

Tietojärjestelmien integrointi -yhteenveto

Leila Korvenkoivu

Virpi Rautiainen

Janne Nurmi

Markku Mäkinen

Pasi Kaunisvesi

Sonja Väänänen

Raportti

28.4.2008

Sisällys

1	Johdanto.....	4
2	Service Oriented Architecture.....	5
2.1	Mitä SOA on?	5
2.2	Palvelukeskeisen arkkitehtuurin taustaa.....	7
2.3	Palvelukeskeiseen arkkitehtuuriin liittyvät ohjelmistot	9
3	SOA Management	13
3.1	Hallittavien resurssien yksilöinti (Identify resources to manage).....	15
3.2	Miksi hallinta korostuu juuri SOAssa?	17
3.3	Miten SOA Management huomioidaan SOA projektissa?.....	18
4	Onko SOA:n käyttöönotossa ongelmia?	20
4.1	Ongelmakohdat	20
4.2	Palvelut ja niiden tunnistaminen ja kuvaaminen.....	21
4.2.1	Palvelujen määrittely	21
4.2.2	Palvelujen tunnistaminen	24
4.2.3	Palvelujen luokittelu	26
4.2.4	Palvelujen dokumentointi ja kuvaaminen	27
4.3	Palvelun hallinta ja ylläpito.....	28
4.3.1	Automatisoitu ja huoltovapaa integraatio	28
4.3.2	Vikasietoisuuden varmistaminen.....	29
4.3.3	Integroinnin valvottavuus	30
4.3.4	Laajennettavuus ja muutoksen hallinta	30
4.3.5	Tietoturva ja käyttäjien hallinta	30
4.4	Saatavuus ja suorituskyky.....	31
4.5	Uudet teknologiat.....	32
4.6	Liiketoimintahyötyjen arviointia.....	32
4.7	Johdon sitoutuminen	33
4.8	Tietoturva	33
4.8.1	Web Service tietoturva.....	33
4.8.2	Sovellusintegraatio ja siihen liittyvät tietoturvatekijät	34
4.8.3	Tietoturvan varmistaminen Web sovellusten osalta.....	34
4.8.4	Palvelukäyttökirjanpito palveluarkkitehtuurissa.....	34

5 Yhteenveto.....	36
6 Termejä.....	37
Lähteet	38

1 Johdanto

Tämä raportti on osa Kehittämisen menetelmät -opintojaksoa. Ryhmämme muodosti tutkimusongelman tietojärjestelmien integroinnin osa-alueelle, jonka ratkaisut liittyvät olennaisesti tietojärjestelmän kehittämisen menetelmiin. Tutkimusongelma oli riittävän haastava ja kiinnostava, liittyi käytäntöön eikä siihen ollut valmiita vastauksia. Tiedonrakentelun tärkein tulos eli oppimisprosessi, uusi tieto, saavutettiin yhteisöllisessä tiedonrakentelussa. Tehtävän tulos on esitetty tässä tiedonrakentelun raportissa.

Opintojaksolla opiskelijaryhmien tuli selvittää ongelmia kolmessa iteraatiossa oman keskittymisalueensa aiheesta. Ensimmäisen iteraation ongelmana ryhmällämme oli ”SOA:n käyttöönottamisen ongelmat” ja halusimme selvittää, onko SOA:n käyttöönotossa ongelmia. Toisen iteraation ongelmana oli ”Miten SOA Management huomioidaan SOA-projektissa”. Tämän jälkeen ryhmä kokoontui ja päätti muuttaa aikatauluun siten, että kolmas iteraatiokierros jää pois. Päädyimme kahteen iteraatioon, jotta aikataulu ei olisi niin tiukka ja ettei järkevän sisällön muodostaminen vaarantuisi.

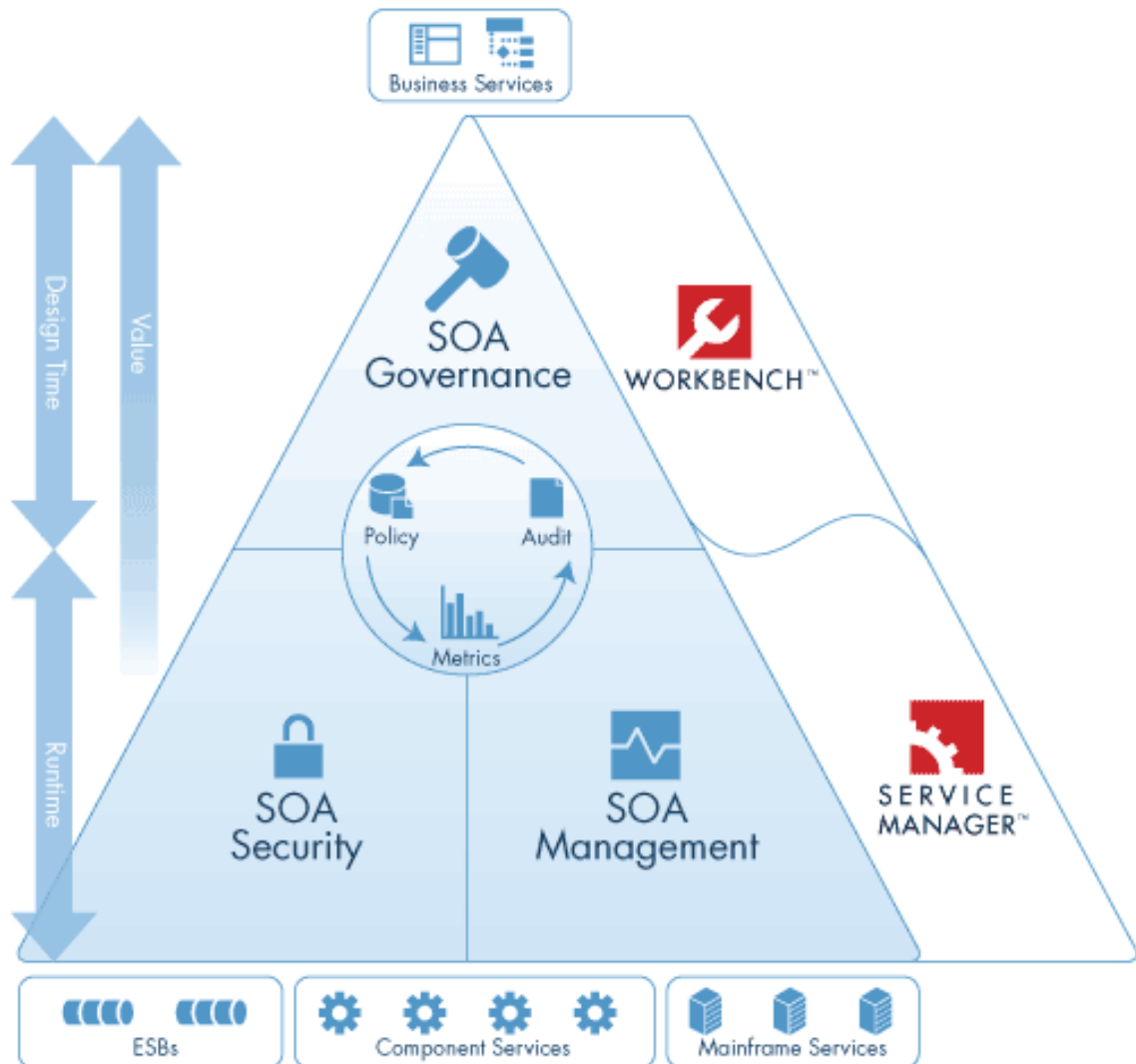
Lopullinen raportti muodostuu siten, että ensin käsittelemme syntynyttä tietoa siitä, mitä SOA on ja miten SOA Management huomioidaan SOA-projektissa. Tämä tieto on olennainen lähtökohta ratkaistaessa SOA:n käyttöönottamisen ongelmia, joita käsitellään viimeiseksi.

2 Service Oriented Architecture

Jotta pystymme käsittelemään SOA:n käyttöönoton ongelmia, on hyvä ensiksi määritellä mitä SOA:lla tarkoitetaan.

2.1 Mitä SOA on?

Seuraavat kuvat kuvaavat SOA:an liittyvää kokonaisuutta.



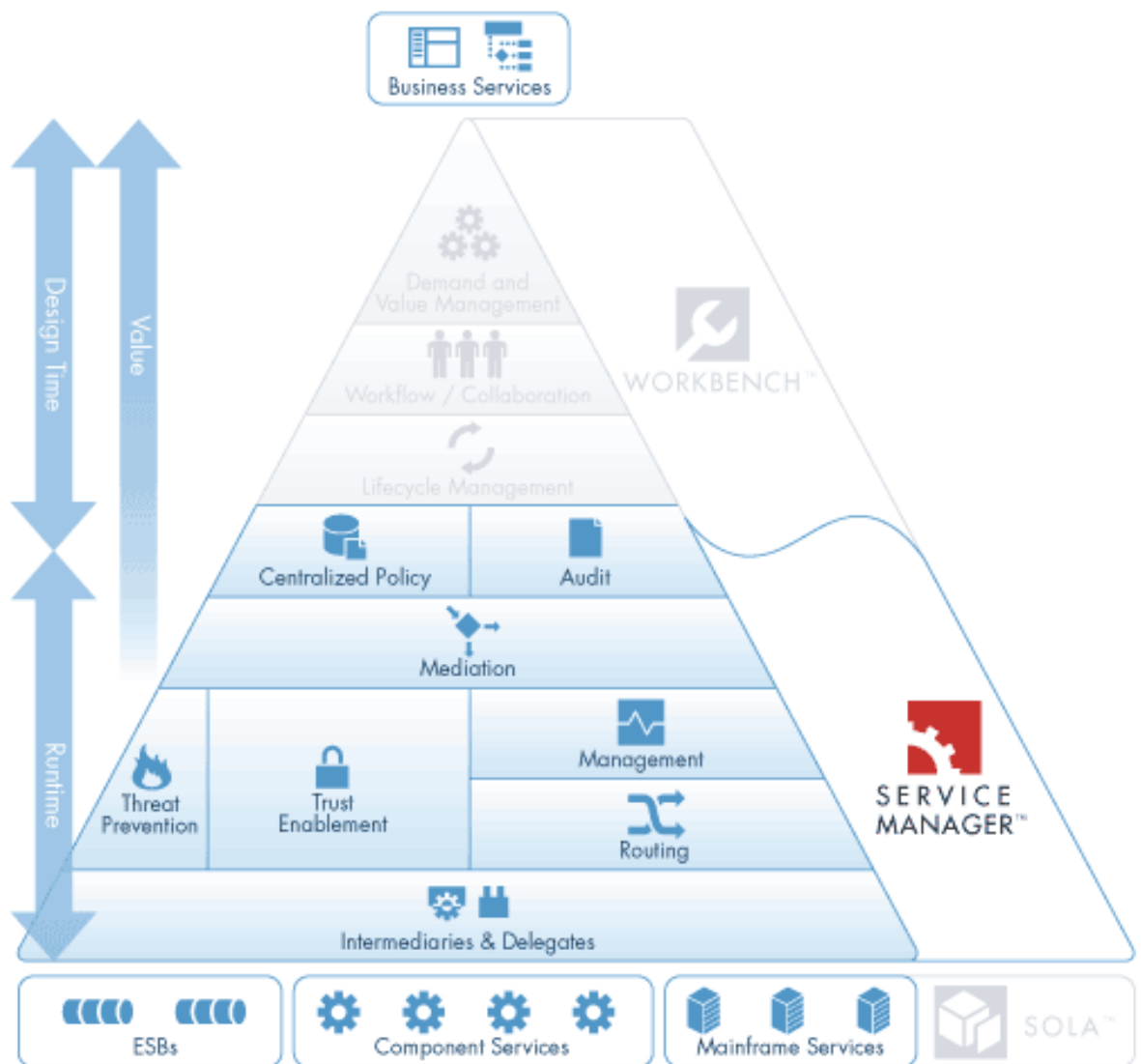
Kuvio 1: SOA kokonaisuus (www.soa.com).

