; Saranummi 2003).

Hankintaprosessi on hyvä toteuttaa projektina, jossa tarjouspyynnön valmistelu erotetaan omaksi kokonaisuudeksi varsinaisesta käyttöönottoprojektista. Valmisteluvaiheessa keskitytään tarpeiden määrittelyyn ja tutustutaan olemassa olevaan tarjontaan sekä esitetään toivomuksia uusista kehittämistarpeista (Kavén, Hartikainen 2001).

Riippumatta siitä, kuinka hankinta on vaiheistettu, tietojärjestelmähankintaan tulisi kuulua hankintaan liittyvä strategia, hankintasuunnitelma, kustannusten, aikataulun ja suoritusten rajat, käyttäjien vaatimukset, tuotteen suunnittelu, itse hankintaprosessi, riskien kartoitus ja hallinta (Bernard ym. 2005).

IEEE-standardissa 1062 (IEEE 1998) ohjelmistojen hankintaprosessi on jaettu yhdeksään vaiheeseen, joita kuvataan taulukossa 4.. Eri vaiheita voidaan toteuttaa osin samanaikaisesti, esimerkiksi toimittajien tunnistaminen (vaihe 4) ja sopimusvaatimusten valmistelu (vaihe 5) eivät välttämättä ole toisistaan riippuvaisia. On huomioitava, että kuvattu prosessi keskittyy tilauksesta tehtävien tai muokattavien ohjelmistojen hankintaan, ei niinkään sellaisenaan käytettävien valmisohjelmistojen.

Tietojärjestelmän hankintaan liittyvää kokonaisprosessia voidaan hallita ja arvioida eri näkökulmista (Bernard ym. 2005). Ensimmäisenä alueena on projektin johtamisen prosessi, johon kuuluu projektin suunnittelu, projektin valvonta ja kontrollointi, tarjouspyyntöjen ja sopimusten valvonta, yhteis-työ/yhteisprojektien johtaminen ja riskienhallinta. Toisena alueena on hankintaan liittyvä tuotanto-prosessi, johon sisältyvät vaatimusmäärittelyt, vaatimusten hallinta, verifiointi(todentaminen) ja validointi(vahvistaminen/kelpoistaminen). Kolmantena alueena on prosessin tuki, johon sisältyy päätösten analysointi ja tekeminen. Kaikkiin osa-alueiden suunnitelmiin sisältyvät myös arvioinnit ja arvioinnin kriteerit, joiden mukaan tietojärjestelmäprojektia voidaan seurata ja arvioida.

Edellä mainitut näkökulmat kulkevat läpi koko hankintaprosessin. Luvun 6.2 sisältö on ryhmitelty taulukon 6.1 päävaiheiden mukaisesti, mutta kussakin vaiheessa on syytä tarkastella sekä johtamisen, hankintaprosessin toteuttamisen (tuotannon) että prosessien tuen näkökulmia.

6.1.4 Julkisen hankintaprosessin erityispiirteitä

Julkisten hankintojen hankintaprosessiin kuuluu seuraavia vaiheita: hankinnan suunnittelu, tarjouspyynnön laatiminen, hankinnasta ilmoittaminen, hankintamenettelyt, menettely, kun tarjous on saatu, sekä hankintapäätös. Hankintaprosessia ohjaavat hankintojen kynnyksarvot ja määräajat (Hankinnat 2007). Koneiden, laitteiden ja järjestelmien hankinnat ovat julkisen hankinnan näkökulmasta investointihankintoja (Remes 2005).

Julkisista hankinnoista annettu laki ja asetukset eivät säätele hankinnan aloitusvaihetta. Hankinnan suunnittelussa on kuitenkin kysymys julkisten varojen tehokkaan käytön kannalta tärkeästä vaiheesta ja siitä, miten julkisyhteisö suunnittelee varojen käyttöönsä (Hankinnat 2007). Esimerkiksi HUS:in julkisten hankintojen (Remes 2005) ensimmäiseen vaiheeseen kuuluvat tarpeen arviointi, hankinnan suunnittelu, budjetointi ja hyväksyntä. Investointihankintaprosessi alkaa useiden eri käyttäjien tekemästä tarveselvityksen laadinnasta ja aikataulun suunnittelusta, johon osallistuu myös hankintaosasto. Sen jälkeen tarveselvitys käsitellään hallinnossa (HUS yhtymähallinto) ja investointiohjelma hyväksytään joko hallituksessa, johtokunnassa, lautakunnassa tai hankintatoimikunnassa (riippuu investoinnin suuruudesta) tarveselvityksen perusteella. Tämän jälkeen investointiohjelma hyväksytään budjettiin valtuustossa, jonka jälkeen käyttäjät laativat investointispesifikaatiot (tekniset vaatimukset) ja niiden perusteella laaditaan tarjouspyyntö sekä valitaan hankintamenettely hankintaosastolla ja materiaalikeskuksessa.

Taulukko 6.1. Ohjelmistojen hankintaprosessin vaiheet (mukailtu IEEE 1998).

Päävaihe	Vaihe hankintaprosessissa	Tehtävien lähtökohdat	Tehtävien tuotokset
Suunnittelu- vaihe	1 Hankintastrategian suunnittelu	Hankkijan tavoitteet	Ohjelmistolta tarvittavat laatu- piirteet, organisaation hankinta- strategia, yleiset toimintakäy- tännöt
	2 Hankintaprosessin toteuttaminen	Valitun hankintaprosessimallin vaiheet, strategia ja toimintatavat, sopimuskäytännöt	Organisaatioon sovitettu hankintaprosessi ja toimittajan arviointi- ja valintaprosessi
	3 Vaatimusmäärittelyt	Ohjelmiston alustavat määrittelyt, toimittajien arviointikriteerit, hankkijan ja toimittajan velvollisuudet, alustavat laatu- ja ylläpitosuunnitelmat	Ohjelmiston tarkat määrittelyt, laatu- ja ylläpitosuunnitelmat, ohjeet tarjousten arviointiin, valvontasuunnitelma, tarjouspyyntö
Sopimus- vaihe	4 Toimittajien tunnistaminen	Tarjoukset, toimittajien aiempi suoriutuminen, toimittajien arviointikriteerit, ohjelmiston tarkat määrittelyt, käyttäjäkyselyn kriteerit ja kysymykset	Lista potentiaalisista ohjelmitoista ja toimittajista, käyttäjäkyselyn tulokset
	5 Sopimusvaatimusten valmistelu	Tarjoukset, toimittajan ja hankkijan vastuut, toimittajien arviointimallit, hankintaehdot, laadunvarmistuspykälät ja maksuehdot	Tarjouspyyntö, hyväksymiskriteerit, toimittajien arviointikriteerit, ohjelmiston arviointi- ja testauskriteerit, maksujen sitominen tuotoksiin, sopimuksen sisältö ja lainmukaisuus
	6 Tarjousten arviointi ja toimittajan valinta	Tarjoukset, tarjousten ja toimittajien arviointimallit ja -prosessit, käyttäjäkyselyn tulokset, laadunvarmistuspykälät	Arvioidut tarjoukset, lista hyväksytyistä toimittajista, toimittajan valinta, valmiiksi neuvoteltu sopimus
Toteutus- vaihe	7 Toimittajan ohjaus	Sopimus, sopimuksen tarkastuspisteet, sopimuksen mukaiset tuotokset, edistymisseuranta, edistymisen arviointikriteerit	Hyväksytyt työpaketit, tarkastuspisteet ja tuotokset, laatu- ja luotettavuusseuranta, palaute toimittajalle
Hyväksymis- vaihe	8 Ohjelmiston hyväksyminen	Hyväksymiskriteerit, arviointikriteerit, testauskriteerit, laatu- ja ylläpitosuunnitelmat, toimittajan arviointikriteerit, hyväksymisprosessin määrittely	Hyväksymisprosessi, hyväksyttävissä oleva ohjelmisto, käyttökelpoinen dokumentaatio
Seuranta- vaihe	9 Ohjelmiston käyttö	Ohjelmistot, dokumentaatio, tukipalvelut, laatusuunnitelma, ylläpitosuunnitelma	Hankinta- ja sopimuskäytäntöjen arviointi, muutuskäytännöt, käyttäjätyytyväisyysarviointi, toimittajapalaute ja -arviointi

Pääsääntöisesti kaikki julkiset hankinnat tulee kilpailuttaa. Hankintalain mukaan hankinta saadaan tehdä ilman tarjouskilpailua vain erityisestä syystä tai niiden ehtojen perusteella, jotka on asetettu suoran neuvottelumenettelyn, suorahankinnan tai puitejärjestelyjen käytölle (Laki 2007/348). Hankintayksikön on tehtävä kansallinen hankintailmoitus kansalliset kynnsarvot ylittävistä hankinnoista. Kynnsarvo ratkaisee hankintamenettelyn siten, että arvon ylittävien hankintojen osalta yleisimmin on käytettävä avointa tai rajoitettua ilmoitusmenettelyä¹ kun taas sen alittavissa hankinnoissa menettely on joustavampaa. Kynnsarvon ylittävien hankintojen ennakoilmoituksilla on myös vähimmäismääräajat, jotka riippuvat siitä onko kyseessä avoin vai rajattu menettely. 1.1.2006 alkaen esimerkiksi julkisten hankintojen kynnsarvona ennakoilmoitusta edellyttävissä tavara- ja palveluhankinnoissa on ollut 750.000 euroa (Hankinnat 2007). Hankintailmoitusmenettelyn tarkoituksena on avoin tiedottaminen tarjouskilpailusta. Jotta todellinen kilpailu toteutuisi, on tärkeää, että yritykset saavat tasapuolisesti tiedon meneillään olevista tarjouskilpailuista. Näin taataan myös aito kilpailu ja mahdollisuus tehdä hankinnat edullisesti. Ehdokkaita on kutsuttava tarjouskilpailuun hankinnan kokoon ja laatuun nähden riittävä määrä. Tarjouskilpailuun on rajoitetussa menettelyssä kutsuttava vähintään viisi ehdokasta, jollei soveltuvia ehdokkaita ole vähemmän (Laki 2007/348).

Hankinnan hyväksymisen ja hankintamenettelyn valinnan jälkeen siirrytään varsinaiseen ostoprosessiin. Vaihe alkaa hankintailmoituksella ja tarjouspyynnön julkaisemisella (ellei ole tehty ilmoitusta), joista huolehtii esim. HUS:in tapauksessa materiaalikeskus (Remes 2005). Tarjouspyyntö menee toimittajille ja tietyn tarjousajan kuluessa toimittajat lähettävät tarjouksensa tarjouksen pyytäjälle, joka ottaa vastaan tarjoukset. Tarjouksia ei saa avata ennen tarjousajan päättymistä, ja niitä tulee käsitellä siten, ettei ketään tarjoajaa syrjitä. Tarjouksen pyytäjän kannattaa laatia avaustilaisuudesta lyhyt pöytäkirja, jossa mainitaan tilaisuuden aika ja paikka, läsnäolijat, saapuneiden tarjousten lukumäärä sekä luettelo tarjoajista (Hankinnat 2007). Valtion viranomaisia koskeva asetus valtion hankinnoista edellyttää pöytäkirjan laatimista. Tilaisuudessa olisi hyvä olla läsnä yksi tai useampi ns. ”esteetön” henkilö, joka ei siis osallistu varsinaiseen valintamenettelyyn.

Seuraavaksi on varsinainen valintaprosessi, jossa organisaation omat asiantuntijat vertaavat hintoja ja ominaisuuksia ja käyttäjät tutustuvat käyttöympäristöön sekä testaavat tarjottua tuotetta. Tämän jälkeen käyttäjät, hankintaosasto tai hankintatoimikunta tekee hankintaesityksen. Hankintalainsäädännön mukaisen tarkistuksen tekevät esim. HUS:illa käyttäjät ja materiaalikeskus (Remes 2005). Hankintapäätöksen hyväksyy hankintaesityksen perusteella hankinnan koosta riippuen joko toimialajohtaja, tulosalueen tulosyksikön johtaja, materiaali johtaja tai virkasuhteessa oleva käyttäjä, jos hänet on tähän valtuutettu. Hankintapäätöksestä lähetetään tieto tarjouspyynnön tekijälle ja samalla hankintapäätöksestä lähetetään toimittajille tieto onko tuote valittu vai ei. Odotusajan jälkeen tehdään sopimus valitusta tuotteesta toimittajan kanssa.

Hankinnan kolmanteen vaiheeseen kuuluu hankinnan käytännön toteutus - toimitus ja käyttö. Investointihankintaan kuuluu tässä vaiheessa tilaus ja tuotteen toimitus. Esimerkiksi HUS:illa materiaalikeskus tekee tilauksen, ja hankintaosasto, materiaalikeskus tai lääkintätekniinen keskus huolehtivat vastaanotosta, hyväksynnästä, asennuksesta ja laskutuksesta (Remes 2005). Hallintaosasto vastaa käyttäjien koulutuksesta ja investoinnin käytöstä.

¹ Avoimessa menettelyssä hankintayksikkö julkaisee hankinnasta hankintailmoituksen (minkä lisäksi voidaan lähettää tarjouspyyntöjä), ja halukkaat toimittajat voivat tehdä tarjouksen. Rajoitetussa menettelyssä hankintayksikkö julkaisee hankintailmoituksen, johon halukkaat toimittajat pyytävät saada osallistua, mutta vain hankintayksikön valitsemat ehdokkaat voivat tehdä tarjouksen. Hankintayksikkö voi ennalta rajata tarjousmenettelyn ehdokkaiden määrää (Laki 2007/348).

Kuntaliiton ja Kauppa- ja teollisuusministeriön julkisten hallintojen neuvontayksikkö tarjoaa tietoa ja opastusta julkisiin hankintoihin liittyen. Yksikön kautta löytyy myös julkisten hankintojen sähköinen markkinapaikka HILMA, jossa julkaistaan julkisten hankintojen ilmoituksia. Lisäksi hankintoihin liittyvää tukea ja malleja löytyy mm. Efecon kautta osoitteessa www.kilpanet.com. Hankintojen osaamisen kehittäminen on myös Valtionhallinnon IT-kehittämisessä (ValtIT) keskeinen osa. Hankintoihin liittyvät kansalliset suunnitelmat sisältävät mm. hankintatoimen osaamiskeskuksen perustamisen sekä hankintakäytäntöjen ja -lainsäädännön uudistamisen. Näillä toimenpiteillä pyritään mahdollistamaan hankintojen tehokas keskittäminen, yhteishankintamenettelyjen käyttö ja puitesopimuskäytäntöjen kehittäminen kunta- ja valtiosektorilla.

Ohjelmistopalvelut ja rajapinnat eivät sinällään vaikuta suoraan julkisen hallinnon hankintamenettelyihin. On kuitenkin huomioitava, että rajapintoja ja sovelluspalveluja voidaan joko hankkia osana muutenkin hankittavia järjestelmiä, osana järjestelmien päivityksiä, tai erikseen tilattuina ja toteutettuina. Koska sovelluspalvelut ovat kokonaista järjestelmää pienempiä osia, yksittäisten ohjelmistokomponenttien hankinnat voivat usein myös alittaa kynnyksarvot, jolloin voidaan joissakin tapauksissa vältyä suurimpien hankintojen ilmoitus- ja kilpailuttamismenettelyiltä. Palvelupohjainen tietojärjestelmäkokonaisuuden kehittäminen edellyttää pienempiä ja useammin tapahtuvia hankintoja ja päivityksiä kokonaisjärjestelmiin, jolloin raskaat kilpailutustoimenpiteet kunkin tarpeen yhteydessä hidastavat ratkaisujen käyttöön saantia huomattavasti. Tätä varten onkin tarpeen suosia puitesopimuksia ja riittävän keveitä tarjous- ja kilpailutusmenettelyjä kokonaisuuden kehittämisessä. Lisäksi hankinnoissa pyritään yhä enemmän parantamaan alueellista yhteistyötä ja organisaatorajojen yli ulottuvien palvelukokonaisuuksien rakentamista. Tällöin hankinnoissa on huomioitava varsinainen käyttäjäorganisaation lisäksi yhä laajemmin myös alueelliset kehityssuunnitelmat ja synkronointi muiden alueellisten hankkeiden kanssa (Kavén, Hartikainen 2001).

Valtaosa rajapintoihin ja sovelluspalveluihin liittyvistä seikoista liittyy tarjouspyyntöjen, tarjousten ja kilpailuttamisen sisältöön. Näitä sisällöllisiä seikkoja huomioidaan seuraavissa luvuissa.

6.2 Hankintojen suunnittelu ja ohjaus

Hankintojen suunnittelussa pohjana voidaan käyttää hankintastrategiaa, joka pohjautuu esimerkiksi tietojärjestelmästrategiaan ja organisaation omaan strategiaan. Tämän pohjalta on toteutettava järjestelmällinen hankintaprosessi organisaatiossa. Toistettavaa hankintaprosessia tukevat valmiit pohjat mm. hankintasuunnitelmille ja hankintoihin liittyville vaatimusmäärittelyille. Kunkin hankinnan osalta hankintasuunnitelmat ja hankintoihin liittyvät vaatimukset tarkennetaan osana suunnittelu-prosessia.

6.2.1 Hankintastrategia

Ohjelmistohankintaan liittyvässä strategiassa suunnitellaan mikä on hankinnan kannalta sopivin tapa (ks. luku 2.1), kuinka hankintaan liittyviä riskejä vähennetään ja mitkä sidosryhmät ovat olennaisia (Bernard ym. 2005). Strategiset linjaukset ja valinnat luovat kehikon tietojärjestelmähankinnoille, mutta toiminnalliset muutokset edellyttävät myös loppukäyttäjien osallistumista (Kaven, Hartikainen 2001).

Organisaation tietojärjestelmien kehittäminen kohti palvelupohjaista kehitystapaa edellyttää tavoiteltujen hyötyjen strategista ja pitkäjänteistä määrittelyä (ks. luku 2). Nämä tavoitteet on syytä kirjata organisaation tietohallintostrategiaan ohjaamaan hankintoja yleisesti. Myös hankintoihin liitty-

vän tietotekniikan tai palveluarkkitehtuurin hallintamallin (governance) käyttöönotto ja peruslinjat kuuluvat strategiaan.

6.2.2 Hankintaprosessin toteuttaminen ja hankintasuunnitelma

Hankintasuunnitelma sisältää hankintastrategian toteutussuunnitelman, ohjelmiston laajuuden määrittelyn ja dokumentoinnin sekä hankintaprojektiin kuuluvien osapuolten toimenpiteiden kuvaukset. Suunnitelmassa on lisäksi määritelty kehittämisspyrkimyksen laajuus, resurssien tarve projektin eri osioissa, sekä se, kuinka kriittiset polut on organisoitu sidosryhmien, johdon ja työntekijöiden osalta (Bernard ym. 2005). Suunnitelmassa kuvataan myös tarvittavat resurssit, kuten työkalut, henkilökunnan hankinta ja koulutus sekä se, kuinka varmistetaan toimittajan osaaminen ja kyky toteuttaa prosessin vaatimukset.

Kustannusten, aikataulun ja suoritusten suunnittelussa ja rajauksissa varmistetaan, että ne ovat yhteensopivia, realistisia ja mahdollisia toteuttaa (Bernard ym. 2005). Sopimusehdoissa kutakin näistä osioista tarkastellaan usein itsenäisesti. Suunnittelussa varmistetaan mitä kokonaiskustannuksiin sisältyy (esim. testaus, opetus, ylläpito ja tuki) sekä kuvataan kuinka kustannuksia, aikataulua ja suorituksia seurataan. Lisäksi kuvataan, kuinka riskeihin ja tuotantomuutoksiin varaudutaan, kuinka muutoksia johdetaan ja kuinka kustannus- ja aikataulumuutoksia arvioidaan sopimuksen perusteella. Hankintaprojektiin liittyen suunnitellaan, kuinka käyttäjät ja muut sidosryhmät osallistuvat vaatimusten määrittelyyn, eri toimijoiden roolit vaatimusmäärittelyssä ja strategia, jolla kehitystä pidetään yllä toimintaympäristössä.

Itse tuotteen suunnitteluun liittyviä näkökohtia ovat mm. vaatimusmäärittelyt ja kuinka oikeita ne ovat, kehitystilanteen seuranta ja arviointi, sekä ohjelmistojen lisäksi muiden projektiin kuuluvien palvelujen sisältö. Lisäksi on määriteltävä ohjelmistojen osalta, sisältyykö hankintaan varsinaisen suoritettavan koodin lisäksi suunnitelma- ja määrittelydokumentteja, testausmateriaalia, välineitä, lähdekoodia, koodikirjastoja, tietokantoja tai -hakemistoja ja asennus-, ylläpito- ja käyttödokumentteja (Scott & Nisse 2004).

Hankintasuunnitelmassa hankintaprosessista tulisi kuvata prosessin sisältö ja lähtökohdat sekä se, kuinka prosessia valvotaan, kontrolloidaan ja edistetään projektin yhteydessä. Lisäksi hankintaan liittyen voidaan kuvata hankintaprojektiin liittyvät riskit, joita on tullut esille hankintastrategian ja suunnitelman yhteydessä. Riskienhallintaan kuuluu arvioida koko projektin riskejä ja niiden todennäköisyyttä ja seurauksia, kuvata riskienhallinnan työkalut, määrittellä riskien tunnistamis- ja vähentämistoimenpiteet ja vastuut riskien arvioinnista (Bernard ym. 2005). Riskejä voi liittyä mm. kustannuksiin ja aikatauluihin sekä toimittajan toimituskykyyn, samoin kuin löydettyjen riskien merkityksen ymmärtämiseen. Riskienhallintaan liittyen on tärkeää tunnistaa, millaiset riskit ovat oman kontrollin ulkopuolella ja millaisilla resursseilla todellisten riskien vaikutuksia vähennetään, sekä miten toimittajia arvioidaan.

Valittu hankintatapa ja hankinnan koko ohjaavat pitkälti hankintasuunnitelman sisältöä. Yhteentoinnivuuden ja yhteensopivuuden varmistaminen korostuvat rajapintoja ja sovelluspalveluita käsittelevissä hankintasuunnitelmissa. Tätä varten hankintasuunnitelmiin voidaan kirjata erityisiä toimenpiteitä yhteensopivuuden katselmointiin ja testaukseen sekä määrittellä keskeiset organisaation arkkitehtuurimäärittelyt hankinnan lähtökohdiksi. Jos ja kun kokonaisjärjestelmä koostuu eri lähteistä hankituista sovelluksista tai komponenteista, erityistä huomiota on kiinnitettävä siihen, miten sovitusorganisoituaan ja ylläpitotyön vastuisiin.

6.2.3 Vaatimusmäärittelyt

Vaatimusmäärittelyihin kuuluu toiminnallisten vaatimusten määrittely, jossa kuvataan se, miten järjestelmän tulee toimia ja mitä sen pitää tuottaa. Lisäksi ei-toiminnallisissa vaatimuksissa kuvataan järjestelmän laatua ja rakenteellisia ominaisuuksia. Näitä ovat järjestelmän osalta mm. tehokkuus, käytettävyys, luotettavuus, siirrettävyys ja itse tekniikkaan liittyvät vaatimukset, kuten käytettävät laitteet, tiedon tuottamisen muoto, yhteensopivuus eri arkkitehtuurien ja käytettyjen tekniikoiden kanssa (Mykkänen ym. 2004). Organisaatioon liittyviä vaatimuksia ovat organisaation strategian ja toimintaohjeiden noudattaminen, toteutustapa ja organisaation omat toimintatavat ja standardit. Ulkoisia vaatimuksia ovat puolestaan yhteentoimivuus sidosryhmien järjestelmien kanssa, eettiset vaatimukset ja lakien, asetusten ja muiden määräysten mukaisuus. Vaatimuksia kirjattaessa on ratkaistava, kuinka vaatimus kuvataan järjestelmän kannalta, miten sen toteutuminen todetaan ja mitataan. Samalla on syytä määritellä käytettävät mittarit.

Käyttäjien kuvaamista tavoitetilavaatimuksista keskeiset tulee ottaa mukaan hankintojen arviointikehikkoon huolella kuvattuina. Kehittämislle asetettavat tavoitteet toteutuvat vain, kun tietojärjestelmä tukee käyttäjien työtä ja sen ongelmien ratkaisua. Toiveiden pitää kuitenkin olla realistisia ja niiden mukaan ottamisessa on huomioitava myös kustannukset (Kavén, Hartikainen 2001).

Toimittaja mukauttaa ohjelmistoa tai toteuttaa siihen haluttuja piirteitä käyttäjien esittämien yksilöityjen vaatimusten perusteella. Ohjelmiston ominaisuudet määräävät mukauttamisen rajat. Nämä yhdessä muodostavat ne reunaehdot, jotka ovat välttämättömiä sovelluskohtaisten erojen esiin saamiseksi, jotta kokonaisvaikutuksiltaan paras vaihtoehto olisi valittavissa eri vaihtoehdoista. Hyvä vaatimusmäärittely mahdollistaa sovelluskohtaisten erojen ja parhaimman vaihtoehdon löytämisen (Kavén, Hartikainen 2001).

Käyttäjävaihtimusten kuvaus keskittyy järjestelmän ulkoiseen, nähtävissä olevaan käyttäytymiseen. Vaatimukset kuvataan usein luonnollisella kielellä, mutta sen seurauksena tarkkuus ja yksikäsitteisyys voivat kärsiä. Lisäksi kuvattuun vaatimukseen voi sisältyä useampia vaatimuksia. Jäljitettävyyden varmistamiseksi järjestelmän suunnittelussa ja toteutuksessa tulee vaatimukset jakaa riittävän pieniin ja yksikäsitteisiin osiin (Mykkänen ym. 2004). Käyttäjävaihtimukset on syytä perustella mm. tulevien muutosten ja niiden vaikutusten arvioimisen tukemiseksi. Itsestäänselvyyksiä ei kannata kirjata, mutta on huomattava että, eri ammattilaisille itsestään selvät asiat eivät ole tuttuja toisen alan ammattilaisille.