Kuviossa 1 SOA kokonaisuus on esitetty pyramidin muodossa, joka on jaettu pienempiin osaluokkiin, joita ovat Ohjaus (Governance), turvallisuus (Security) ja hallinta (Management). Näiden keskellä on menettelytavat (Policy), mittarit (Metrics) ja tarkastukset (Audit). Pyramidin rakentuu palveluväylistä (ESBs Enterprise Services bus), komponenttipalveluista (Com-

ponent Services) ja keskuskonepalveluista (Mainframe Services). Pyramidin huipulla ovat liiketoiminnan palvelut (Business Services).

Pyramidin oikealla puolella olevat nuolet kertovat mikä osio on suunnitteluvaiheeseen (Design time) liittyvää aluetta ja mikä on suoritusaikaan (Runtime) liittyvää aluetta. Arvo-nuoli (Value) kertoo, että mitä ylemmäksi pyramidissa nousta, sitä enemmän se tuo arvoa liiketoiminnalle.

Kuviossa 2 on kuvattu SOA:n hallinnan eri osioita. Workbench ja Service Manager osiot ovat SOA Softwaren ohjelmistoja, joilla pyramidin osioita hoidetaan.



Kuvio 2: SOA-hallinnan osa-alueet (www.soa.com).

2.2 Palvelukeskeisen arkkitehtuurin taustaa

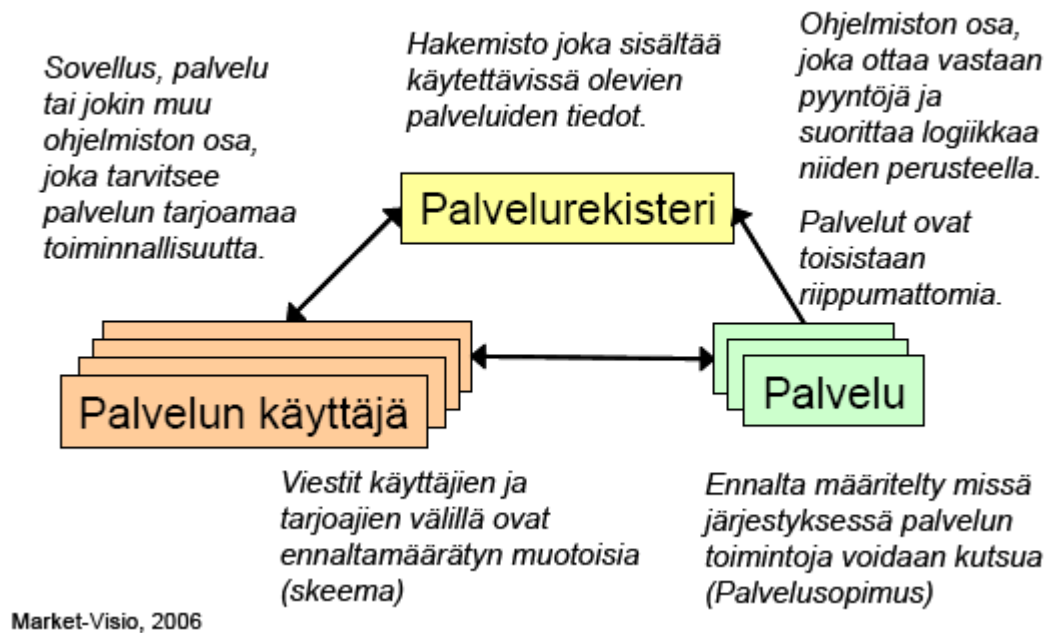
Nimensä mukaisesti palvelukeskeinen ajattelumalli perustuu ohjelmistojen muodostamien toiminnallisuuden paketoimiseen liiketoiminnan tarpeisiin perustuviin palveluihin. Palvelukeskeinen toimintamalli ei korvaa käytössä olevia järjestelmiä. Käytössä olevien järjestelmien toiminnallisuuden voidaan hyödyntää kuten tähänkin asti tai toiminnallisuuden voidaan koostaa palveluita ja näin ollen hyödyntää olemassa olevia järjestelmiä palvelukeskeisen toimintatavan mukaisesti. (Market-visio. 2007.)

Palvelukeskeisen arkkitehtuurin määritelmä riippuu tarkastelun lähtökohdista. Palvelukeskeistä arkkitehtuuria voidaan liiketoiminnan lähtökohdista kuvailla arkkitehtuurina, jossa olemassa olevista ja uusista toiminnallisuuden kootaan tarpeen mukaisia palveluita, joita hyödynnetään liiketoiminnassa. (Market-visio. 2007.)

Keskeisiä palvelukeskeisen arkkitehtuurin osia ovat palvelun käyttäjä, palveluksi paketoitu toiminnallisuus sekä palvelurekisteri (Kuvio 3). Palvelut sisältävät liiketoiminnan vaatimaa toiminnallisuutta. Niitä muodostetaan käytössä olevia ohjelmistoja ja tietolähteitä hyödyntäen tai luomalla kokonaan uutta toiminnallisuutta sovelluskehityksen kautta. Yleisesti palvelut muodostetaan yhdistelemällä useita edellä mainituista lähteistä. Palveluina voidaan hyödyntää myös kumppanien tai palveluntarjoajien tarjoamaa toiminnallisuutta. Palveluiden tunnuspiirteitä ovat:

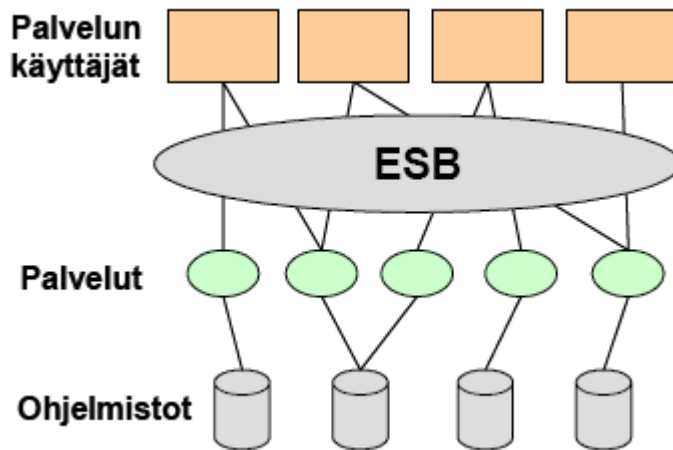
- Riippumattomuus muista palveluista. Riippumattomuuden ansiosta palveluita voidaan joustavasti muokata liiketoiminnan tarpeita vastaaviksi ilman, että muutoksia tehdään muualle kuin tähän yhteen palveluun.
- Palveluiden ja palveluiden käyttäjien tekninen toteutus ovat toisistaan riippumattomia. Palvelut kuorutetaan esimerkiksi Web Services -standardeilla, joiden avulla ne ovat palveluiden käyttäjien käytettävissä.
- Tavoitteena on, ettei ole kahta samanlaista palvelua. Uudelleenkäytettävyys saavutetaan, kun palveluita hyödynnetään useamman palvelun käyttäjän toimesta (Kuvio 3).
- Palvelu ei saa olla liian pieni, jolloin se on lähinnä tekninen objekti, eikä liian suuri, jolloin siitä tulee uudelleenkäytettävyden kannalta haastava. (Market-visio. 2007.)

Liiketoimintaprosessin edetessä vaiheeseen, jossa tarvitaan tietynlaista toimintoa, palvelun käyttäjä tiedustelee palvelurekisteristä, mikä palvelu täyttäisi tämän tarpeen. Palvelun käyttäjä voi olla esimerkiksi ohjelmisto tai sen osa. Palvelurekisteri, joka sisältää tiedot käytettävissä olevista palveluista, vastaa palvelun käyttäjälle ja toimittaa tiedot sopivasta palvelusta. Tämän tiedon avulla palvelun käyttäjä lähettää palvelupyynnön haluamalleen palvelulle. Palvelu toimittaa pyynnön mukaisen palvelun käyttäjälle. (Market-visio. 2007.)



Kuvio 3: Palvelukeskeisen arkkitehtuurin keskeiset komponentit

Olemassa olevan IT-ympäristön kehittäminen palvelukeskeiseen suuntaan on useimmin vaiheittain etenevä prosessi. Liikkeelle voi lähteä yksittäisten palveluiden muodostamisesta ja näiden hyödyntämisestä esimerkiksi point-to-point periaatteen avulla, käyttäen WebService-tekniikoita. Kun palveluita ja palveluiden hyödyntäjiä on suuri joukko, törmätään samoihin ongelmiin, kuin perinteisessä point-to-point – sovellusintegroinnissa, eli esimerkiksi erilaisten rajapintojen määrä kasvaa hallitsemattomaksi. Välikerroksen ohjelmistot, kuten palveluväylät (Enterprise Service Bus / ESB), tulevat tarpeellisiksi palveluiden ja palveluiden käyttäjien määrän kasvaessa. Palveluväylä muodostaa integraatiokerroksen, joka mm. välittää ja muuntaa palveluita (Kuva 4). (Market-visio. 2007.)