Systeemin vaatimukset tarkentavat käyttäjävaihtimuksia ja ovat perustana järjestelmän suunnittelulle, joten niiden on oltava tarkkoja ja johdonmukaisia. Nämä voidaan esittää tekstin lisäksi erilaisina kaavioina. Sovellusalueen vaatimukset kuvaavat kohdealueen tyypillisiä vaatimuksia. Hankalaa on usein se, että vaatimukset ovat kohdealueen asiantuntijoille itsestäänselvyyksiä ja jäävät siksi kertomatta vaatimusten määrittelijälle. Toisaalta näiden vaatimusten ymmärtäminen edellyttää alueen syvällistä tuntemusta ja voi olla ulkopuoliselle vaikeata ymmärtää. Hankinnan pohjaksi kannattaa laatia lista pakollisista ominaisuuksista ja toivotuista kehittämiskohteista sekä näkemys siitä millainen on lähitulevaisuuden kehittämistoiminta. Tästä on hyötyä myös potentiaalisten toimittajien kannalta.

Sovelluspalvelujen ja rajapintojen hankinta voi olla osa jotain järjestelmää tai erillinen tiettyyn tarpeeseen kohdistuva hankinta. Rajapintojen ja sovelluspalvelujen vaatimuksia muodostuu useita eri reittejä. Palvelujen ja rajapintojen sisältö (tiedot ja toiminnot) pyritään kuvaamaan sekä vaatimusmäärittelyissä että rajapintojen tai sovelluspalvelujen määrittelyissä lyhyesti ja selkeästi "scope". Keskeisten käyttäjävaihtimusten on vastattava palvelujen ja rajapintojen sisältöä. Ei-toiminnallisista vaatimuksista mm. tehokkuus, käytettävyys, skaalattavuus ja luotettavuus määritellään samantyypp-

pisesti sovellusohjelmistoissa ja sovelluspalveluissa, mutta sovelluspalveluissa on huomioitava myös tulevaisuudessa mahdollisesti laajeneva käytön volyymi. Yhteensopivuuden osalta korostuvat arkkitehtuurin määrittelyt sekä rajapintojen saatavuus ja toteutustavat. Näiden lisäksi on syytä määrittellä, millaisia mukautettavuustarpeita organisaatiolla on, ja millaisilla mekanismeilla (policy) näitä tarpeita hoidetaan.

Palvelupohjaisessa kehitystyössä hankinnat ovat aina toissijainen vaihtoehto: ensisijaisesti pyritään hyödyntämään jo hankittuja tai kehitettyjä sovelluspalveluja tai sovelluksia. Palveluarkkitehtuurilla tavoitellut säästöt perustuvat juuri tähän seikkaan. Uusien tai muuttuneiden tarpeiden yhteydessä voi olla mahdollista esimerkiksi tehdä uusi versio sovelluspalvelusta, rakentaa sovitin tai muutoksia palvelua käyttävään sovellukseen tai muuttaa organisaation policy-määrittelyjä, joilla ohjataan ei-toiminnallisia vaatimuksia. Myös toiminnallisten ominaisuuksien osalta on tarpeen määrittellä, mitkä ominaisuudet kuuluvat palvelurajapintaan, mitkä paikalliseen mukauttamiseen esimerkiksi policy-määrittelyjen avulla, mitkä taas esimerkiksi prosessimäärittelyjen tai palveluista koostettujen sovellusten vastuulle.

6.2.4 Tarjouspyynnön valmistelu

Tarjouspyynnön laatiminen on ehkä koko hankintaprosessin tärkein vaihe. Epäselvä tarjouspyyntö voi johtaa siihen, että tarjousten vertailu on vaikeaa tai jopa mahdotonta (Hankinnat 2007). Tarjouspyyntö on valmisteltava siten, että se mahdollistaa tarjousten keskinäisen vertailun. Tarjouksen tekemiseen on myös varattava kohtuullinen aika, jonka pituus esim. julkisissa hankinnoissa riippuu menettelytavasta (avoin menettely/rajoitettu menettely). Tarjouspyyntö tehdään hankinnan laadun, arvon ja kiireellisyyden mukaan kirjallisesti, telexinä, telekopiona, sähköpostina tai muulla vastaavalla tavalla. Myös tarjoukset voidaan pyytää em. tavoilla, mutta muutoin kuin kirjallisesti annetut tarjoukset pyydetään vahvistamaan kirjallisesti (Asetus 1416/1993). Tarjouspyynnöstä tulee kaikille tarjoajille selvitä lopulliset arviointiperusteet, joilla valinta tehdään. Lähtökohtaisena ratkaisuperusteena tulisi olla kokonaistaloudellinen edullisuus, jolloin määräävänä tekijänä on yleensä laatu, ei pelkästään hinta (Kavén, Hartikainen 2001).

Tarjouspyynnössä tulisi olla hintaliite, josta ilmenevät käyttöoikeus-, ylläpito-, asennus-, koulutus- ym. kulut taulukoina, jotka liitetään lopulliseen hankintasopimukseen. Erityisesti ylläpitomaksun kattavuudesta tarvitaan tieto: milloin on kyse uudesta ohjelmaversiosta ja milloin ylläpitomaksuun sisältyvä muutostyö. Uusista ohjelmistoversioista on kuvattava yhteensopivuus entisten versioiden kanssa sekä se, korvaako uusi versio entisen (revision) vai onko lisättävissä järjestelmään entisen version lisäksi (variant) (Scott & Nisse 2004).

Toteutettaessa tietojärjestelmähankintaa "avaimet käteen" periaatteella kuvataan tarjouspyyntöön visio siitä, mitä halutaan ja mitä se saa maksaa. Tämän jälkeen tuotteen toimittajat esittävät, mitä vision osia budjetin puitteissa on mahdollista toteuttaa ja miten. Vaiheistetussa hankintatavassa karotitetaan ensin käyttäjävaatimukset joko ulkopuolisen tai järjestelmää hankkivan organisaation itsensä toteuttamana. Varsinainen hankintaprojektikin voidaan toteuttaa joko kokonaisratkaisuna tai ositettuna suunnitteluun ja toteutukseen (Saranummi 2003). Hankintaprojekti itsessään on mahdollista toteuttaa joko omana toteutuksena tai ostopalveluna kokonaan tai osittain.

Hankinnan koko ja luonne ohjaavat ensisijaisesti tarjouspyynnön sisältöä. Sovellusohjelmistojen hankinnan yhteydessä on tärkeää, että tarjouspyynnön liitteenä ovat viittaukset tai kuvaukset tarvittavista rajapinnoista. Tarjouspyynnöissä voi olla viittauksia standardeihin tai avoimiin määrittelyihin, mutta niiden lisäksi tarjouspyynnöissä on kuvattava asiakkaan erityisvaatimuksia ja ei-toiminnallisia vaatimuksia, jotka eivät ole osa rajapintojen tai sovelluspalvelujen tietoja ja toimin-

nallisuutta. Mikäli hankinnan kohteena ovat sovelluspalvelut, on valittava, miten tarkalla tasolla olevia palvelumäärittelyjä tarjouspyynnöissä käytetään. Käytännössä sovelluspalveluihin kohdistuvan tarjouspyynnön liitteenä voi olla joukko rajapintamäärittelyjä, kuvaus arkkitehtuurista ja teknisestä ympäristöstä, johon tarjousten kohteena olevien ohjelmistojen on sovittava (mukaan lukien valmiit palvelut, joiden hyödyntämiseen nojaututaan), sekä luettelo suorituskyky-, saatavuus-, ja ylläpito- sekä muita paikallisia vaatimuksia.

6.3 Valinta ja sopimukset

Hankintaprosessin valinta- ja sopimusvaihe nojautuu hankintapäätökseen, hankintasuunnitelmaan sekä riittävällä tasolla määriteltyihin hankinnan tavoitteisiin ja vaatimuksiin, joita on käytetty tarjouspyynnön sisällössä. Hankintaprosessin tarjouspyyntö- ja sopimusvaiheessa kartoitetaan tai tunnistetaan toimittajat tai pyydetään avoimesti tarjouksia, valmistellaan hankintaan liittyvä sopimus, arvioidaan saadut tarjoukset tai vaihtoehdot hankinnan toteuttamiseksi ja tehdään hankittavan soveluksen tai toimittajan valinta.

6.3.1 Toimittajien tunnistaminen

Avoimet tarjouspyynnöt saatetaan avoimesti tiedoksi (esimerkiksi julkisten hankintojen hankintaportaalien (HILMA) kautta). Suljetuissa, kynnsarvot alittavissa tai puitesopimuksilla ohjatuissa hankinnoissa voidaan joissakin tapauksissa nojautua ennalta määriteltyjen ja tunnistettujen toimittajien käyttöön. Asiakkaan ja toimittajan yhteistyön toimivuuden kannalta voidaan pyrkiä muodostamaan myös pitkäaikaisia kumppanuuksia yhden tai useamman toimittajan kanssa.

Mitä tarkemmalle tasolle vaatimukset on määritelty, sitä selkeämmin voidaan tunnistaa valmiita tuotteita ja ohjelmistoja, joilla vaatimuksiin voidaan vastata. Mikäli rajapintojen ja sovelluspalveluiden hankinnoissa nojaututaan avoimiin määrittelyihin tai standardeihin, voi markkinoilta löytyä jo useita valmiita tuotteita ja toimittajia, joilla on aikaisempaa kokemusta vaatimusten mukaisten ratkaisujen toimittamisesta. Standardoitujen ja jo käytössä olevien ratkaisujen hyödyntämisellä voidaan pyrkiä kustannussäästöihin ja ratkaisujen luotettavuuden parantamiseen.

Toimittajien tunnistamisvaiheeseen voi liittyä tiedonkeruuta saatavilla olevista tuotteista, ohjelmitoiesittelyjä ja tiedonkeruuta ohjelmistojen käyttäjiltä muista organisaatioista (IEEE 1998). Lisäksi toimittajia voidaan arvioida mahdollisten aiempien sopimusten perusteella.

6.3.2 Sopimusvalmistelu

Tarjouspyynnön lisäksi sopimusta varten on tarkennettava lukuisia seikkoja. Tuotosten laatu, maksutavat, sopimuksen purku- ja muutosehdot, hyväksymis- ja toimitusehdot sekä sopimuksen lainmukaisuusseikat ovat sopimusvalmisteluun kuuluvia seikkoja.

Sopimusta varten on määriteltävä tilaajan ja toimittajan välinen suhde kokonaisuuden ja sen eri tehtävien osalta, sekä se, miten todetaan sopimuksen täyttyminen. Riippuen tarjouspyynnön tarkkuustasosta, tietty osa vaatimuksista voidaan määritellä pakollisiksi. Sopimukseen liittyen on myös määriteltävä, kenellä on valtuudet vastata toimittajan kysymyksiin, ja se, seurataanko pitkäkestoisen toimituksen edistymistä määriteltyjen tarkistuspisteiden avulla.

Vakiosopimustekstit soveltuvat yleensä huonosti tietojenkäsittelyjärjestelmien hankintaan, mutta esimerkiksi yksittäisten laitteiden tai ohjelmien hankinnassa ne ovat sopivia. Kokonaisjärjestelmien hankinnassa voidaan laatia yksilöllinen sopimus, johon on mahdollista liittää vakiosopimuksen ehtoja soveltuvien osien. Esimerkiksi sosiaali- ja terveysministeriöllä ja valtioneuvostolla on erilaisia perussopimusmalleja (www.tietosuoja.fi/6328.htm), joita voi käyttää hankintasopimuksia laadittaessa. Sopimukset eivät kuitenkaan voi kattaa kaikkia mahdollisia tilanteita ja siksi myös atk-ammattilaisten eettiset säännöt on hyvä muistaa. Sopimusehdoissa on syytä varautua myös tietojärjestelmän toimittajien omistuspohjan muutoksiin. Vastuiden jakaantuminen on ongelmallista, kun kyseessä on monen toimittajan kokonaisuus. Tämä on erityisen yleistä tilanteissa, jotka liittyvät ohjelmistopalvelujen käyttöön ja tarjoamiseen useissa sovelluksissa. Tilannetta voidaan selkeyttää yhteistyösopimuksella, jossa toimittajat yhteisvastuullisesti takaavat järjestelmän toimivuuden ja asiakas osallistuu vain järjestelmän toimivuuden testaamiseen ja jossa maksut sidotaan hyväksytyihin työsuorituksiin. Toinen selkeä malli on sellainen, jossa ohjelmistopalvelua tarjoava sovellus, sen rajapinta ja ei-toiminnalliset ominaisuudet on kuvattu niin tarkasti, että sitä sovelletaan täsmälleen samalla tavalla kaikissa palvelua käyttävissä sovelluksissa. Sopimustekstissä on hyvä olla yksiselitteinen määritelmä siitä, milloin on kyseessä uuden käyttöoikeusmaksun vaativa versio ja milloin versiopäivitys sisältyy ylläpitokustannuksiin (Kavén, Hartikainen 2001).