Kuvio 4: Palveluväylä osana palveluarkkitehtuuria.

(Market-visio. 2007.)

2.3 Palvelukeskeiseen arkkitehtuuriin liittyvät ohjelmistot

Keskeiset ohjelmistosegmentit, joita hyödynnetään palvelukeskeistä arkkitehtuuria kehitettäessä ovat:

- Yhdistelmäsovellusten kehitysympäristöt (Integrated Service Environment, ISE)
- Palveluväylät (Enterprise Service Bus / ESB)
- Palvelurekisterit (Service Registry)
- B2B Gateway -ohjelmistot
- Prosessien hallintaohjelmistot (Business Process Management / BPM)

(Market-visio. 2007.)

Yhdistelmäsovellusten kehitysympäristöt

Yhdistelmäsovellusten (Integrated Service Environment, ISE) kehitysympäristöt pitävät sisällään ainakin seuraavat toiminnallisuudet:

- Palveluiden kokoaminen ja orkestrointi
- Kevyt prosessien hallinta
- Kehitysympäristö
- Kehityksen automatisointi ja yksinkertaistaminen

ISE-ratkaisut muodostavat kehitysympäristön yhdistelmäsovelluksille. ISE-ratkaisujen päätarkoitus on sovellusten kokoaminen yhdistämällä palveluita. Ratkaisuihin sisältyy myös sovelluskehitys- sekä prosessienhallintaominaisuuksia. Kehityksen automatisointi ja yksinkertaistami-

nen tarkoittavat esimerkiksi koodigeneraattoreita tai viitekehyksiä, jotka piilottavat kehityksen monimutkaisuutta. (Market-visio. 2007.)

Palveluväylät

Palveluväylät (Enterprise Service Bus, ESB) pitävät sisällään ainakin seuraavia toiminnallisuuk-
sia:

- Tuki Web Services -standardeille
- Viestinvälitys
- Tietoturva
- Laajennettavuus
- Hallintaominaisuudet
- Sääntöperusteinen reititys
- Viestinvälitysprotokollien muuntaminen
- Viestimudon muuntaminen
- Viestin validointi

(Market-visio. 2007.)

Palveluväylät ovat Web Service -standardeja hyödyntäviä väliohjelmistoja, jotka tukevat älykäs-
tä ohjelmistojen välistä kommunikaatiota. Palveluväylä muodostaa integraatiokerroksen, joka
mm. välittää sanomia ja tekee sanomiin tarvittavat muunnokset. (Market-visio. 2007.)

Aiemmin hyödynnettyihin integraatiotuotteisiin verrattuna palveluväylät mahdollistavat entistä
löyhemmän yhteyden integroitavien sovellusten tai palvelun ja palvelun käyttäjän välillä. Palve-
lurväylien käyttö tulee ajankohtaiseksi, kun sovellusten välisten esim. Web Service-
standardeihin perustuvien yhteyksien lukumäärä nousee niin suureksi, että niitä on vaikea halli-
ta. (Market-visio. 2007.)

Palveluväylät tuovat hyötyjä erityisesti organisaatioille, joissa liiketoiminta edellyttää kykyä mu-
kautua nopeisiin muutoksiin tai olemassa olevan mahdollisesti kirjavien sovelluskannan hallinta
ja integrointi on työlästä. Palveluväylien ominaisuuksien ollessa rajatut, on laajoissa organisaa-
tiotason integraatiohankkeissa hyödynnettävä usein laajempia integraatiotuotteita. Haastavissa
integraatioympäristöissä palveluväyliä täydennetään edelleen esimerkiksi adaptereilla, kehit-
tyneemmällä prosessien hallinta- ja workflow-ominaisuuksilla tai organisaatioiden välistä integ-

raatiota tukevilla ominaisuuksilla. Kyseiset toiminnallisuudet eivät kuulu perustason palveluväyliin. (Market-visio. 2007.)

Palvelurekisterit

Palvelurekisterit (Service Registry) sisältävät seuraavia toiminnallisuuksia:

- Palvelutietojen hallinta
- Palveluiden hakuominaisuudet
- Turvallisuus - esim. pääsynhallinta palveluihin, roolien/ryhmien/käyttäjien hallinta, palveluiden hallinta oikeuksien mukaan.
- Palveluiden hallinta - esimerkiksi kontrolloitu palveluiden julkaisu ja löytäminen sekä ilmoitukset muutoksista palveluihin.
- Palveluiden luokittelu - esim. versiot, tietopalvelut / tekniset palvelut / liiketoimintapalvelut, taksonomian noudattamisen tarkistus.

(Market-visio. 2007.)

Palvelurekisterit ovat avainasemassa palveluiden hallitun ja tehokkaan hyödyntämisen kannalta. Palveluita kehitettäessä rekisteristä löydetään olemassa olevat palvelut, joita voidaan hyödyntää uudelleen sellaisenaan tai modifioida uusiin käyttötarkoituksiin. Palveluita hyödynnettäessä, palvelun ajon aikana, haluttu palvelu ja siihen liittyvät metatiedot löytyvät rekisterin avulla.

(Market-visio. 2007.)

Varsinaista palvelurekisterituotetta ei välttämättä tarvita, jos palveluita on kehitteillä tai kehitetty vain muutamia. Tällöin tiedot palveluista voidaan kerätä muuhun keskitettyyn paikkaan. Kun palveluiden määrä kasvaa, tarvitaan varsinaisen palvelurekisterituotteen ominaisuuksia, joita on lueteltu yllä olevassa listassa. (Market-visio. 2007.)

B2B Gateway –ohjelmistot

B2B Gateway –ohjelmistot sisältävät seuraavia toiminnallisuuksia:

- Keskitetty kumppaniprofilien luominen ja hallinta
- Viestien muunnokset
- Sisäänrakennettu palveluväylä ja palvelurekisteri
- Viestinvälityksen turvallisuus
- Adapterit
- Liiketoimintaprosessin hallinta (Business Process Management / BPM)

- Portaalit

Perinteisesti ydintoiminnallisuus B2B Gateway -ohjelmistoissa on kumppaniprofilien hallinta. Jatkossa tuotteita erottavaksi tekijäksi nousee voimakkaammin niiden sisältämät palveluväylät ja palvelurekisterit, joiden avulla palveluita voidaan tarjota kumppaniverkoston käyttöön. (Market-visio. 2007.)

Prosessien hallintaohjelmistot

Prosessien hallintaohjelmistojen toiminnallisuuksia (Business Process Management, BPM)

- Prosessien mallintaminen
- Prosessimoottori / Prosessien orkestrointi
- Tuki ihmisten ja ohjelmistojen suorittamille tehtäville
- Prosessien analysointi ja seuranta
- Työnkulun sääntöjen luominen ja hallinta
- Prosessirekisteri
- Integraatio-ominaisuudet (palveluväylät)
- Tietosisällön hallinta (dokumenttienhallinta)

Palvelukeskeisen arkkitehtuurin lähtökohta on tuottaa liiketoimintaa tukevia tietojärjestelmiä. Liiketoimintaprosessien kuvaaminen, analysointi ja optimointi muodostavat pohjan palvelukeskeisten sovellusten kehittämiseksi. Liiketoiminnan edustajilla on keskeinen rooli toiminnan asettamien tarpeiden välittämisessä tietohallinnon asiantuntijoille. Ohjelmistojen kehitystä tehostaa se, että sovellusten kehittäjät pystyvät jatkamaan työtä liiketoimintaedustajien kehittämien prosessimallien pohjalta. Tämä yhteistyö konkretisoi liiketoiminnan ja tietohallinnon yhteistyön tiivistymistä organisaatioiden kehittäessä palvelukeskeistä arkkitehtuuria. (Market-visio. 2007.)

Toiminnan joustavuutta edistää, että liiketoiminnan edustajat pystyvät prosessien hallintaohjelmistojen avulla tekemään muutoksia toimintatapoihin ilman tietohallinnon asiantuntijoiden osallistumista. (Market-visio. 2007.)

3 SOA Management

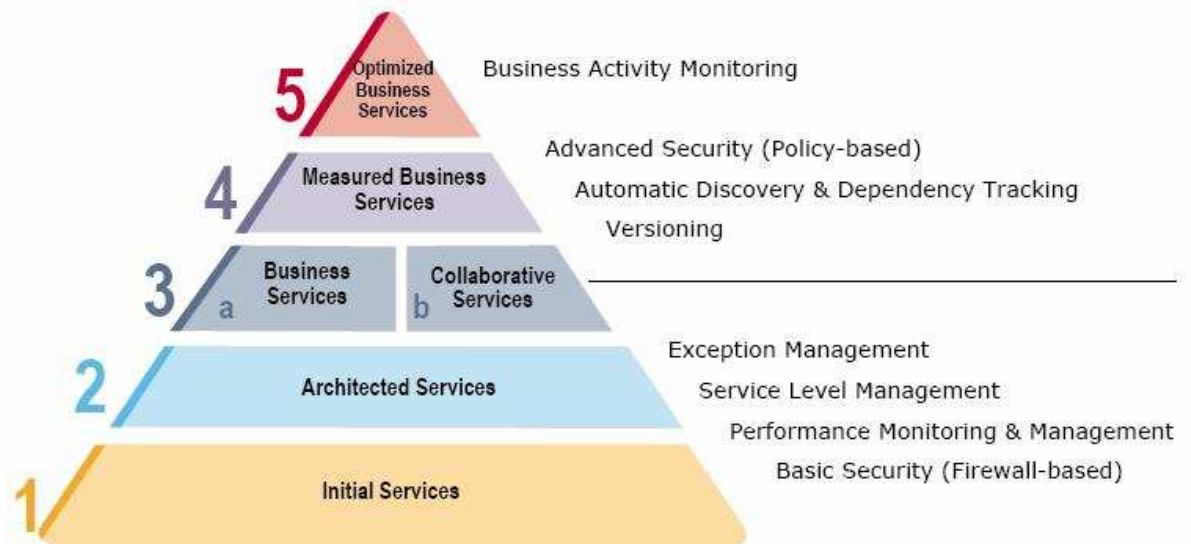
SOAn käyttöönotossa on monia hallinnoitavuushaasteita:

- Systeemiarkkitehdit, jotka hyväksyvät uudet standardit ja toteutuskehukset (framework:it).
- Kehittäjät, jotka ovat vastuussa vaatimuksista.
- Laadunvarmistus, joka vastaa itse käyttöönoton testauksesta.
- Tietoturvaosasto, joka integroi uuden järjestelmän olemassa oleviin tietoturvaratkaisuihin.
- Systeemioperaatiot, jotka varmistavat luotettavuuden ja suorituskyvyn
- Liiketoimintaoperaatiot, jotka varmistavat palvelutasot.

Tiedon epäyhtenäisyys aiheuttaa järjestelmiin monimutkaisuutta, koska tietojärjestelmät, tiedon versiot ja formaatit voivat olla erilaisia; tietoturvamenetelmät, identiteetit ja käyttöoikeudet poikkeavat toisistaan; tai tiedonsiirtomekanismit ovat erilaisia. Kokonaisvaltainen SOA arkkitehtuuri ja sen menestyksenkäs käyttöönotto ottaa huomioon muun muassa seuraavat asiat:

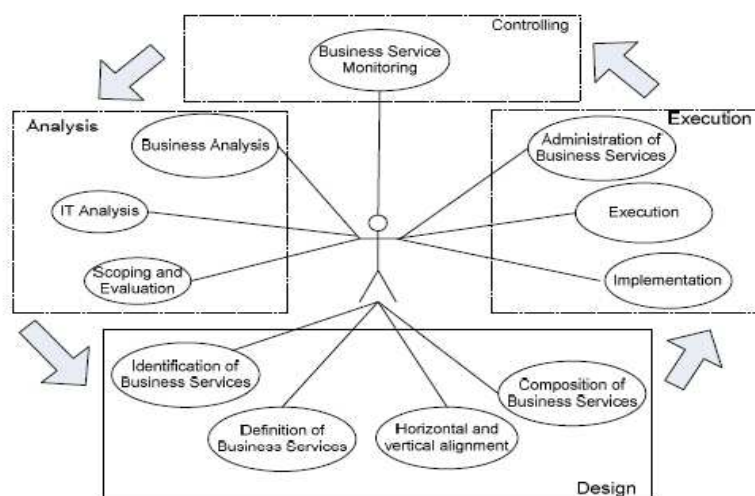
- suorituskyvyn hallinta
- tietoturva
- virhetilanteiden havainnointi ja korjaus
- automaattisen palveluiden haun
- versionhallinta ja migraatiot
- käyttötason hallinnointi
- poikkeuksien hallinnointi
- liiketoiminta-aktiviteettien monitorointi.

(Pogonoski, Paul.)



Kuvio 5: Huomioitavat asiat SOA:n käyttöönotossa

Koska SOA ei ole tekniikkaan sidottu suunnittelumalli, niin SOA:n hallinnalle ei riitä pelkkä Webservice-palvelujen hallinta, vaan sen tulee huomioida myös muilla tekniikoilla toteutetut palvelut. Näitä voidaan valvoa esimerkiksi tapahtumien vasteaikojen valvonnalla tai palvelun käyttämien resurssien monitoroinnilla. SOA hallinta sisältää palvelujen hallinnan lisäksi muun muassa palvelujen version hallintaa, palvelujen muutoksen hallintaa, julkaisujen hallintaa, saatavuuden hallintaa, liiketoimintaprosessien hallintaa, konfiguraatioiden hallintaa ja roolien hallintaa. Alla olevassa kuvassa (kuvio 6) on kuvattu liiketoimintapalvelujen hallinnan elinkaaren eri vaiheet. (Bloomberg, Jason. 2006.)



Ku

vio 6: Liiketoimintapalvelujen hallinnan elinkaaren vaiheet

SOAn hallinnan menestys tarvitsee työkalut kaikkiin implementoinnin vaiheisiin kehityksestä testaukseen ja monitorointiin. Vaikka itse tekniikka onkin uutta, sen käyttöönotolla on samat tavoitteet kuin muilla vuosien varrella tulleilla tekniikoilla: paras käyttö kehitysresursseista, vähentää päällekkäisyyksiä, helpompi integroitavuus kumppanien kanssa ja mahdollisuus kytkeytyä liiketoiminnan ja verkkojen välille. Onneksi SOA teknologia on kypsynyt työkalujenkin osalta, jolloin on mahdollista jo hallita komponenttitasolla asioita: monitoroida, kontrolloida ja turvata hajautettujen verkkojen tuomat haasteet. (Betz, Mark.)

Oraclen Senior Director Dave Shafferin mukaan monet käyttäjät uskovat tehneensä merkittäviä investointeja IT-järjestelmiin ja eivät näe enää tarvetta SOAn käyttöönottoon: ERP järjestelmät pitävät sisällään prosessien automatisointia. Markkinoiden toimijat taas mainostavat SOA tarjoavan uuden abstraktisen kerroksen vanhojen järjestelmien päälle, joka auttaa liiketoimintaprosessien uudelleenorganisoimisen helpommin kuin vanhan ERP järjestelmän modifoinnin. Esimerkiksi Oraclen SOA- tuotteet tarjoavat kokonaisvaltaisen BPM (Business Process Management) ratkaisun, joka on suunniteltu toimimaan minkä tahansa nykyisen järjestelmän kanssa. SOA-arkkitehtuuriin pohjautuva teknologia mahdollistaa helpoimman polun suurkone-ympäristön modernisoinnin puhtailla rajapinnoilla, jossa itse hallinnointi on pääasemassa. (Business Review Ltd..)

3.1 Hallittavien resurssien yksilöinti (Identify resources to manage)

Eräs avainhaasteista SOA Managementissa on määrittellä, mitä resursseja pitäisi hallita. IBM erottelee hallittavien resurssien yksilöimiseen kaksi lähestymistapaa, joita voidaan käyttää yhdessä tai erikseen. Nämä ovat suunnittelu (design) ja havainto (discovery).

Suunnittelu (design)

Tässä lähestymistavassa resurssit, joita hallintaan ovat yksilöity aika-analyysissa ja suunniteluun pohjautuvaan ei-toiminnallisiin (non-functional) vaatimuksiin ja SLA:han (Service Level Agreement). Korkean tason tehtäviä ovat:

- Yksilöidä palvelut ja päästä päähän toimintoketjut.
- Yksilöidä resurssit.
- Määrittellä kriteerit monitorointiin.

Havainnot (discovery)

Tätä lähestymistapaa käytetään kun sovellus on otettu käyttöön joko testausympäristönä tai tuotantoympäristössä. Tämän jälkeen voidaan löytää tärkeimmät resurssit hallintaan. Havainnot voidaan käyttää yksilöimään sovellusta tai tukemaan infrastruktuuria, jotka perustuvat käytössä syntyneisiin yksityiskohtiin. (Best Practices for SOA Management. IBM Corp. 2007.)

SOA palveluiden tehokas hallinta on tärkeää, jotta saavutetaan SOAsta haluttu hyöty. Tähän tarvitaan sovelluksia, jotka monitoroivat ja valvovat päästä päähän toteutuksia, jotka perustuvat SOAlla tehtyihin palveluihin ja sovelluksiin. Nämä tarvitsevat älykkyyttä ja näkymättömyyttä. Onnistuneeseen SOA palveluiden levitykseen ja niiden hallintaan tarvitaan käytössä olevien sovellusversioiden ylläpitämiseen ja muutoksiin työkalua. Automatisoitu prosessien hallinta auttaa varmistamaan oikeellisuuden nykyisten ja tulevien sovelluksien kanssa ja vähentää kustannuksia sekä virheitä. Päästä päähän näkymättömyys perustuu jaettuihin resursseihin ja niiden ylläpitoon, jota auttavat monitorointi- ja hallintapalvelut. Nopeat muutokset ovat ominaisia SOA- ympäristössä. SOAan perustuvien sovellusten ja palveluiden jakaminen perustuu IT-resursseihin. Mikä tahansa tehty muutos yhteen palveluun tai sovellukseen voi aiheuttaa toisiin palveluihin ja ohjelmiin ei haluttuja, odottamattomia toimia. (SOA Management and Security.)

IBM on jakanut SOA-käsitteen seuraaviin skenaarioihin:

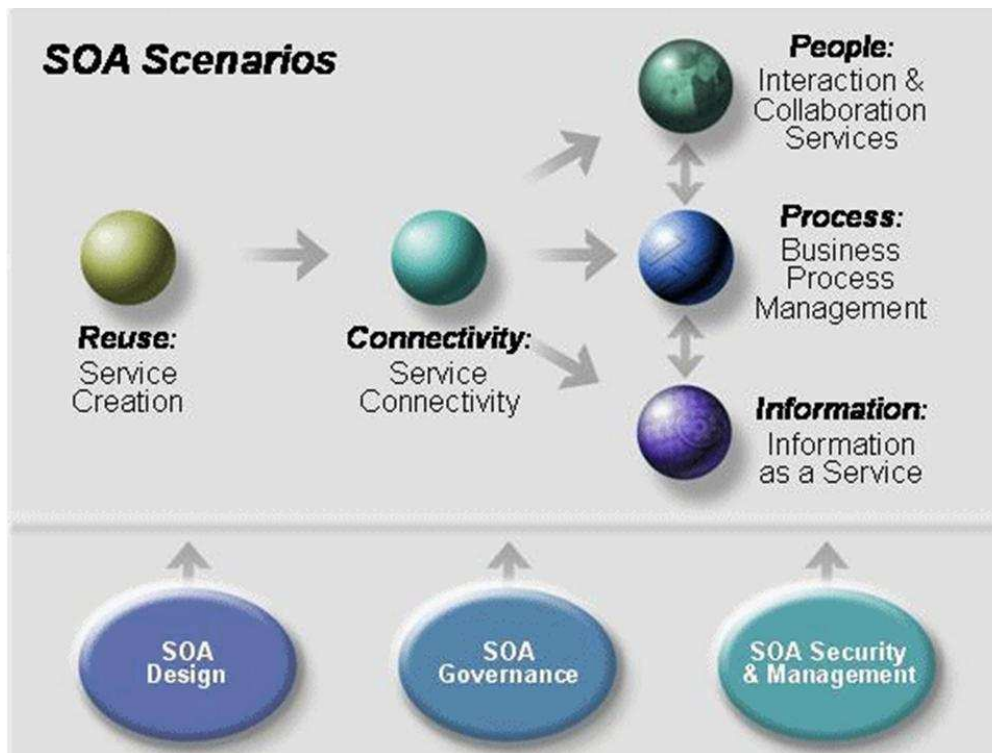
Ydinskenaariot:

- Palvelujen luonti
- Palvelujen liittäminen
- Vuorovaikutuspalvelut (käyttöliittymä)
- Liiketoimintaprosessin hallinta (liiketoimintalogiikka)
- Tietopalvelut (tietomalli)

Tukiskenaariot:

- SOA Design
- SOA Governance
- SOA Security & Management

Näiden eri skenaarioiden suhteen voi nähdä parhaiten seuraavasta kuvasta (Kuvio 7).