Kustannukset tulisi sopimuksessa ilmoittaa mahdollisimman kattavasti ja yksilöidysti, jotta vältytään ennakoimattomilta kustannuksilta. Kustannuksista, joiden tarkka suuruus ei ole tiedossa, tulee ilmoittaa kustannusarvio - esimerkiksi pidemmän aikavälin kustannuksista kuten ylläpitokustannuksista. Sopimuksessa voi olla maininta, ettei asiakas ole velvollinen maksamaan laskuja, joiden perustetta ei löydy hintaliitteestä (Kavén, Hartikainen 2001).

Usein on myös järkevää erottaa toisistaan hankinta- ja ylläpitovastuut joko erillisiin sopimuksiin tai siten, että hankintasopimukseen sisältyy tietty perusylläpito ja mahdollisia reunaehtoja ylläpidon jatkamiselle uusien sopimuksin. Sopimuksessa on oleellista, että vastuut on yksilöity mahdollisimman tarkasti. Maksut on hyvä sitoa hyväksytyihin ja tarkastettuihin työsuorituksiin. Päätöksentekovaiheessa huomioimattomia merkittäviä lisäkustannuksia ei sopimuksenteon jälkeen enää saa syntyä (Kavén, Hartikainen 2001).

Komponentti- ja palvelupohjaisessa ohjelmistokehityksessä sopimuksella viitataan rajapintamäärittelyyn ja muihin dokumentteihin, joiden pohjalta ohjelmistojen yhteentoimivuutta säädellään palvelun tarjoajan ja käyttäjän välillä. Varsinaisissa juridisissa sopimuksissa näillä "sopimuksilla" on vastaava merkitys kuin vaatimus- ja muilla järjestelmädokumenteilla: ne kuvaavat hyväksymisedellytyksiä toimitettaville ohjelmistoille. Sinällään sekä rajapintadokumentaatio että palvelun laatuvaatimusten dokumentointi voivat olla osapuolia sitovia.

Palvelutasosopimus (SLA, Service-Level Agreement) on palvelun tarjoajan ja asiakkaan välinen sopimus, jossa kuvataan, mitkä vaatimukset palvelun tulee täyttää. Palvelutasosopimuksen ominaisuuksien tulee olla mitattavia, ymmärrettäviä ja merkityksellisiä. Tyypillisiä palvelutasosopimusten ominaisuuksia ovat mm. palvelun saatavuusprosentti, samanaikaisten käyttäjien palvelukyky, muutoksista ilmoittamistapa, käytön tilastointi sekä suorituskyky. Palvelutasovaatimuksia voidaan usein kohdistaa sekä teknisiin että organisaatioiden tarjoamiin palveluihin. Teknisistä palveluista voidaan säilyttää palvelutason kuvauksia palveluhakemistoissa, jossa myös muita palvelujen metatietoja säilytetään.

6.3.3 Tarjousten arviointi ja toimittajan valinta

Tarjottavan järjestelmän ja sen sovelluskomponenttien kattavuudesta on tarjouksesta saatava täsmällinen kuvaus (Kavén, Hartikainen 2001). Tarjouksen perusteella hankinnan tekijän on arvioitava, täyttävätkö saadut tarjoukset asetetut vaatimukset ja valittava toimittaja, jonka kanssa sopimus tehdään. Arviointi- ja valintaprosessiin kuuluu saatujen tarjousten tasapuolinen arviointi, mahdolliset vierailut, varsinainen valinta sekä tarkat sopimusneuvottelut.

Arvioinnissa on tavoitteena muodostaa näkemys siitä, miten tarjottu järjestelmä tukee asetettuja tavoitteita sekä nykyistä työskentelyä ja mitä uusia mahdollisuuksia se avaa toimintojen kehittämiseksi. Saavutettu lisäarvo on keskeisin tekijä ratkaisua tehtäessä. Arvioinnissa ratkaistaan, minkä verran painotetaan teknologian ominaisuuksia (kuten tehokkuus, käytettävyys, luotettavuus) ja työperäisiä tavoitteita ja päämääriä sekä toimintojen kehittämistä (Kavén, Hartikainen 2001). Hankinnasta päätävillä tulee olla myös yhteinen käsitys tuettavasta ydintoiminnasta ja siihen tarvittavasta arkkitehtuurista.

Tarjousten pohjalta tehtävää arviointia ja päätöksentekoa varten on hyvä laatia painotettu arviointikehikko. Arviointikriteerit voidaan antaa myös toimittajien käyttöön tarjouspyynnön yhteydessä, jolloin on heti nähtävissä päätökseen vaikuttavien tekijöiden painoarvo. Esimerkiksi Helsingin kaupungin terveystieteiden osaston hankittavien ohjelmistojen arviointimallissa vuodelta 2001 arvioitaviin 19 osa-alueeseen kuuluivat mm. jononhallinta, hoitajaksot/vuodeosastohoito, ajanvaraus jne. (Kavén, Hartikainen 2001). Kullakin osa-alueella oli useita arviointikohtia, joissa arvioitiin mm. eri toimintojen olemassaoloa ja helppokäyttöisyyttä.

Päätöksen tekemiseksi tarvitaan kuva lyhyen ja pitkän aikavälin kustannuksista, saavutettavasta hyödystä ja siitä kuinka nämä toteutuvat eri toimijoiden osalta. Kustannusarviot ja niiden muodostuminen, laadunvarmistustoimenpiteet sekä tarjouksessa esitetyt tekniset ja toiminnalliset lähestymistavat ovat erityishuomion kohteena arvioinnissa. Hankintaprosessin aikana syntyneet kysymykset on syytä esittää kirjallisesti ja myös vastaukset on saatava kirjallisesti väärinymmärrysten välttämiseksi. Hyvässä sopimuksessa osapuolet ymmärtävät toisiaan ja mahdollisesti epäselvyyksiä aiheuttavista asioista sovitaan jo alkuvaiheessa (Kavén, Hartikainen 2001, IEEE 1998).

Tarjousten saamisen jälkeen kannattaa neuvotella toimittajien kanssa vertailtavuuden parantamiseksi. Vertailtavuutta parantaa lisäksi se, että kaikilla toimittajilla on samat lähtötiedot ja kysymykset sekä se, että vastaukset ostajan esittämiin kysymyksiin annetaan kirjallisina. Tarjouksia käsiteltäessä myös hylkäysperusteet on tuotava selkeästi esille; esimerkiksi käytettäessä julkisissa hankinnoissa kokonaistaloudellista arviointitapaa on perusteet aina esitettävä jo tarjouspyynnössä. Oleellista on, että kaikkien kilpailussa mukana olevien tarjoajien on saatava samat mahdollisuudet ennen lopullista päätöksentekoa (Kavén, Hartikainen 2001).

Tietojärjestelmien arkkitehtuurin ja sovelluspalveluiden arviointia voidaan lähestyä toiminnallisten ja organisaation hyötyjen näkökulmasta arvioimalla mm. toiminnallista joustavuutta, sovellusten liitettävyttä, jo tehtyjen investointien hyödyntämistä, hankinta- ja integraatiokustannusten vaikutuksia, tietojärjestelmäympäristön vähittäistä kehittämistä, olemassa olevien sovellusten uudelleenkäyttöä jne. (ks. luku 6.1).

Osana sovellusta tarjottavat sovelluspalvelut ja -komponentit voivat olla varta vasten toteutettavia, erikseen maksettavia valmiita lisäpiirteitä, tai osa perussovelluskokonaisuutta. Vastuut on lisäksi määriteltävä erityisen selkeästi, mikäli osana hankittavaa sovellusta on muiden kuin varsinaisen sovellustoimittajan tekemiä palveluita tai komponentteja. Sovellus voi myös toimia muiden tarjoamien palvelujen tai rajapintojen hyödyntäjänä, jolloin on varmistettava mahdollisuuksien mukaan

siitä, että eri ohjelmistot toimivat yhdessä rajapinnan kautta. Tätä varten tarkistetaan, että sovelluksissa käytetään samoja rajapintamäärittelyjä ja niiden versioita sekä se, että tekniset ratkaisut ja mahdolliset valinnaiset tai lisäpiirteet ovat yhteensopivia. Myös ulkoisten osapuolten tarjoamia rajapintojen testauspalveluja tai niistä tuotettuja raportteja voidaan käyttää tai edellyttää, jos sellaisia on saatavilla.

Yleisen sopimusvalmistelun lisäksi tarvittavat sopimuksen seikat tarkennetaan ja lyödään lukkoon vasta toimittajan valinnan jälkeen. Sopijaosapuolilla on usein osapuolikohtaisia erityisehtoja sopimukseen. Sopimusneuvotteluissa tarkennetaan kehittämiseen, käyttöönottoon sekä tukeen ja ylläpitoon liittyvät kysymykset. Sopimuksen tarkennuksessa on kiinnitettävä huomiota molempien osapuolten etuihin. Lisäksi tarkassa sopimuksessa voi olla määritellyt minimi- ja maksimihinnat tai eri tarkistuspisteisiin sidotut maksut (IEEE 1998).

6.4 Toteutus, hyväksyminen ja seuranta

Kun hankintaan liittyvä sopimus on tehty, siirrytään sopimuksen täytäntöönpanoon. Tähän liittyen on seurattava sopimuksen velvoitteiden täyttymistä, sopimuksen mukaisten tuotosten ja palvelujen toimittamista sekä hyväksyttävä sopimuksen mukainen toimitus. Lisäksi sopimuksen mukaisten tuotteiden ja palvelujen käytöstä sekä hankintaprosessista kannattaa kerätä kokemuksia ja kehittää edelleen hankinta- ja ylläpitokäytäntöjä.

6.4.1 Toimituksen ohjaus

Hankinnan toimitusvaiheessa seurataan, toimitetaanko sovitut tuotokset (laitteet, ohjelmistot, dokumentit, palvelut jne.) aikataulussa ja sovitussa muodossa (IEEE 1998). Käyttöönottoa varten on luotava edellytykset organisaatioissa. Tätä helpottaa, mikäli käyttäjät ovat vaikuttaneet prosessiin jo vaatimusvaiheessa. Sopimukseen liittyen on hyvä olla määriteltynä yhteydenpitokäytännöt käyttöönotossa ja toteutuksessa ilmenneiden ongelmien ja epäselvyyksien selvittämiseen. Tarjousten arvioinnissa ja toimittajan valinnassa käytettyjä mittareita tulisi käyttää laadun ja edistymisen seuraamiseen, ja edistymisestä sekä kohdatuista ongelmista on syytä kerätä molemminpuolista palautetta.

6.4.2 Hyväksyminen ja käyttöönotto

Ennen ohjelmiston hyväksymistä on arvioitava ja testattava, vastaako ohjelmisto asetettuja vaatimuksia ja sopimusta. Ohjelmistohankinnan hyväksymisen yhteydessä on varmistuttava, että ohjelmiston toiminnallisuus, suorituskyky, luotettavuus, saatavuus, mukautettavuus, tukipalvelut, asennettavuus, käytettävyys, dokumentaatio ja kustannukset ovat sovitulla tasolla (IEEE 1998).

Toteutettua sovellusta verrataan hyväksymistestauksessa järjestelmän vaatimukseen. Jos vaatimuksia ja tavoitteita on tarkennettu hankintaprosessin aikana, on varmistuttava siitä, että hyväksymistestauksen kriteerit vastaavat sovitua tilannetta (IEEE 1998). Arvioinnissa ja testauksessa pyritään huomaamaan, poikkeako toteutettu ohjelmisto vaatimusmäärittelyistä, ja vastaavatko myös ohjelmiston ei-toiminnalliset ominaisuudet asetettuja tavoitteita.

Erityisen olennaista rajapintojen ja sovelluspalvelujen kannalta on testata järjestelmää yhdessä muiden samassa ympäristössä toimivien komponenttien kanssa liitettävyyden ja toimivuuden varmistamiseksi. Käyttöympäristössä testauksen lisäksi testaamista voidaan tehdä simuloitussa ympäris-

tössä. Hyväksymistestauksen tulisi olla osa hankinnan hyväksymiskriteerejä, ja ohjelmiston hankkijan edustajat voivat osallistua testaukseen myös muuten kuin omassa ympäristössään.

Ohjelmiston hyväksymisprosessiin voi kuulua laatu- ja ylläpitosuunnitelmien mukaisuuden varmistaminen ohjelmistossa (IEEE 1998). Myös ohjelmiston ulkoinen sertifiointi tai sertifiointivaatimusten katselmointi voi kuulua hyväksymisprosessiin. Sopimuksissa on usein lausemia, että maksu suoritetaan vasta sopimuksen lopussa tai kaikkien määriteltyjen tuotosten hyväksyttävän toimituksen jälkeen. Lisäksi sopimusten mukaisesti voidaan lykätä tai pienentää maksuja, jos eteneminen tai tuotokset eivät ole sovittuja.