Kuvio 7: SOA:n eri skenaarioiden suhteet

Suurimmat haasteet SOA hallinnalle ovat:

1. Palvelujen suhteiden ymmärtäminen
 - Koska yhdestä palvelusta voivat olla riippuvaisia useat sovellukset, tulee palvelujen saavutettavuutta (availability) ja tehokkuutta (performance) tarkkailla jatkuvasti.
2. Palvelujen hallinta resursseina
 - Koska palveluilla on suora vaikutus liiketoimintaan, tulisi niitä kohdella kuten muitakin resursseja (esim. laitteita, yms.) ja niistä pitäisi luoda myös SLA-sopimukset.
3. Varmistaa ei-toiminnallisten vaatimusten täyttyminen
4. Tunnistaa hallittavat palvelut
 - Hallittavat palvelut pitäisi pystyä johtamaan SLA-sopimuksista ja ei-toiminnallisista vaatimuksista.
5. Komposiittisovellusten päästä-päähän (end-to-end) valvonta
 - SOA-ratkaisussa yksi liiketoimintaprosessi voi koostua useasta eri ympäristössä toimivasta palvelusta. Niiden monitorointi voi olla haasteellista.

(Best Practices for SOA Management. IBM Corp. 2007.)

3.2 Miksi hallinta korostuu juuri SOAssa?

Siirryttäessä kohti SOA-maailmaa, muodostuu liiketoimintaprosessien kautta joukko palveluiden välisiä loogisia riippuvuussuhteita. Näistä riippuvuussuhteista syntyy rajoitteita ja vaatimuksia palveluiden ja prosessien kehittämiseen. Palvelut, niiden toteuttamat vaatimukset sekä rajaukset, niiden elinkaari ja mahdollinen kontekstiriippuvaisuus sekä suhteet muihin palveluihin ja prosesseihin on pystyttävä mallintamaan ja hallitsemaan, jotta kehitystoimenpiteiden aiheuttamat muutosvaikutukset olisivat millään tavoin ennakoitavissa. Käytännön kokemukset osoittavat, että ylevistä tavoitteista huolimatta, järjestelmäkeskeisessä kehitysprosessissa palveluverkostojen hallintaan ei kiinnitetä riittävästi huomiota. Kokemuksen osoittava myös että palveluverkoston hallintaan käytettävä aika kasvaa hallitsemattomassa ympäristössä eksponentiaalisesti suhteessa palveluverkoston kokoon. (IT Universitas nro 9.)

Muutos lähtee liikkeelle päätöksestä kehittää liiketoimintaa. Jokainen päätös johtaa joukkoon toimenpiteitä, mutta samalla se johtaa joukkoon odottamattomia vaikutuksia palveluverkostossa. Jälkimmäisten määrä on yleensä suorassa suhteessa palveluverkoston kompleksisuuteen ja kokoon. Myös suunnitelluista toimenpiteistä syntyy kustannuksia, mutta nämä ovat ennakoitavissa. Odottamattomat vaikutukset ja niistä syntyvät kustannukset on mahdotonta ennakoida varsinkin, koska jokainen palveluverkoston kohdistuva toimenpide, suunniteltu tai ennakoimaton, yleensä johtaa joukkoon uusia ennakoimattomia toimenpiteitä. Ennakoimattomien vaikutusten analysointi ja korjaus, yleensä kiireellä, on itsessään virheeltistä työtä. Käytännössä hallitsematon palveluverkosto toimii kustannustehokkaasti ainoastaan riittävän pienissä kokonaisuuksissa. Mallintamalla formaalilla ja koneellisesti tulkittavassa muodossa palveluverkosto elinkaarivaiheineen, pystytään palveluverkoston kohdistuvien toimenpiteiden muutosvaikutukset analysoimaan luotettavasti, kattavasti ja kustannustehokkaasti jo toimenpiteitä harkittaessa. Samalla liiketoimintaa kehittävien päätösten tueksi saadaan luotettava tieto muutoksen kustannusvaikutuksista jolloin kehittämisen kannattavuus voidaan aidosti arvioida. (IT Universitas nro 9.)

3.3 Miten SOA Management huomioidaan SOA projektissa?

WM-Data (Jan Mickoc) on jaotellut SOAn hallintatarpeita seuraaviin osa-alueisiin:

- versionhallinta
- elinkaarenhallinta
- muutoksen hallinta
- konfiguraation hallinta
- identiteettien hallinta

- tietoturvan hallinta
- laatutason hallinta.

(Systemityöyhdistys, SOASig)

SysOpenDigia (Kari Parkkinen) tuo esiin SOA Governancen haasteita sekä toimenpidesuosituksia SOA projektille:

- tee SOAsta karkean tason Business Case, laadi palveluiden elinkaari
- vastuuta SOA Demand ja SOA Supply ja dokumentoi nykytila
- yrityksen sovelluskehitysmallin läpikäynti ja palveluohjeistuksen lisääminen siihen
- yrityksen projektinohjausmallin läpikäynti ja palveluohjeistuksen lisääminen siihen
- ohjeistuksen jalkauttaminen projektiin
- huolehdi, että liiketoimintaprosessien kuvaukset ovat käytettävissä.

(Runtime governance for SOA.)

4 Onko SOA:n käyttöönotossa ongelmia?

4.1 Ongelmakohdat

Verkon keskustelualueella heränneessä keskustelussa ryhmä tunnisti seuraavat osa-alueet, joihin ongelmia liittyy:

1. Palvelut ja niiden tunnistaminen ja kuvaaminen
 - Palvelujen tunnistaminen
 - Palvelutarpeiden kartoitus
 - Liiketoimintaprosessin mallintaminen
 - Kuvausten riittävä tarkkuustaso
 - Palveluiden jako sopiviin osiin
 - Palveluiden kuvaus
 - Prosessien tunnistaminen ja mallintaminen
3. Palvelujen hallinta ja ylläpito
 - Palvelujen dokumentaatio (sis. mm. BPM-tuotteiden (ja miksei muidenkin tuotteiden) dokumentaatiomahdollisuudet)
 - Palvelujen uudelleenkäyttö
 - Prosessin omistajat
3. Saatavuus ja suorituskyky
 - Vasteajat
 - 24/7-vaatimus ja sidosryhmät
 - Huoltokatkojen suunnittelu
 - Sopimukset (sis. ylläpitosopimukset laitteille, ohjelmistoille, tietoliikenteelle) ja SLA (Service Level Agreement)
 - Pitkän ja lyhyen aikavälin suunnittelu
4. Uudet teknologiat
 - Webservice
 - BPEL
 - BPM
5. Koskee useita eri järjestelmiä
 - Eri järjestelmiä ja teknologioita
 - Useiden projektien vaiheistaminen
 - Aikataulun laatiminen

6. Johdon tuki ja liiketoimintahyödyt
 - Johdon sitouttaminen
 - Liiketoimintahyötyjen arviointi
7. Resurssit
 - Aika ei riitä integraatioprojekteille
 - Osaaminen puutteellista
8. Tietoturva
 - Tunnistaminen
 - Tekninen tietoturva
 - Käyttäjien hallinnointi
9. Testaaminen

Jokaisen kohdan alle on myös merkitty mahdollisia alakohtia, jotka liittyvät kyseiseen alueeseen. Kohtaa 9 testaaminen ryhmä ei käsitellyt sen enempää, vaan päätti jättää sen käsittelyn testaamisen tiedonrakenteluryhmän tehtäväksi.

Näistä löydettyistä ongelmista ryhmämme otti tarkempaan käsittelyyn seuraavat aiheet:

1. Palvelut ja niiden tunnistaminen ja kuvaaminen
2. Palvelujen hallinta ja ylläpito
3. Saatavuus ja suorituskyky
4. Uudet teknologiat
6. Johdon tuki ja liiketoimintahyödyt
8. Tietoturva

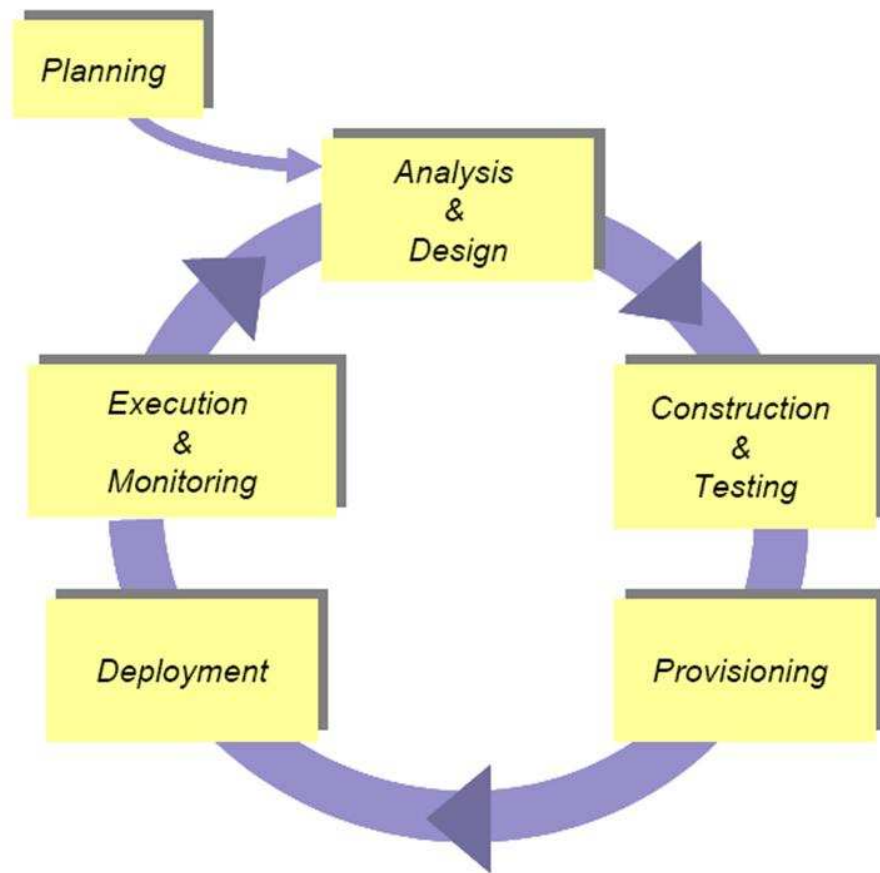
4.2 Palvelut ja niiden tunnistaminen ja kuvaaminen

4.2.1 Palvelujen määrittely

Seuraavassa on kuvattu lyhyesti Michael P. Papazogloun ja Willem-Jan van den Heuvelin artikkelissaan Service-Oriented Design and Development Methodology esittelemän SOA-suunnittelun ja kehityksen menetelmä.

Menetelmä koostuu eri vaiheista, joita ovat: Esisuunnittelu (planning), analysointi ja suunnittelu (analysis & design), rakentaminen ja testaaminen (construction & testing), provisiointi (provisioning), käytötönnotto (deployment) ja suoritus ja valvonta (execution & monitoring). Nämä

vaiheet muodostavat kokonaisuudessaan kehityksen elinkaaren. Seuraavaksi esitellään palvelujen tunnistamisen ja määrittelyn kannalta tärkeimmät vaiheet, jotka ovat esisuunnittelu, analysointi ja suunnitteluvaiheet.



Kuvio 8: Palvelujen suunnittelun ja toteuttamisen menetelmä

1. Esisuunnittelu (Planning)

Esisuunnitteluvaiheessa analysoidaan organisaation liiketoimintatarpeet, tarkastetaan nykyinen teknologiaympäristö ja suunnitellaan uuden ympäristön vaatimukset ja määritellään sen vastaavuudet saatavissa oleviin ympäristöihin. Tässä vaiheessa tehdään myös analyysi kustannuksista ja hyödyistä (johdon vakuuttaminen).

2. Analyysivaihe (Analysis)

Analyysivaiheessa kuvataan ensiksi nykyiset palvelut (services) ja liiketoimintatarpeet (business processes) ja tästä saadaan "as-is"-prosessimalli. Tämän jälkeen tehdään prosessien uudelleen-suunnittelu, jossa suunnitellaan ja analysoidaan nykyiset prosessit ja mietitään miten niitä tulisi muuttaa. Tästä saadaan "to-be"-prosessimalli.

Analyysivaihe koostuu neljästä eri osasta:

- Prosessien tunnistaminen (Process Identification)
- Prosessien tarkentaminen (Process Scoping)
- Prosessien aukkojen analysointi (Business Gap Analysis)
- Prosessien toteutusmahdollisuuksien arviointi (Process Realization Analysis)

Prosessien tunnistamisessa tutustutaan mallinnettavaan prosesseihin, jotta ymmärretään mitä palveluja ko. prosesseihin kuuluu ja tunnistetaan ne palvelut. Jos yrityksessä on jo luotu yksittäisiä palveluita, niin sijoitetaan ne oikeisiin prosesseihin. Samalla tunnistetaan myös eri prosessien yhteydet. Prosessien tunnistamisessa voidaan käyttää apuna standardiprosesseja, joita ovat esim. RosettaNetin tarjoamat valmiit prosessit.

Prosessien tarkentamisessa tutkitaan prosessien kokoa ja pyritään jakamaan "isot" ja monimutkaiset prosessit pienempiin ja yleiskäyttöisempiin prosesseihin. Tällä vältetään liian isoilta ja vaikeasti ylläpidettäviltä prosesseilta. Tässä vaiheessa määritellään myös prosessin alku ja loppu, asiakkaat ja käyttäjät, syötteet ja tulosteet sekä ulkoiset yhteydet.

Prosessien aukkojen analysoinnissa tarkennetaan prosessien ja palvelujen määrittelyihin yksityiskohtia. Samalla verrataan suunniteltujen palveluiden toiminnallisuuksia jo olemassa olevien palvelujen toiminnallisuuksiin ja pyritään löytämään sieltä jo valmiina olevia palveluita, joita voitaisiin käyttää hyväksi näiden prosessien toteuttamisessa. Vaiheen tuloksena syntyy ehdotuksia kehittää suunniteltuja palveluita, uudelleenkäyttää olemassa olevia palveluita tai ostaa valmiita palveluita.

Prosessien toteutusmahdollisuuksien arvioinnissa arvioidaan eri toteutusmahdollisuuksia kustannusten, riskien, hyötyjen sekä vaatimusten näkökulmista.

Uusien liiketoimintaprosessien toteutukselle on 4 eri vaihtoehtoa:

- Green-field development
- Top-down development
- Bottom-up development
- Meet-in-the-middle development

Palveluarkkitehtuurin soveltaminen terveydenhuollossa - Osa 2: prosessien ja palvelujen määrittely ja suunnittelu -dokumentissa on todettu Top-down ja Bottom-up –menetelmistä seuraavaa:

- Alhaalta ylös -menetelmässä käytetään pohjana olemassa olevia järjestelmäkuvauksia ja muita dokumentteja, jossa jo toteutetut sovelluspalvelut ja rajapinnat muodostavat pohjan uudelleenkäytölle.
- Ylhäältä alas -menetelmässä lähdetään liikkeelle tavoitetilan prosessikuvauksista, joista johdetaan tarvittavat palvelut ja rajapinnat.

3. Suunnitteluvaihe (Design)

Palvelujen suunnittelussa tulee huomioida palveluiden rakeisuuden (granularity) aste, palveluiden uudelleenkäyttö (reusability) ja palveluiden itsenäisyys (composability).

Palvelujen suunnittelu sisältää kolme osaa, jotka ovat:

- Struktuurinen määrittely (rajapinnat, portit, toiminnot)
- Toiminnallinen määrittely (toimintasäännöt, palvelujen käytön säännöt)
- Poliitikkojen (Policy) määrittely (hallittavuus, tietoturva, käyttöoikeudet)

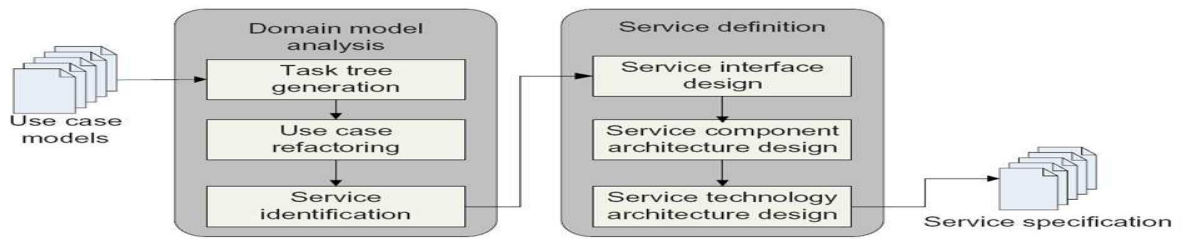
Suunnitteluvaiheessa kuvataan myös prosessit ja niiden rakenteet sekä prosessiin liittyvät roolit. Samoin kuvataan myös ei-toiminnalliset ominaisuudet, joita ovat muun muassa suorituskykyvaatimukset ja tietoturva-vaatimukset. (Papazoglou, Michael P. & van den Heuvel, Willem-Jan.)

4.2.2 Palvelujen tunnistaminen

Oikean kokoisten palvelujen tunnistaminen käyttötapausten perusteella

Yukyong Kimin ja Kyung-Goo Dohin dokumentissa The Service Modeling Process Based on Use Case Refactoring -kuvataan menetelmä sopivankokoisten palvelujen tunnistamiseen käyttötapausten perusteella.

Haasteena palveluiden määrittelyssä on sopivankokoisten palvelujen määrittely. Jos palvelut ovat liian karkealla tasolla, niiden uudelleenkäytettävyys kärsii. Jos ne ovat taas liian tarkalla tasolla, niin niiden hallinnasta ja käyttämisestä voi tulla hankalaa. Seuraavassa kuvataan lyhyesti dokumentissa esitellyn mallin palvelujen määrittelyyn käyttötapausten avulla.



Kuvio 9: Palvelujen tunnistaminen käyttötapauksen avulla.

1. vaihe

Ensiksi saaduista käyttötapauksista muodostetaan tehtäväpuut (task tree). Tehtävä on pienin käyttäjän havainnoiva toiminto, eli tehtävät esittävät käyttäjän ja järjestelmän dialogia.

2. vaihe

Toimintopuiden avulla tehdään käyttötapauksen uudelleensuunnittelu (refaktorointi) ja esim. erotetaan omiksi käyttötapauksiksi useille käyttötapauksille yhteiset tehtävät.

3. vaihe

Refaktoroidusta käyttötapausmallista johdetaan tarvittavat palvelut.

4. vaihe

Kun palvelut on löydetty, niin seuraavassa vaiheessa määritellään palvelun rajapinnat ja kuvataan itse palvelu.

5. vaihe

Tässä vaiheessa määritellään palvelun arkkitehtuuri, jossa kuvataan mm. palvelun riippuvuudet muista palveluista.

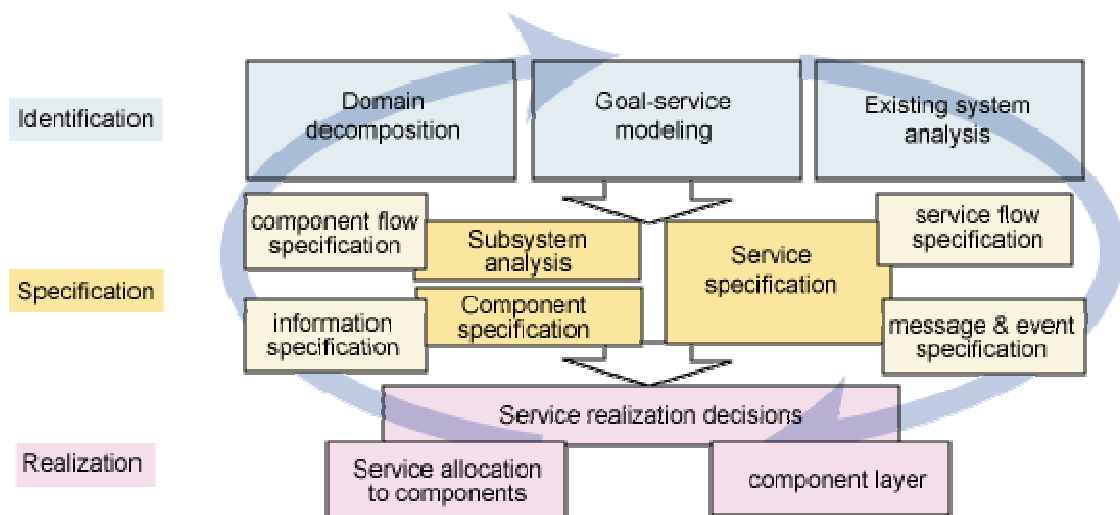
6. vaihe

Lopuksi viimeisessä vaiheessa valitaan palvelun toteutustekniikat (esim. SOAP, WSDL, yms.).
Kim, Yukyong & Doh, Kyung-Goo.)