Ohjelmiston hyväksymisen jälkeen seuraa käyttöönotto. Uuden tietojärjestelmän käyttöönotto voi isossa organisaatiossa olla vuosia kestävä hanke. Isot hankkeet on syytä pilkkoa osaprojekteihin, jollainen voi olla esimerkiksi käyttöönotto tietyssä yksikössä tai tietyn osion käyttöönotto. Käyttöönottoa voidaan myös vaiheistaa esimerkiksi siten, että ensimmäisessä vaiheessa uusi teknologia ja perustietotekniikka kehitetään ja "ajetaan sisään", toisessa vaiheessa toteutetaan pilottiprojekti ja vasta kolmantena otetaan tietojärjestelmä käyttöön koko organisaatiossa (Ripatti 1999). Lähtökohdana käyttöönottilanteen suunnittelulle on toimiva ja testattu tekninen käyttöympäristö sekä koulutettu ja osaava henkilökunta. Käyttöönotto suunnitellaan siten, että se häiritsee mahdollisimman vähän organisaation toimintaa ja esimerkiksi potilaspalvelua. Käyttöönotto etenee tehdyn tarkan suunnitelman mukaisesti ja alussa siihen voi kuulua vanhan ja uuden järjestelmän rinnakkaiskäyttöä. Käyttöönotossa on järjestettävä riittävä tehostettu käyttäjätuki, jota on tarvittaessa saatavissa vuorokauden ympäri.

Palveluarkkitehtuurin mukaisessa kehityksessä pyritään välttämään suuria ja toimintaa häiritseviä käyttöönottoja. Kokonaisjärjestelmää pyritään kehittämään pala kerrallaan siten, että uudet ohjelmistot sovitetaan yksitellen toimimaan muiden kanssa. Tämä aiheuttaa merkittäviä vaatimuksia organisaation sovellusarkkitehtuurille, käytetylle infrastruktuurille ja käyttöönoton toimintamalleille. Pitkälle kehittyneessä palvelupohjaisessa arkkitehtuurissa uusien piirteiden ja kokonaisuuksien lisääminen järjestelmään tapahtuu lisäämällä "täsmäpalveluja" toimivaan järjestelmäkokonaisuuteen. Muutoksia voidaan tehdä uusien palvelujen, prosessimäärittelyjen, koostettujen sovelluksien tai organisaation policy-määrittelyjen tasolla.

Käyttöönotto tarvitsee tuekseen organisatorisia ja työnjaollisia ratkaisuja (Hasu 2003). Uuden teknologian käyttöönotto häiritsee usein työhön liittyvää toimintaa ja edellyttää toimintojen yhteensovittamista. Tämä asettaa haasteita käyttöönoton johtamiselle. Käyttöönoton suunnittelun ja johtamisen kannalta on tärkeää tunnistaa, jos organisaatiossa on meneillään merkittäviä, tietojärjestelmän käyttöönotosta riippumattomia siirtymävaiheita, jotka voivat edistää tai haitata käyttöönottoa ja käytön vakiintumista. Tietojärjestelmän käyttöönotto muuttaa väistämässä joitakin toimintatapoja, mutta ei ole aina yksinkertaista tunnistaa kaikkia muutoksia tai varmistaa, että valtaosa muutoksista on halutun suuntaisia.

Käyttöönotto voidaan nähdä kriittisenä siirtymänä tuotekehittäjältä käyttäjälle, jolloin siirtyminen ei ole kertatapahtuma vaan tapahtuu vähitellen vuorovaikutuksessa tuotekehittäjien ja käyttäjien sekä muiden osapuolten kanssa (Hasu 2003). Käyttöönottoprosessi voi parhaimmillaan olla osapuolille uuden oppimista ja luoda yhteistyössä edellytyksiä innovaation vakiintumiselle ja leviämislle. Käyttöönotto voi myös merkitä organisaatioille ponnahduslautaa uuteen, kehittyneempään toimintatapaan. Lähestymistapa edellyttää huomion kiinnittämistä paikallisiin käyttöönottokokeiluihin ja niissä tapahtuvaan vuorovaikutukseen sekä oppimiseen.

6.4.3 Seurantavaihe

Ohjelmiston hankinnan ja käyttöönoton jälkeen sitä käytetään osana organisaation toimintaa. Hankinnasta ja käyttöönotosta saatuja myönteisiä ja kielteisiä kokemuksia voidaan hyödyntää hankintaprosessin edelleen kehittämisessä. Hankinnan ja käyttöönoton jälkeen on hyödyllistä tutkia ja arvioida hankintakäytäntöjä, käyttäjä- ja asiakastyytyvää sekä toimittajan suoriutumista (IEEE 1998). Hankintakäytännöistä voidaan tunnistaa kehityskohteita ja heikkouksia, joita on tarpeen kehittää edelleen, tai tunnistaa alueita, joille tarvitaan lisäohjeistusta tai uusia menettelytapoja. Toisaalta voidaan myös levittää ja vahvistaa hyviä tuloksia tuottaneita toimintoja. Käyttäjätyytyväisyyttä ohjelmistoihin voidaan tutkia heti käyttöönoton jälkeen ja jonkin ajan kuluttua käyttöönotosta. Se, saavutetaanko ohjelmistolla haluttuja pidemmän aikavälin tai strategisia vaikutuksia, selviää usein vasta kohtalaisen pitkän ajan kuluttua käyttöönotosta. Käyttöönoton yhteydessä tehtävillä arvioinneilla voidaan lähinnä varmistua ohjelmiston toimivuudesta. Lisäksi käyttötoiminnassa on syytä seurata tarvittavan ylläpitotyön määrää sekä esiintyvien virhetilanteiden ja löydettyjen ohjelmistovirheiden määrää ja vaikuttavuutta. Toimittajan suoriutumisen seurannasta kerättävät tiedot ovat käytettävissä esimerkiksi uusien hankintojen yhteydessä.

Sovelluspalvelujen käytön hyödyt muodostuvat vasta toistuvan käytön myötä. Vaatimusmäärittelyn, hankinnan ja käyttöönoton yhteydessä tunnistetut vaihtoehtoiset käyttökohteet muodostavat pohjan palvelujen uudelleenkäytölle, jolle on muodostettava edellytykset eri yksiköissä, sovelluksissa ja prosesseissa. Palvelupohjaisessa arkkitehtuurissa sovelluspalvelujen käyttöä on syytä seurata sekä teknisellä että toiminnallisella tasolla. Tekninen seuranta on tarpeen esim. palvelujen kuormituksen seuraamiseksi ja kokonaisjärjestelmän keskeisten ydinpalvelujen suorituskyvyn varmistamiseksi. Toiminnallisella tasolla voidaan seurata eri palvelujen ja prosessien käyttöä ja etenemistä sekä tunnistaa mahdollisia kehityskohteita (ks. luku 6.3).

6.5 Hankintojen tarkistuslistoja

Tässä luvussa esitetään tarkistuslistoja ohjelmistojen hankinnalle. Listat perustuvat valmiisiin tarkistuslistoihin (IEEE 1998, NIH 2006, Scott & Nisse 2004, Holley ym. 2006) sekä edellisissä luvuissa käsiteltyihin seikkoihin erityisesti sovelluspalveluaspektien ja rajapintojen osalta. Tarkistuslistat on kehitetty erityisesti sovelluspalveluihin ja rajapintoihin liittyviä seikkoja silmällä pitäen, ja niissä ei ole kaikkia esimerkiksi valmissovelluksia tai kokonaisjärjestelmiä koskevia asioita. Esimerkiksi käyttäjäkyselyn tarkistuslista on jätetty pois, koska kyselyjä tehdään vasta, kun ohjelmisto on ollut käytössä jonkin aikaa. Tarkistuslistoja voidaan käyttää ohjeellisesti edellä kuvatun prosessin eri vaiheissa, ja niistä on pyritty karsimaan päällekkäisyyksiä.

Hankintastrategian tarkistuslista:

- Mihin tarpeeseen hankinnalla pyritään vastaamaan? Kuinka hankinta palvelee ko. tarvetta?
- Onko tarve toteutettavissa jo hankittuja ohjelmistoja, toimintatapoja tai infrastruktuuria uudelleen käyttämällä, mukauttamalla tai muuttamalla?
- Onko tarve yleistettävissä siten, että hankinnalla voidaan palvella tulevaisuudessa myös muita kuin välitöntä tarvetta?
- Mitä ohjelmistoja ja muita teknisiä seikkoja (esim. laitteet) hankintaan kuuluu? Muodostavatko ne yhden vai useita kokonaisuuksia?
- Kuka tekee päätöksen hankinnasta?
- Mikä on hankinnan aikataulu? Kuinka sen toteutumista seurataan?

- Kuka (mikä toimintayksikkö) tulee omistamaan hankinnan kohteen, siihen liittyvät tiedot, toiminnot ja prosessit? Ketkä hankintaa tai siihen liittyviä tietoja ja toimintoja käyttävät?
- Mitkä ovat hankinnan kokonaiskustannukset ja miten ne muodostuvat ja sijoittuvat ajallisesti? Kuinka hankinnan kustannuksia seurataan ja budjetoidaan?
- Jos kyseessä on julkinen hankinta, ylittääkö hankinnan arvo laissa määriteltyt kynnsarvot, ja millaista hankintamenettelyä edellytetään?
- Miten hankinnan kustannukset ja tuotot kohdistetaan omistajien ja käyttäjien välillä? Miten päätetään ohjelmiston käyttäjäkunnan laajentamisesta ja uudelleenkäytöstä?
- Mitä palveluja (ylläpito, tuki, virheiden korjaus, tietoliikenne, käyttäjien ohjaus ym.) hankintaan kuuluu?
- Millaista hankintamenettelyä käytetään (esim. avoin tarjouskilpailu, tarjouspyyntö, puitesopimuksen ohjaama hankinta, hankintojen kynnsarvot)?
- Vastaako ylläpidosta ja tuesta toimittaja vai hankkija?
- Mitkä ovat hankinnan mahdolliset riskit? Miten toimitaan toimituksen tai käyttöönoton keskeytyessä tai viivästyessä?
- Mitä mittareita käytetään hankinnan tavoitteiden saavuttamisen mittaamiseksi?
- Mikä on hankittavan ohjelmiston luonne ja koko? Kuinka se liitetään osaksi organisaation tietojärjestelmäkokonaisuutta ja -arkkitehtuuria?
- Kuinka ohjelmistot ja palvelut ovat myöhemmin löydettävissä uudelleenkäyttöä varten?

Hankittavan ohjelmiston määrittelyn tarkistuslista:

- Minkä tyyppinen on hankittava ohjelmisto (esim. kokonaisjärjestelmä, erillisjärjestelmä, paikallisesti asennettava sovelluspalvelu, sovitin olemassa olevaan järjestelmään, laitteen mukana tuleva ohjelmisto jne.)?
- Kuuluuko hankintaan ohjelmistolisenssejä, ja onko lisenssiehdot tarkasti ja hyväksyttävästi määritelty?
- Kuuluuko hankintaan tietojen konversiota vanhoista järjestelmistä, kuka siitä vastaa?
- Jos hankinta perustuu aiempiin ohjelmistoihin (esim. sovittimet), onko niistä saatavilla riittävä dokumentaatio?
- Sisältyykö hankintaan ohjelmiston lähdekoodi?
- Millainen on ohjelmiston tarvitsema alusta ja suoritusympäristö? Kuka vastaa siitä? Tarvitaanko nykyiseen suoritusympäristöön lisäyksiä ja muutoksia, ja kuka niistä vastaa?
- Miten määritellään ohjelmiston käyttämät laitteet ja resurssit suhteessa asiakkaan ympäristöön?
- Mitä dokumentaatiota hankintaan sisältyy (käyttäjädokumentit, ylläpitodokumentit, asennusohjeet, esittelymateriaalia, muuta)? Onko dokumentaatio riittävää ja ajan tasalla? Tarvitaanko asiakaskohtaista dokumentaatiota ja kuka siitä vastaa? Onko dokumentaatio käytettävissä jo arviointivaiheessa?
- Millaiset versiopäivitykset ohjelmistoon ja sopimukseen sisältyy? Aiheutuuko versiovaihdoksista tai päivityksistä katkoksia ohjelmiston käyttöön ja kuinka niiden osalta toimitaan?
- Onko selvästi määritelty, millaiset lisäykset, muutostyöt tai versiot kuuluvat hankinta- ja ylläpitosisäilytyksen piiriin, ja tarvitaanko esim. tiettyjä uusia versioita varten erillisiä hankintoja?
- Kuinka ohjelmisto vastaa turvallisuusvaatimuksiin? Millaiset käyttäjähallinta-, tietojen suojaus-, käytön seuranta-, turvallisuusauditointi- ja lokimenetelmät ohjelmistossa on?
- Millaisia rajoitteita (käyttötarkoitus, kuormitus jne.) ohjelmiston käytölle on asetettu?
- Millaisia versioita ohjelmistosta on saatavilla? Mikä on sopiva ja arvioitava versio?
- Millaisia riippuvuuksia ja rajapintoja ohjelmistossa on muihin ohjelmistoihin? Ovatko rajapinnat yhteensopivia jo käytettyjen muiden ohjelmistojen kanssa? Miten varmistetaan ohjelmistojen toimiminen yhdessä hankinnan tai päivitysten yhteydessä?

- Mitkä ohjelmiston ominaisuudet, toiminnot ja tiedot on määriteltävä ja rakennettava yleiskäyttöisiksi ja uudelleenkäytettäviksi? Kuinka niitä voidaan hyödyntää myös muissa kuin alkuperäisessä käyttötarkoituksessa? Mitkä ovat tapauskohtaisesti mukautettavia ominaisuuksia?
- Mitkä ovat mahdollisia uusia sovelluksia, joissa ohjelmistoa voidaan käyttää nykyisen tarpeen lisäksi? Sisältääkö ohjelmisto ydinpalveluita, joita voidaan hyödyntää monissa prosesseissa?
- Mitkä osat tai ominaisuudet ohjelmistosta ovat päällekkäisiä jo organisaatiosta löytyvien ratkaisujen kanssa, ja miten ohjelmistossa pystytään hyödyntämään jo hankittuja ratkaisuja?

Hankkijan ja toimittajan vastuiden tarkistuslista:

- Onko tilaajan ja toimittajan vastuuhenkilöt hankintaan liittyen nimetty?
- Onko käyttäjä- tai ylläpitokoulutus tarpeen ja kuka siitä vastaa? Missä ympäristössä koulutus suoritetaan?
- Sisältyykö hankintaprosessiin esittelytilaisuuksia, tutustumista toimittajan toimintaan tai auditointia?
- Kuinka ohjelmisto arvioidaan hankintaprosessin aikana ja ennen hyväksymistä? Käytetäänkö arvioinnissa sertifiointi- tai testauspalveluja?
- Kuka vastaa tarvittavista ohjelmistoasennuksista käyttöönoton ja päivitysten yhteydessä? Millaista tukea asennuksiin on saatavilla?
- Kuka vastaa virhetilanteiden selvittämisestä ja korjaamisesta?
- Kuka vastaa tarvittavista muutoksista ohjelmistoon ja dokumentaatioon? Vaikuttavatko muutokset ohjelmiston takuuseen tai käyttöehtoihin?
- Miten todennetaan ohjelmiston vastaavuus erityyppisiin esitettyihin vaatimuksiin?
- Millainen ja kuinka pitkä takuu ohjelmistolle ja dokumentaatiolle myönnetään? Liittyykö takuuseen erikoisehtoja vai onko se ehdoton? Alkaako takuu läpi menneestä hyväksymistestauksesta? Kattaako takuu myös suorituskykytason määritellyssä ympäristössä?
- Onko hankintaan liittyen määritelty tuotokset ja toimituksen vaiheet aikatauluineen? Onko eri vaiheiden käynnistymis- ja päättymispisteet määritelty?
- Kuka vastaa käyttäjädokumentaation tuottamisesta, tulostamisesta ja jakelusta?
- Kuka vastaa hankinnasta tiedottamisesta?

Toimittajan arvioinnin tarkistuslista hankintaprosessin alkuvaiheeseen:

- Kuinka hyvin toimittaja on suoriutunut aiemmista sopimuksista ja toimituksista?
- Onko saatavilla tietoa ohjelmiston tai toimittajan aikaisemmilta asiakkailta tai toimituksista?
- Mitä tietoja toimittajan vakavaraisuudesta on saatavilla, esim. luottoluokitus, toimintahistoria?
- Kuinka laajat ja osaavat resurssit toimittajalla on liittyen toteutuksiin, käytön tukeen ja kyseessä olevan ohjelmiston kehittämiseen?
- Onko toimittajalla käytössä dokumentoidut ja riittävät kehitys-, ja ohjaus-, dokumentointi- ja testauskäytännöt? Kuka vastaa testauksessa käytetyistä tiedoista ja tietokannasta?
- Onko toimittajalla keinot varmistua vaatimusten jäljitettävyydestä ohjelmistossa, konfigurointien hallintajärjestelmä ja virhetilanteiden, kysymysten ja muutospyyntöjen hallintajärjestelmä?
- Millaiset vakuudet ja sopimukset toimittajalla on ohjelmiston ylläpitopalvelujen määrästä ja laadusta?
- Voidaanko ohjelmistoa kokeilla ja esitellä käyttöympäristössä ennen käyttöönottopäätöstä?

Laatu- ja ylläpitosuunnitelmien tarkistuslista:

- Kuinka ohjelmiston ja dokumentaation laatuvaatimukset on määritelty?
- Kuinka ohjelmiston testaus suoritetaan? Missä ympäristössä ja kenen toimesta?
- Millaisia muutoksia ohjelmistoon voidaan tehdä sopimuksen piirissä, ja mitkä on hoidettava erillisin sopimuksin?
- Millainen on ohjelmistosta löydettyjen virheiden ilmoittamis- ja korjausmenettely? Millaisia aikarajoja virheiden korjaukselle asetetaan? Ovatko korjaukset versio- tai asiakaskohtaisia?
- Käytetäänkö ohjelmiston arviointiin ja hyväksymiseen esittelyjä, käyttäjätutkimusta, testausta ja dokumentaation katselmointia? Kuka vastaa näistä toimenpiteistä (toimittaja, hankkija, kolmas osapuoli)?
- Millaista testidokumentaatiota tuotetaan tai vaaditaan: testaussuunnitelma, testausmenetelmät, testidata, testitulokset?
- Miten seurataan, mitkä käyttäjäryhmät ja mitkä muut ohjelmistot käyttävät ohjelmistoa?
- Onko ylläpito vaatimukset määritelty esimerkiksi dokumentaation ja teknisen tuen osalta?
- Kattaako tekninen tuki virheiden korjaukset, ohjelmistomuutokset, uudet versiot, dokumentaation päivitykset, asennustuen ja koulutuksen?
- Kuka vastaa teknisestä tuesta takuuajana ja sen jälkeen, ja kuka vastaa sen kustannuksista?
- Kuka saa muuttaa ohjelmistoa? Kuka hyväksyy muutokset? Kuinka varmistetaan, että ohjelmiston käyttäjät ja muut sitä hyödyntävät ohjelmistot toimivat oikein muutoksen jälkeen?
- Kuka päättää ohjelmiston uudelleenkäytöstä uusissa sovelluksissa, yksiköissä tai prosesseissa?
- Kuka vastaa muutosten rahoittamisesta ja tekemisestä, jos ohjelmiston käyttö lisääntyy ja tarvitaan esimerkiksi infrastruktuurin tai kapasiteetin lisäämistä?
- Millaisia varmistus- ja saatavuusvaatimuksia ohjelmistoon liittyy?

Sopimuksen ja maksujen tarkistuslista:

- Liittyykö hankintaprosessin aikana välitettäviin tietoihin salassapitoon liittyviä ohjeita tai vaatimuksia, ja onko ne määritelty sopimuksessa?
- Millaisia hinnoitteluperusteita ja -järjestelyjä ohjelmistoon liittyy? Miten ohjelmiston, dokumentaation ja tukipalvelujen kustannukset sovitaan?
- Sisältyykö sopimukseen ohjelmistomuutoksia, koulutusta, konversioita, asennuskustannuksia, käyttökokeiluja, käyttökustannuksia, neuvontapalveluja, ylläpitokustannuksia ja matkakustannuksia?
- Sisältyykö sopimukseen ohjelmiston lisäpiirteiden suunnittelua tai toteuttamista?
- Millaiset ovat lisenssimaksut, lisenssiehdot ja niiden uusimismenettelyt? Mitä ohjelmistoja ja palveluja lisenssimaksu sisältää?
- Seuraako hinnoittelu käyttäjien tai käytön määrää, ja kuinka tätä seurataan?
- Käytetäänkö ohjelmistossa vakiosopimuksia? Voidaanko sopimukseen tutustua ohjelmiston arvioinnin yhteydessä? Mistä sopimusehdoista voidaan neuvotella paikallisesti?
- Onko hankinnan eri osat hinnoiteltu, ja kullekin osalle määritelty täyttymiskriteerit ja aikataulut?
- Onko määritelty, kuinka aikataulussa pysymättömyys tai puuttuvat tai ylimääräiset tehtävät tai tuotokset vaikuttavat maksuihin?
- Liittyykö osa maksusta hankinnan lopulliseen hyväksymiseen, ja kuinka lopullinen hyväksyminen suoritetaan?
- Voidaanko sopimuksissa käyttää näiden tarkistuslistojen sisältöä vastauksineen?

Toimituksen etenemisen, toimittajan suoriutumisen ja toimituksen hyväksymisen tarkistuslista:

- Miten ohjelmiston vastaavuus vaatimuksiin varmennetaan: dokumentaation perusteella, esitelyn avulla, käyttäjäkyselyllä, testauksella? Kuinka varmistetaan toiminnallisten ja ei-toiminnallisten vaatimusten toteutuminen?
- Onko järjestelmän käyttöönotossa pysytty sovitussa aikatauluissa? Onko eri vaiheiden valmistuminen todettavissa?
- Miten todetaan hyväksymis- ja testauskriteerien mukainen toimitus?
- Noudatetaanko määriteltyjä kehitysmenetelmiä, dokumentointikäytäntöjä jne.?
- Ovatko ohjelmistosta löytyneet virheet ja käytön virhetilanteet löydetty, dokumentoitu ja korjattu? Vastaavatko korjaukset tarpeita?
- Onko ohjelmisto saatu toimimaan lopullisessa käyttöympäristössä muiden siihen liittyvien ohjelmistojen kanssa?
- Onko kaikki sopimuksen mukaiset tuotokset ja palvelut todistettavasti ja hyväksyttävästi tuotettu ja suoritettu?
- Onko käyttöönoton jälkeiset korjaukset ja toimenpiteet käynnistetty tai suoritettu?
- Onko saatu riittävä koulutus ja apu käyttöönotossa?
- Kerätäänkö käyttöönotosta palautetta järjestelmällisesti?

Ohjelmiston arvioinnin ja hyväksymisen tarkistuslista:

- Kattaako ohjelmiston toiminnallisuus asetetut vaatimukset? Vastaavatko toiminnalliset ominaisuudet ja tiedot sopimusta ja asiakkaan vaatimuksia?
- Toimiiko ohjelmisto riittävän tehokkaasti käyttöympäristön käyttäjämäärällä? Voidaanko ohjelmiston suorituskykyä seurata? Onko suorituskykyä testattu?
- Kuinka ohjelmiston luotettavuus todetaan? Onko sitä käytetty tai testattu riittävästi? Kuinka se on testattu ja ovatko testitulokset saatavilla? Miten ohjelmisto toipuu virhetilanteista?
- Kuinka ohjelmiston saatavuus todetaan? Onko ohjelmistossa kuormantasausominaisuuksia? Kuinka toimitaan tilanteissa, joissa ohjelmisto ei ole saatavilla?
- Voidaanko ohjelmistoa mukauttaa tai muokata sovitusti? Onko ohjelmisto käytettävissä määritellyissä käyttötilanteissa? Onko ohjelmisto käytettävissä muuttuneissa tai uusissa käyttötilanteissa, ja millaisin toimenpitein?
- Mitä ohjelmiston ominaisuuksia tai rajapintoja voidaan jakaa ja hyödyntää myös muissa kuin välittömään tarpeeseen vastaavissa sovelluksissa ja prosesseissa? Millaisia reunaehtoja uudelleenkäytölle on?
- Miten ohjelmiston ylläpito, päivitykset, muutokset ja lisäykset hoidetaan? Sisältääkö sopimus ylläpitotehtäviä?
- Mitä toimenpiteitä ohjelmistoasennus vaatii? Tarvitaanko siihen koulutusta? Liittyykö käyttöönottoon konversioita ja kuka ne suorittaa?
- Onko ohjelmiston käytettävyyden riittävä hyvä? Onko käytön ohjeistus riittävää?
- Kuinka ohjelmisto toimii yhdessä muiden ohjelmistojen kanssa? Toimivatko rajapinnat määrittelyjen mukaisesti ja yhdessä muiden käytettävien ohjelmistojen kanssa?
- Onko ohjelmiston dokumentaatio riittävää ja ymmärrettävää?
- Mitkä ovat hankinnan toteutuneet kustannukset ja ovatko ne sopimuksen mukaisia? Ovatko suorat ja epäsuorat kustannukset jäljitettävissä?

Hankintaprosessin lisäksi tarkistuslistojen sisältämiä seikkoja on syytä koota, kysellä ja arvioida käyttöönoton jälkeen. Monet tarkistuslistojen kohdista ovat yleisiä ja sovellettavissa erityyppisissä ohjelmistohankinnoissa. Hankinnan luonteesta ja määritellyistä vastuista riippuu, mitkä tarkistuslistojen asioista ovat kussakin projektissa tarpeen.

6.6 Yhteenveto ja johtopäätökset hankinnoista

Tietojärjestelmähankintojen perustana on oltava pitkäjänteinen suunnitelmallinen toiminnan kehittäminen ja tarkoitus mahdollistaa joustava tiedon tuottaminen, saanti ja käyttö. Hankintaprosessiin liittyvä työ on usein pitkän suunnittelun tulosta. Työtä ohjaavat paikallisten tarpeiden lisäksi myös alueelliset, valtakunnalliset ja kansainväliset ratkaisut ja standardointi. Tietojärjestelmähankinta koskee usein myös suurta osaa henkilökunnasta. Arviointia hankintojen yhteydessä tulee suorittaa eri näkökulmista, kuten alue- ja organisaationäkökulmista sekä henkilöstön, asiakkaiden ja taloudellisesta näkökulmasta. Kullekin osa-alueelle tulee määritellä tarkemmat kriteerit (EPPU-projekti, 1998). Tavoiteltujen hyötyjen saavuttamiseksi on uudistettava organisaation toimintamalleja haluttujen tavoitteiden edellyttämällä tavalla, ja ohjelmistot muodostavat vain osan ratkaisusta. Tietojärjestelmien hankinnan ja käyttöönoton sekä toimintamallien uudistamisen tulee pohjautua organisaatioiden visioihin ja strategioihin.