Palvelujen tunnistaminen IBM:n SOMA-mallissa

IBM:n SOMA-mallin (Service-oriented modeling and architecture) mukaan palveluita voidaan tunnistaa seuraavilla metodeilla:

- top-down –menetelmä
- bottom-up –menetelmä
- middle-out –menetelmä



Kuvio 10: IBM:n SOMA-malli

Top-down menetelmässä (kuvassa Domain decomposition) lähdetään liikkeelle liiketoimintaprosessin kuvauksista, joista johdetaan prosessit, aliprosessit ja käyttötapaukset, joista voidaan muodostaa palveluita. Bottom-up menetelmässä (kuvassa Existing system analysis) lähdetään liikkeelle olemassa olevista järjestelmistä ja pyritään löytämään palvelut sieltä. Middle-out menetelmä (kuvassa Goal-service modeling) on yhdistelmä kahdesta edellisestä menetelmästä. (Arsanjani, Ali. 2004.)

4.2.3 Palvelujen luokittelu

Palveluarkkitehtuurin soveltaminen terveydenhuollossa - Osa 2: prosessien ja palvelujen määrittely ja suunnittelu -dokumentissa on kuvattu palvelujen tunnistamista seuraavasti.

Yleisesti suositeltu tapa tunnistaa ja määrittellä sovelluspalveluja on yhdistää prosessien määrittely sekä olemassa olevien sovellusten tutkiminen tarvittavien ja käytettävissä olevien sovelluspalvelujen määrittelymiseksi.

Palvelujen luokittelun tavoitteena on tunnistaa samantyyppisiä ratkaisuja samantyyppisiin tarpeisiin, auttaa kokonaisarkkitehtuurin hahmottamisessa palvelujen osalta sekä tarjota valmiita malleja erityyppisten sovelluspalvelujen tuottamiseen tai hankintaan.

Palveluita voidaan luokitella muun muassa seuraavilla tavoilla:

- arkkitehtuurin kerrokset (järjestelmäpalvelut, liiketoimintapalvelut, prosessipalvelut)

- palvelun tyyppi (peruspalvelut, välittäjäpalvelut, prosessipalvelut, organisaatiotason palvelut)
- teknisyys ja yleiskäyttöisyys (infrastruktuuri, ydinpalvelut, lisäpalvelut)
- palvelun kattamien integrointivaatimusten ensisijainen luonne (tietopalvelut, toiminnalliset palvelut, käyttäjäkeskeiset palvelut, prosessipalvelut)
- toiminnallinen luokittelu (tekniset palvelut, yleispalvelut, toimintokohtaiset palvelut)
- rakenteisuuden aste
- karkeajakoisuus ja abstraktiotaso

(Kuopion Yliopisto. 2007.)

4.2.4 Palvelujen dokumentointi ja kuvaaminen

Palveluarkkitehtuurin soveltaminen terveydenhuollossa - Osa 2: prosessien ja palvelujen määrittely ja suunnittelu -dokumentissa on kuvattu palvelujen dokumentointia seuraavasti.

Tunnistettujen palvelujen dokumentoinnin, ymmärrettävyyden ja rajausten kannalta on olennaista kuvata keskeisimmät sovelluspalveluun liittyvät seikat. Kuvausta tarvitaan sekä tarkemman suunnittelun tai hankintavaihtoehtojen ohjaamisessa että jo hankittuja tai käytössä olevia palveluja arvioitaessa uusissa käyttötilanteissa.

Palveluista kuvataan:

- palvelun nimi
- palvelun tarkoituksen lyhyt kuvaus
- palvelun luokittelut
- mihin prosesseihin liittyy
- mihin sovelluksiin/tuotteisiin liittyy
- mihin määrityksiin liittyy
- yhteydet ja riippuvuudet muihin palveluihin
- arvio saatavuudesta/ toteutettavuudesta/ mukauttamistarpeista
- muut kommentit.

Palvelusta voidaan dokumentoida myös paljon muita asioita, mutta näiden asioiden dokumentointi antaa yleiskuvan palvelusta.

Palvelut koostuvat operaatioista, joista kuvataan seuraavat asiat:

- operaation nimi, joka kuvaa lyhyesti kyseisen toiminnon tai palvelun tavoitteen
- rajapinta (ja sitä kautta palvelu), johon operaatio kuuluu
- selite operaation tarkoituksesta
- operaatiokutsun tietojen tai sanoman määrittely
- vastauksen tai paluuarvon tietojen tai sanoman määrittely
- tarkempi kuvaus operaation toiminnasta.
- tieto, mitkä tiedot ovat pakollisia
- lisätietoja toteutusmalleista
- poikkeustilanteet.

(Kuopion Yliopisto. 2007.)

4.3 Palvelun hallinta ja ylläpito

Seuraavat kohdat on poimittu kirjasta Järjestelmäintegraatio, tarve, vaihtoehdot, toteutus. Nämä eivät perustu pelkästään SOA-ratkaisuihin, vaan kaikkeen järjestelmäintegraatioon yleensä.

Järjestelmienvälinen informaationsiirto pitäisi toteuttaa siten, että se on kontrolloitavissa ja monitoroitavissa yhdestä tai useammasta pisteestä. Lisäksi arkkitehtuuria suunniteltaessa tulee ymmärtää eri järjestelmien keskinäinen toiminta ja tehtävän jako. Informaatiovirran tulee olla keskeytymätöntä ja tarkoituksen mukaista. (Tähtinen, Sami. 2005.)

4.3.1 Automatisoitu ja huoltovapaa integraatio

Arkkitehtuuria suunniteltaessa tulee päämääränä olla rakenne, joka mahdollistaa mahdollisimman vähän manuaalisia välivaiheita. Mikäli jokin integroitava ohjelmisto tarvitsee informaatiota jostain muusta järjestelmästä ja kykenee sitä pyytämään, integraatoratkaisun tulee kyetä vastaanottamaan järjestelmästä tuleviin kyselyihin ja välittämään pyyntö eteenpäin. Integraation tulee kyetä siirtämään automaattisesti dataa järjestelmien välillä ilman integraation pyyntöäkin.

Arkkitehtuurissa tulee ottaa huomioon ne herätteet, joiden perusteella integraatoratkaisun tulee integraatiossa käyttää. Herätteiden ei tarvitse ole yksinkertaisia. (Tähtinen, Sami. 2005.)

4.3.2 Vikasietoisuuden varmistaminen

Automatisoinnista ei ole hyötyä, mikäli ratkaisu ei ole toimintavarma. Toimimattomuus kostahtuu liiketoiminnan tarpeiden huonona palvelemisena, ylimääräisenä ylläpito- ja korjailutyönä sekä pahimmillaan yrityksen liiketoimintaprosessien pysähtymisenä.. Arkkitehtuuria suunniteltaessa tulee tiedostaa mahdolliset yksittäiset pisteet joiden lomaantumisen voi katkaista yrityksen liiketoimintaprosessit.

Järjestelmäintegraatoratkaisun vikasietoisuus tulee esiin kahdella eri tasolla: Ratkaisun teknisen alustan toimintana (rauta ja integraatioteknologia) ja varsinaisen integraatoratkaisun toimivuutena eli millä tavalla informaatiota siirretään ja kuinka vikatilanteissa toimitaan. Integraatoratkaisun tekninen toimivuus voidaan varmistaa samoin keinoin kuin minkä tahansa yrityssovelluksen toimivuus. Laadukkaat ohjelmistokomponentit, oikein mitoitettu ja parametroitu ratkaisu, henkilöstön riittävä koulutus, hyvänlaatuinen ja korkean käytettävyyden takaava laitteisto sekä jatkuva varpaillaan olo ovat hyvä lähtökohta.

Looginen vikasietoisuus on tärkeää, koska siinä nähdään muiden järjestelmien ongelmat. Jos jostain syystä on jokin integraation liitetyistä järjestelmistä pois käytöstä, ei integrointiprosessi voi toimia oikein. Näihin tilanteisiin tulee varautua kun tehdään integrointiratkaisun parametroitua.

1. Integraatoratkaisun tulee huomata tilanteet, joissa rajapinta integroitavaan järjestelmään ei toimi tai integroitavasta järjestelmästä vastaanotettu informaatio on väärämuotoista tai muutoin viallista. Rajapinnan toimimattomuuden tiedostaminen ja informaation muotoiseikkojen tarkastaminen ovat teknisesti helppoja toteuttaa, sisällön oikeellisuuden tarkastaminen voi olla vaikeaa.
2. Uudelleen yritykset, jos esim. järjestelmään ei saada muodostettua yhteyttä.
3. Vaihtoehtoisen ratkaisun löytäminen.. Integraatioprosessit ovat usein aikakriittisiä: niiden täytyy toimia tietyssä aikaikkunassa tai reaaliajassa. Näin olleen uudelleen yrityksiin ei ole aina aikaa.
4. Häilytykset. Kun integraatoratkaisu ei löydä tapaa toipua vikatilanteesta, sen tulisi kyetä informoimaan tarpeellisia osapuolia.

Vain pieni osa järjestelmäintegraatoratkaisun suunnittelusta on toiminnallisuuteen liittyvää. Loput käytetään varautumiseen mahdollisiin tai mahdottomiin vikatilanteisiin. Jos näitä ei tehdä heti suunnittelun yhteydessä voi se kostautua projektin kustannusten ylittymisenä että yrityksen kriittisen informaatiovirtojen katkeiluna. (Tähtinen, Sami. 2005.)

4.3.3 Integroinnin valvottavuus

Valvottavuus on tärkeää integraation läpinäkyvyyden kannalta. Valvonta voi olla joko tosiaikaista tai tietyllä viiveellä tapahtuvaa. Tosiaikaisesta käytetään usein nimitystä Business Activity Monitoring (BAM). Valvottavuus niveltyy vikatietoisuuteen. Kukaan ei ole kiinnostunut siitä, kun automaatio toimii. Vikatilanteissa saavat kuitenkin liikkeen aikaiseksi. (Tähtinen, Sami. 2005.)