Onnistuneiden hankintojen perusedellytyksiä ovat (IEEE 1998):

- hankinnan tavoitteiden saavuttamiseen tarvittavien laatuominaisuuksien tunnistaminen ja määrittely,
- laatuominaisuuksien huomiointi hankintojen suunnittelussa, arvioinnissa ja hyväksynnässä,
- hankintastrategian kehittäminen,
- hankintaprosessin kehittäminen ja sen käyttöönotto.

Kuten aiemmista luvuista käy ilmi, hankintoihin liittyy suuri joukko tekijöitä ja tehtäviä. Kaikkien tarvittavien seikkojen huomiointi ja ratkaiseminen hankinta- tai päivityskohtaisesti vie runsaasti aikaa ja resursseja. Onkin syytä valita rajattu ja selvä joukko hankintatapoja ja määritellä selkeät ja toistettavat menettelyt valittujen hankintatapojen toteuttamiseksi. Lisäksi sopimuskäytäntöjen ja puitesopimusten avulla voidaan tehostaa hankintoja ja niihin liittyviä menettelyjä. On myös järkevää määritellä sopimuksiin riittävän laajoja kokonaisuuksia, jotta kaikki pienetkin lisäykset ja muutokset eivät vaadi raskasta hankinta-, kilpailuttamis- ja sopimusprosessia.

Hankinnoissa tulee tarkastella saavutettavaa kokonaisuhyötyä halvimman tarjouksen tai pelkkien kustannussäästöjen sijaan. Erityisesti tietojärjestelmien toimivuudelle asetettuja laadullisia tavoitteita on vaikeaa arvioida hankintojen yhteydessä. Tietojärjestelmän tai sovelluspalvelun sisällöllinen sopivuus käyttötilanteeseen on yleensä arvioitavissa määrittelydokumenttien perusteella, mutta suorituskyvyn, luotettavuuden ja joustavuuden arviointiin tarvitaan yleensä testausta ja kokeiluja. Järjestelmän arviointi ja sopivuus käyttäjille sekä se, miten turvalliseksi, luotettavaksi ja miellyttäväksi järjestelmä koetaan, selviää lopullisesti vasta käyttöönoton yhteydessä. Koska ohjelmiston ylläpidosta ja tukipalveluista muodostuu ajan mittaan merkittävin kustannuserä, on arviointi- ja sopimusvaiheessa kiinnitettävä erityistä huomiota niihin liittyviin palveluihin ja dokumentaatioon. Jatkuvan ylläpidon selkeä erottaminen hankinnasta on tarpeen myös versiohallinnan, muutostöiden ja lisäpiirteiden, korjausten ja tukipalvelutarpeiden muutosten vuoksi.

Palvelu- ja rajapintapohjaisen kehityksen kannalta tekniset ratkaisut mahdollistavat uudenlaisia tapoja tietojärjestelmäkokonaisuuden kehittämiseen, mutta niihin liittyvät sopimusten ja hankintojen mallit eivät ole täysin vakiintuneet. Myös palvelupohjaisen järjestelmän kehittämisessä on otettava huomioon hankintoja koskevat lait, ohjeet ja menettelytavat. Käytännössä hankintoja ohjaava lainsäädäntö ja oikeusturvan toteuttaminen estävät esimerkiksi dynaamisen tiettyyn tilanteeseen sopivi-

en palvelujen etsimisen verkosta, joka olisi teknisesti mahdollista. Hankintanäkökulmaa on käsitelty erittäin vähän SOA-kirjallisuudessa, mikä osaltaan selittää sitä, että palvelupohjaista arkkitehtuuria on toteutettu lähinnä organisaatioiden sisäisessä kehitystyössä. Ohjelmistohankintojen standardeissa (esim. IEEE 1998) käyttöönottoa käsitellään varsin vähän. Vaikka standardissa kuvataan hankintaa etupäässä asiakasorganisaation näkökulmasta, se keskittyy vahvasti ohjelmistoaspekteihin, ja käsittelee toiminnan kehittämistä ja loppukäyttäjien osallistumista hankinnoissa pinnallisesti.

Verrattuna kokonaisjärjestelmien ja valmisohjelmistojen hankintaan palvelupohjaisessa järjestelmässä korostuu arkkitehtuurin määrittely ja hallinta. Arkkitehtuurin määrittely ja ohjelmistojen sekä palveluiden arkkitehtuuriin sopivuuden varmistaminen ovat avaintehtäviä, kun kokonaisarkkitehtuuria kehitetään palvelupohjaiseksi ja modulaariseksi. Tämä aiheuttaa uusia tehtäviä hankintoihin ja ohjelmistojen arviointiin liittyen, mutta voi myös vähentää tarvittavan hallinnollisen työn määrää sikäli, kun hankinnat ovat entistä pienempiä ja niitä voidaan tehdä puitesopimusten piirissä. Avointen määritysten, rajapintastandardien ja integrointiprofiilien hyödyntäminen hankinnoissa parantaa yhteensopivuutta muiden käytettävien ohjelmistojen kanssa, vähentää paikallista räätälöintiä ja voi parantaa hankittavien ohjelmistojen luotettavuutta. Avointen määritysten osalta on kuitenkin arvioitava, mihin saakka niillä pystytään vastaamaan paikallisiin tarpeisiin, ja miltä osin tarvitaan tarkennettuja määrittelyjä. Sovelluspalvelujen hankinnassa on kiinnitettävä huomiota uudelleenkäytettävyyteen erityisesti uudelleenkäytettävien ydinpalvelujen sekä käytettävän teknisen infrastruktuurin osalta. Tällöin on välttämätöntä yleistää "perustoiminnallisuus" ja laajentaa sitä palveluja mukauttamalla tai esim. versioimalla. Käytettävien sovittamismenetelmien valinta kuuluu olennaisena osana hankintastrategiaan. Yleistäminen ja uudelleenkäytettävyys ja niitä tukevan infrastruktuurin kehittäminen tyypillisesti lisäävät kustannuksia ensimmäisissä hankinnoissa. Ne kuitenkin tarjoavat huomattavia säästöjä myöhemmissä projekteissa, kun hankittuja sovelluspalveluja käytetään uudelleen. Palvelupohjaisten sovellusten hyödyntämiseen tarvittava tekninen infrastruktuuri taas on yhä laajemmin osana muutenkin käytettäviä palvelimia, käyttöjärjestelmiä ja kehitysvälineitä (Wilkes 2003).

Toinen merkittävä erityispiirre palvelupohjaisten järjestelmien kehittämisessä ja hankinnassa on se, että hankintojen yhteydessä ei huomioida pelkästään jo tiedossa olevia vaatimuksia, vaan uudelleenkäytön ja ohjelmiston ennakoimattomien käyttötapojen tarpeita (Sprott 2004). Tämä vaatii yksittäistä käyttöönottoprojektia laajempaa vaatimusten hallintaa ja yleistämistä. Sovelluspalvelujen ja rajapintojen hankinnassa onkin järkevää yhdistää "palveluvaraston" kokoaminen yksittäisten projektien ja tarkkojen tarpeiden mukaisiin hankintoihin. Ensisijainen vaihtoehto ei olekaan uuden järjestelmän hankinta, vaan jo olemassa ja käytössä olevien tietotekniikkaratkaisujen ja sovelluspalvelujen uudelleenkäyttö ja -yhdistely. Tähän seikkaan liittyviä kysymyksiä käsitellään myös mm. luvussa 6.2.

Palvelupohjainen arkkitehtuuriajattelu muuttaa hankintoja. Tarpeen tunnistamisen jälkeen ensimmäinen kysymys on oltava, onko tarve toteutettavissa jo käytettävissä olevien ohjelmistojen ja palvelujen avulla. Verrattaessa esimerkiksi vuosituhaten vaihteessa kehitettyjä hankintojen standardeja ja uutta SOA-kirjallisuutta on nähtävissä siirtymä organisaation tietotekniikkakysymyksistä kohti toiminnallisia ja vastuukysymyksiä. Kehitys heijastaa pyrkimystä tietotekniikan ja toiminnan lähentämiseen. Hankinnoissa ja kehityksessä on kuitenkin muistettava myös ohjelmistoista ja tietotekniikkahankinnoista opitut seikat, kuten käytettävyysvaatimukset ja jäljitettävyys, jotka voivat unohtua pelkästään liiketoimintanäkökulmaan keskityttäessä. Palvelupohjainen ajattelu on yhdistettävä perinteisiin hankintatapoihin vähitellen myös organisaatioiden siirtymän helpottamiseksi.

Hankintakäytäntöjen kehittäminen siten, että tietotekniikka ja toiminta ovat lähempänä toisiaan vaatii monien esteiden poistamista: yhtenäisten päätösten tekemiselle tarkkaan tietoon perustuen on toimintatapoihin liittyviä esteitä, toiminnasta ja sen kehittämisestä vastaavat tahot on saatava osal-

listumaan sekä lyhyellä että pitkällä tähtäimellä kehitystyöhön, ja tietotekniikkahankkeiden seurantaan ja arviointiin tarvitaan järjestelmällisyyttä (Holley ym. 2006). Hankintaprosessien mallit ja tarkistuslistat ovat yksi tapa lisätä tarkkuutta ja järjestelmällisyyttä hankinnoissa.

7 Yhteenveto

Palveluarkkitehtuurin yleisten tavoitteiden osiin purku, mittaaminen ja arviointi on mahdollista ja ehdottoman tarpeellista, jos tavoitteiden saavuttamisesta halutaan varmistua. Mittaamisessa on pysyttävä valitsemaan toisiaan täydentäviä mittareita, jotka mittaavat mahdollisimman suoraan tavoitteiden toteutumiseen vaikuttavia tekijöitä. Sekä käyttäjäorganisaation, sovelluskehityksen että teknisellä tasolla on saavutettavissa merkittäviä hyötyjä avointen tekniikoiden, uudelleenkäytön sekä rajapinta- ja sopimuskeskeisyyden kautta.

Arkkitehtuuri on keskeinen, mutta suhteellisen näkymätön mahdollistaja toimintaa tukevien IT-ratkaisujen kehittämiseksi. Näkymättömyyttä pyritään vähentämään taloudellisten ja muiden hyötyjen osoittamisella, mutta tärkeä osa hyötyjen ja tarpeiden täyttämisen lisäksi on riskien välttäminen. Hyötyjen saavuttaminen voidaan osoittaa mittareiden avulla, ja viime kädessä mittauksen kohteena tulee olla organisaation omien tavoitteiden toteutuminen. Eri hyödyt realisoituvat kuitenkin siirtymän eri vaiheissa: alussa projektitasolla ja myöhemmin uudelleenkäytön, yhtenäistämisen ja joustavuuden kautta.

Hallinta (governance) on nousemassa yhä tärkeämmäksi näkökulmaksi organisaatioiden kokonaisarkkitehtuurien kehittämisessä. Monet tämän dokumentin osista liittyvät nimenomaisesti palveluarkkitehtuurin hallintaan (SOA governance), joka edellyttää tavoitteiden selkeää määrittelyä ja mittaamista. Kuten monet rajakäsitteet, myös palveluarkkitehtuurin hallinta nähdään helposti "uutena" toiminta-alueena. Olemassa olevat tietojärjestelmien hallinta- arviointi- ja kehittämisperiaatteet tarjoavat kuitenkin pohjaa myös palveluarkkitehtuuriin liittyvälle hallinnalle, ja palveluarkkitehtuurin erityisominaisuudet huomioivat lähestymistavat on syytä kehittää niiden pohjalta.

Myös hankintatapojen muuttaminen on välttämätöntä siirryttäessä kohti palvelupohjaista toimintatapaa. Uudelleenkäyttömahdollisuudet ja arkkitehtuuri projektikohtaisten tarpeiden lisäksi on otettava keskeiseksi kriteeriksi hankinnoissa. Hankintoja koskevissa ohjeissa ja säädöksissä on monia seikkoja, joiden sovittaminen palvelu- ja integraatiokeskeiseen toimintatapaan on haasteellista.

Dokumentin otsikko on sikäli harhaanjohtava, että useimmat tämän dokumenttiin kuvatuista malleista ja seikoista eivät ole terveydenhuoltospesifejä. Mukana olevat seikat on kuitenkin valittu terveydenhuoltospesifiä soveltamista silmällä pitäen SerAPI-hankkeen osapuolten osoittamien tarpeiden mukaisesti. Nämä tarpeet liittyvät ensisijaisesti sairaanhoitopiirien ja sairaaloiden toiminnan tukemiseen ja kehittämiseen tietotekniikan ja ohjelmistotuotteiden avulla. Terveydenhuollossa palveluarkkitehtuurilla haetaan samoja hyötyjä kuin muillakin toimialoilla, mutta erityisesti korostuvia tarpeita ovat päällekkäisten tietojen syötön ja ylläpidon vähentäminen, suurten kertakäyttönottojen välttäminen ja siirtyminen vähittäiseen järjestelmien kehittämiseen, nopeampi korjausten tai uusien piirteiden saanti järjestelmiin, ja viime kädessä tarkemmin paikallisiin tarpeisiin vastaavat järjestelmäratkaisut. Tietohallinnon kannalta siirtymän mukana tulee tarve lisähallinnalle: muutokset, hankinnat ja kehittäminen on rajattava entistä pienempiin osiin, valmiiden sovellusten ja ratkaisujen liittämiseen ja uudelleenkäyttöön on määriteltävä pelisäännöt, migraatiotavoitteet välietappeineen on määriteltävä, ja panostettava infrastruktuurin, sovellusten ja toiminnan hallintaan ja mittaukseen.

SerAPI-projektin keskeisiä tavoitteita ovat olleet rajapintaratkaisut ja sovelluspalvelut, jotka pohjautuvat terveydenhuollon ammattilaisten ja tietohallinnon tarpeisiin. SerAPI:ssa kehitetyt ratkaisut täydentävät esimerkiksi kansallisia arkisto- ja reseptiratkaisuja, ja niitä on liitetty potilasjärjestelmiin, alueellisiin tietojärjestelmiin sekä hoidon ja hallinnon erillisjärjestelmiin. Rajapintojen, tuotteiden ja arkkitehtuurin kehityksen ohjaukseen ja mittaukseen löytyy valmiita malleja, joita on dokumentoitu tähän oppaaseen. Hankkeen esimerkkitoimitukset ja konkreettisiin integrointi- ja kehi-

tystarpeisiin vastaavat määrittelyt tarjoavat konkreettisia esimerkkejä joidenkin tässä dokumentissa kuvattujen tavoitteiden toteuttamisesta ja hyötyjen saavuttamisesta, mutta monia tavoitelluista hyödyistä ei ole päästy validoimaan SerAPI-hankkeessa. Useissa muissakin maissa ja organisaatioissa on käynnissä samantyyppistä kehitystä, ja tätä kehitystä on seurattu etenkin kansainvälisten standardien kehittämisen kautta, tuottaen tietoa ja selvityksiä myös kansallisten ratkaisujen kehitykseen.

Eri kehitysstrategioiden ja ratkaisujen laatuun vaikuttavien tekijöiden tunnistaminen ohjaa keskeisesti tavoitteiden pohjalta tehtävää kehitystoimintaa. Tämän oppaan toisessa osassa kuvataan tarkemmin malleja ja kehitysprosesseja, joilla palvelupohjaisia ratkaisuja ja arkkitehtuureja kehitetään. Se myös sisältää esimerkkejä prosessimallinnuksesta ja prosessilähtöisestä palvelujen tunnistamisesta, joka on yhdistettävä tässä dokumentissa kuvattuihin hankinta-, arviointi- ja mittaustehtäviin. Monia tässä dokumentissa lyhyesti tai pinnallisesti mainittuja seikkoja käsitellään tarkemmin osassa 2. Sen yhteenvedossa myös kuvataan tarkemmin tilannetta palveluarkkitehtuurin hyödyntämisen suhteen terveydenhuollossa.

Lähteet

Arsanjani A. Service-oriented modeling and architecture: How to identify, specify, and realize services for your SOA. IBM Developerworks, 2004, viitattu 30.8.2006, <http://www-128.ibm.com/developerworks/webservices/library/ws-soa-design1/>.

Asetus 1416/1993. Asetus valtion hankinnoista, 1993.

Bernard T, Gallagher B, Bate R, Wilson H. CMMI Acquisition Module (CMMI-AM), Version 1.1. Software Engineering Process Management Acquisition Support Program, Technical Report, CMU/SEI-2005-TR-011, ESC-TR-2005-011; Carnegie-Mellon University, 2005.

Campbell EM, Sittig DF, Ash JS, Guappone KP, Dykstra RH. Types of unintended consequences related to computerized provider order entry. *J Am Med Inform Assoc* 2006; 13(5):547–556.

Carney D, Fisher D, Morris E, Place P. Some Current Approaches to Interoperability. Integration of Software-Intensive Systems Initiative, Technical note CMU/SEI-2005-TN-033, August 2005.

Central Computer & Telecommunications Agency 2002. Service Support, IT Infrastructure Library Series. Central Computer & Telecommunications Agency, 2002.

Chen H-T. Theory-Driven Evaluations. Sage Publications Inc., Newbury Park, California, 1990.

Cooper J, Fisher M (eds.). Software Acquisition Capability Maturity Model (SA-CMM), Technical Report, version 1.0. Carnegie Mellon University, 2002.

Cronholm S, Goldkuhl G. Strategies for Information Systems Evaluation- Six Generic Types. *Electronic Journal of Information Systems Evaluation* Volume 6 Issue 2, 65-74, 2003.

DeLone WH, McLean ER. The DeLone and McLean Model of Information Systems Success: A Ten-Year Update. *Journal of Management Information Systems*, Vol. 19, No. 4, 9-30, 2003.

Doll WJ, Torkzadeh G. The Measurement of End-User Computing Satisfaction. *MIS Quarterly*, Vol.12, Issue 2, 258-274, 1988.

EPPU -projekti 1998. EPPU-projekti, erikoissairaanhoidon ja perusterveydenhuollon potilastietojärjestelmien uudistaminen sairaanhoitopiirien yhteistyönä loppuraportti, versio 1.2. Keski-Suomen sairaanhoitopiiri, Kymenlaakson sairaanhoitopiiri, Satakunnan sairaanhoitopiiri, Uudenmaan sairaanhoitopiiri, Vantaan sosiaali- ja terveystieteiden tutkimuskeskus, 31.12.1998.

Etezadi-Amoli J, Farhoomand AF. A structural model of end user computing satisfaction and user performance. *Information & Management* 30, 65-73, 1996.

Hankinnat 2007. Julkiset hankinnat. [Hankinnat.fi](http://www.hankinnat.fi), Kuntaliiton ja KTM:n Julkisten hankintojen neuvontayksikkö, 2007, viitattu 21.8.2006, <http://www.hankinnat.fi/>

Hasu M. Käyttöönotto: kriittinen siirtymä perustutkimuksesta erikoissairaanhoidon - MEG-aivojen kuvannuslaite. Teoksessa: Miettinen R, Hyysalo S, Lehenkari J ja Hasu M. Tuotteesta työvälineeksi? Uudet teknologiat terveydenhuollossa. 2003.

Holley K, Palistrant J, Graham S. Effective SOA Governance. White paper, IBM, March 2006.

Honey A. SOA for HL7v3 - full version. Healthcare Services Specification Project, Health Level Seven & Object Management Group, 2006.

IEEE 1998. IEEE Recommended Practice for Software Acquisition. IEEE Std 1062, 1998 Edition. Software Engineering Standards Committee, IEEE Computer Society, 1998.

Igbaria M., Tan M. The consequences of information technology acceptance on subsequent individual performance. *Information & Management* 32, 113–121, 1997.

ISO/IEC 2001. Software engineering – Product quality – Part 1: Quality model. International Organization for Standardization and International Electrotechnical Commission, ISO/IEC 9126-1:2001, Geneva, 2001.

Juutilainen J, Tuomisto J. Ympäristöterveys. Kuopion yliopisto, 2002, haettu 12.2.2007: http://www.uku.fi/avoin/ymp_terveys/oppimateriaali/ymp_tulostus2.pdf

Kavén P, Hartikainen K. Tietojärjestelmähankinnat terveyden- ja sosiaalihuollossa ostajan näkökulmasta. Suomen Kuntaliitto, Helsinki, 2001.

KTI 2006. Kiinteistötalouden ja kiinteistöjohtamisen keskeiset käsitteet: Rahoituskäsitteet. (Kiinteistötalouden instituutti ry., viitattu 13.6.2006, http://www.kti.fi/sanasto.php?sivu=_102.)

Laki 2007/348. Laki julkisista hankinnoista 2007/348.

Lund MS, den Braber F, Stølen K, Vraalsen F. A UML profile for the identification and analysis of security risks during structured brainstorming, Sintef ICT Cooperative and Trusted Systems, STF40 A03067-Unrestricted Report, 2004.

Marquis H. How to measure it service quality. DITY Newsletter, itSM Solutions, 2006, viitattu 26.2.2007: <http://www.itmsolutions.com/newsletters/DITYvol2iss6.htm>

Mykkänen J., Porrasmäe J., Rannanheimo J., Tikkanen T., Sormunen M., Korpela M., Häyrynen K., Eerola A., Häkkinen H., Toivanen M. Terveydenhuollon sovellusintegraatoratkaisujen määrittely, 2004. Plugit-hanke selvityksiä ja raportteja 4.

NIH 2006. CIT IT Acquisition Documentation Checklist. Center for Information Technology, National Institutes of Health, 2006. viitattu 13.6.2006, <http://enterprisearchitecture.nih.gov/YourPart/File/CITAcquisitionChecklist.htm>

Nykänen P. 2005, Tietojärjestelmien kehittämis- ja käyttöönottoprosessi. Viitattu 21.2.2006: http://www.med.utu.fi/hoitotiede/perusopiskelu/materiaalipankki/Turku_101105-osa2_pirkkonykanen.ppt

OGC 2002. Applications Management, IT Infrastructure Library Series. Office of Government Commerce, 2002.

OGC 2004. Service Delivery, IT Infrastructure Library Series. Office of Government Commerce, 2004.

Øvretveit J. Evaluating Health Interventions. Open University Press, McGraw-Hill, Berkshire, 2003.

Perepletchikov M, Ryan C, Tari Z. The Impact of Software Development Strategies on Project and Structural Software Attributes in SOA. RMIT University School of Computer Science and Information Technology, 2005.

Remes LM. Hankintatoimen kehittäminen julkisessa terveydenhuollossa. Teknillinen Korkeakoulu, Tuotantotalouden osasto, Yritysstrategia ja kansainvälinen toiminta. Diplomi-insinööri opinnäyte-työ, Espoo, 2005.

Ripatti S. Tietojärjestelmän käyttöönotto. Teoksessa: Saranto K, Korpela M. Tietotekniikka ja tiedonhallinta sosiaali- ja terveydenhuollossa. WSOY, 1999.

Rissa M. IT-investoinnin kannattavuuden arviointi - ROI ei ole aina oikea mittari. 2004. Viitattu 13.6.2006: <http://www.rissa.fi/RC/roifi.htm>.

Samtani G, Sadhwani D. Web Services Return on Investment: Working out what you're Getting out of Web Services. Web Services Architect, August 14, 2002.

Saranummi N. Healthcare Technology Assessment and Evaluation. VTT Information Technology, 2003.

Saranummi N. SI-projektin vaiheet. Järjestelmäintegraatio-opas, VTT Information Technology, 2003.

Schmelzer R. The ROI of SOA. Zapthink LLC, 2005, viitattu 21.2.2006: <http://www.zapthink.com/report.html?id=ZAPFLASH-20050127>

Scott JA, Nisse D (eds.). Software Configuration Management. In: Guide to the Software Engineering Body of Knowledge, 2004 version. A project of the IEEE Computer Society Professional Practices Committee, IEEE Computer Society, 2004, pp. 7-1 - 7-17.

Sprott D, Wilkes L, Veryard R, Stephenson J. Web Services Roadmap: A CDBI Report Series-Guiding the Transition to Web Services and SOA. CDBI, 2003.

Sprott D. Best Practice Report - The Business Case for SOA. CDBI Journal, June 2006.

Talmon J, Enning J, Castañeda G, Eurlings F, Hoyer D, Nykänen P, Sanz F, Thayer C, Vissers M. The VATAM guidelines. International Journal of Medical Informatics 56(1-3), 107-115, 1999.

Teo TSH, Wong PK. An Empirical Study of the Performance Impact of Computerization in the Retail Industry. Omega, Vol.26, No. 5, 611–621,1998.

Torkzadeh G, Doll WJ. The development of a tool for measuring the perceived impact of information technology on work. Omega, Int J Mgmt Sci 27, 327–339, 1999.

Turunen P. Tietojärjestelmien arviointimenetelmien valinta terveydenhuolto-organisaatiossa - si-dosryhmänäkökulma. Turun kauppakorkeakoulun julkaisuja Sarja A-5:2001.

Webmethods 2006. Business Activity Monitoring (BAM) - The New Face of BPM. Whitepaper, WebMethods, Inc., June 2006, viitattu 1.6.2007: http://www1.webmethods.com/PDF/whitepapers/BAM-The_New_Face_of_BPM.pdf

Wilkes L. ROI - The Costs and Benefits of Web Services and Service Oriented Architecture. CDBi forum, 2003.

WS-I 2006a. Basic Profile Working Group. WS-I Basic Profile. WS-I, 2006, viitattu 21.8.2006, <http://www.ws-i.org/deliverables/workinggroup.aspx?wg=basicprofile>.

WS-I 2006b. Basic Security Profile Working Group. WS-I Basic Security Profile. WS-I, 2006, Viitattu 21.8.2006, <http://www.ws-i.org/deliverables/workinggroup.aspx?wg=basicsecurity>.