4.3.4 Laajennettavuus ja muutoksen hallinta

Eräs integraatioarkkitehtuurin suunnittelun tärkeimmistä lähtökohdista on muutoksenhallinta. Yrityksen tietojärjestelmät muuttuvat ja vaihtuvat jatkuvasti ja mikäli nämä järjestelmät jakavat informaatiota toistensa kanssa integraatioarkkitehtuurin välityksellä myös integraatoratkaisun on mukauduttava tilanteeseen.

Kun järjestelmä vaihtuu versiopäivityksen tai toimittajan muutoksen vuoksi, voi tapahtua seuraavaa:

1. Mikään ei muutu. Uusi järjestelmä tarjoaa täysin samat rajapinnat ja toiminnallisuudet kuin aiemmin.
2. Tekniset rajapinnat muuttuvat mutta käyttötarkoitus ja käyttötapa säilyvät samoina.
3. Järjestelmän käyttötarkoitus muuttuvat. Järjestelmä voi olla joko suppeampi tai laajempi käyttötarkoitukseltaan kuin aikaisempi järjestelmä.

Teknisestä näkökulmasta ajatellen muutos voidaan ennakoida parhaiten hyvän dokumentaation avulla. Yritysten informaatiovirtojen pitäisi olla tiedossa arkkitehtuuritasolla. Yrityksen tietoteknisen arkkitehtuurin dokumentaation tulisi olla aina ajan tasalla – ja tarpeen mukaan saatavilla. (Tähtinen, Sami. 2005.)

4.3.5 Tietoturva ja käyttäjien hallinta

Järjestelmien välillä siirtyvä informaatio on usein kriittistä yrityksen liiketoiminnan kannalta. Tällaiseen liittyy aina luottamuksellista materiaalia kuten asiakastietoja ja yrityksen kirjanpitoon liittyviä lukuja. Järjestelmäintegraation kannalta keskeisimmät tietoturvan soveltamisalueet liittyvät luottamuksellisuuteen, eheyteen ja saatavuuteen.

Tietoturva liittyy myös tietotekniikan toimintavarmuuteen, tässä korostuu tietotekniikan eheyden vaatimukset. Myös informaation saatavuus voidaan nähdä hyvin teknisenä toimintavarmuuteen liittyvänä tekijänä. Palvelukeskeisessä arkkitehtuurissa ohjelmiston tulisi saada informaatio käyttöönsä välittömästi, kun se sitä pyytää. Tätä kautta informaation saatavuus linkittyy suoraan integraatoratkaisun toimintavarmuuteen. (Tähtinen, Sami. 2005.)

4.4 Saatavuus ja suorituskyky

Palvelulähtöisessä arkkitehtuurissa toimijat kytkevät omaan palvelukokonaisuuteensa sidosryhmien palveluja. Käyttäjän näkökulmasta käytettävä palvelu on yksi kokonaisuus, eikä käyttäjä tiedä, että taustalla voi olla ja on useammankin eri toimijan palveluita. Sekä palveluntarjoajalle että käyttäjälle tärkeitä elementtejä palvelussa ovat *saatavuus* ja *suorituskyky*. Tietojärjestelmän tai tietojärjestelmäpalvelun saatavuus on sitä, että palvelu on käytettävissä silloin kun sitä tarvitaan. Hyvä suorituskyky on sitä, että tietojärjestelmäpalvelu toteuttaa halutun toimenpiteen siten, että palvelua on mielekäs käyttää.

Palvelun saatavuuden ja suorituskyvyn varmistavia tekijöitä ovat toimijan oman sovelluskehityksen laadun varmistamisen lisäksi sopimusteitse varmistaa se, että sovelluksen ylläpidosta vastaavan toimittajan häiriöhuolto (esimerkiksi reagointi- ja palveluaika) sekä käyttöpalvelun tarjoajan palveluaika ja palvelulupaus vastaavat toimijan tavoitteita. Lisäksi tämä on varmistettava sopimusteitse myös niiden sidosryhmien kanssa, joiden tietojärjestelmäpalveluja toimija käyttää omissa palveluissaan. Sopimukseen kuvataan mm. palvelutasovaateet (Service Level Agreements – SLA) sekä sanktiointi, mikäli sovittuja palvelutasovaateita ei saavuteta. Palveluun voidaan sisältyä kuulumaan suunniteltuja, ennakolta yhdessä sovittuja palvelun käytön katkoja, esimerkiksi ympäristön tai tietoverkon huoltotoimenpiteisiin. Näitä yhdessä suunniteltuja ja etukäteen hyväksytyjä palvelukatkoja ei huomioida käytettävyyttä alentavaksi tekijäksi, kun lasketaan SLA:n mukaisia hyvityksiä raportointijaksolta.

Palveluiden käytettävyyysvaatimus on yhä useammin 24/7 vaatimus, jolla tarkoitetaan, että palvelun tulee olla aina saatavilla (24 h vuorokaudessa, 7 päivänä viikossa). Käytännössä tietojär-

jestelmät ja tietoverkot vaativat säännöllistä huoltoa, joista aiheutuu usein katko ko. komponentin käyttöön. Korkean palvelutason vaativissa tietojärjestelmäympäristöissä kaikki palvelun tarjoamiseen tarvittavat komponentit kahdennetaan, jolloin huoltotoimenpiteitä voidaan tehdä siten, että palveluun käyttäjän näkökulmasta ei aiheudu katkoa. Verkottuneessa palveluympäristössä myös ympäristöjen huoltokatkot tulee sopia sidosryhmien kanssa samoihin ajankohtiin, jotta palvelussa on mahdollisimman vähän suunniteltujakin käyttökatoja loppukäyttäjän näkökulmasta.

4.5 Uudet teknologiat

WebService-arkkitehtuurista löytyy useita eri tulkintoja, tuotearkkitehtuurista filosofisiin malleihin. Hyvä Webservice-arkkitehtuuri tarjoaa kehyksen, jossa voidaan systemaattisesti määrittellä komponentit palveluiden erilaisiin laatutarpeisiin.

Kokonaisarkkitehtuurikehys myös auttaa valitsemaan oikeat työkalut parantamaan tuottavuutta kehityksen yhteydessä ja käyttöönnotosta koko järjestelmän käyttöön aikana (esim. Suorituskykytestit). Pieniä koodinpätkiä voidaan lisätä SOAP-palvelimeen välimuistiksi, lokien kirjoittamiseen ja yhteysistunnon hallinnoimiseen.

BPEL parhaat käytännöt

Seuraavassa on lueteltu 5 yleisintä virhettä SOA-käyttöönnotossa:

1. Tarpeettomien palveluiden rakentaminen
2. Palvelutasojen seuraamisen ohjelmoiminen suoraan palveluihin
3. Synkronisten kysymys-vastaus-tyyppisten palveluiden rakentaminen
4. Heikko suorituskyky ja skaalautuvuustestaus ennen käyttöönottoa
5. Heikko ohjelmointikoodin uudelleenkäyttö

4.6 Liiketoimintahyötyjen arviointia

SOA-filosofia perustuu siihen, että saadaan liiketoimintaprosessit ja teknologiasovellukset toimimaan saumattomasti yhteen. Tällöin kokonaisuudesta saadaan enemmän irti ja entistä kustannustehokkaammin. SOAn on määrä ammentaa järjestelmistään ulos ominaisuuksia, palveluja ja niiden kautta lisäarvoa ja kilpailukykyä. (Öhrnberg. 2008. s.16-17.)

Eniten SOAa tarvitsevat isoja tietomassoja käsittelevät yritykset ja organisaatiot. Telealalle, pankeille ja vakuutusyhtiöille SOA tulee olemaan merkittävä kilpailutekijä. Kustannuspaineiden ja tulevaisuudessa myös työvoimapulan kanssa painivalle julkiselle sektorille SOA tuo niinkään isoja hyötyjä. (Öhrnberg. 2008. s.17.)

AMR Researchin selvityksen mukaan syyt SOA-investoinneille ovat seuraavat:

- 22% Mahdollistaa järjestelmien nopeamman ja edullisemmän muuttamisen.
- 18% Paras teknologia yksittäisen projektin vaatimuksiin.
- 17% Alentaa it-kuluja uudelleenkäytön kautta.
- 16% Kehittää tietotekniikkataitoja tärkeällä uudella alueella.
- 14% Modernisoi järjestelmäarkkitehtuuria.

(AMR Research. 2008.)

4.7 Johdon sitoutuminen

Kauppalehden (25/2008) mukaan on puhuttu paljon siitä, että yrityksissä liiketoimintajohto ja IT-johto eivät puhu samaa kieltä. IBM:n Suomen toimitusjohtajan Johan Sandellin mukaan komponenttimalleilla yrityksen johto voi entistä helpommin nähdä liiketoiminnan kokonaisuuden ja löytää mahdollisia synergiahyötyjä eri liiketoiminta-alueiden välillä. Sandellin mukaan komponenttimallien avulla bisnes ja IT saatetaan saman pöydän ääreen. Sandell toteaa myös, että tämä on eräänlainen yritysjohton ohjausväline, eräänlainen tulenjohtokartta. Tämä on tarpeen aina kiristyvässä globaalissa kilpailussa. Yrityksiltä vaaditaan yhä enemmän tehokkuutta ja nopeampaa reagointia sekä toiminnallista ketteryyttä. SOA mahdollistaa näiden mallien soveltamisen. (Öhrnberg. 2008. s.16-17.)

Yksi johdon sitoutumiseen vaikuttavia tekijöitä on kustannustehokkuus. IBM:n mukaan SOA on avaintekijä liiketoiminnan muutoksille, jotka voivat johtaa kustannustehokkaaseen IT-toteutukseen. Uusien teknologioiden liiketoiminnalliset hyödyt täytyy tuoda esille, jotta liiketoiminnan johto ja IT-toimijat saadaan vakuutetuiksi. (IBM Systems Journal. 2007.)

4.8 Tietoturva

4.8.1 Web Service tietoturva

Dokumentti ” Web Service Security” on malliesimerkki toimittajakohtaisesta ohjeistosta, joka ottaa yksityiskohtaisesti kantaa menettelytapoihin ja toimintamalleihin tietoturvallisten Web Service toteutusten aikaansaamiseksi. Dokumentti yksilöi myös mahdollisia riskitekijöitä, jotka liittyvät julkaistavien palveluiden toiminnallisuuksiin ja esim. huonon suojaustason omaaviin verkkoihin.

4.8.2 Sovellusintegraatio ja siihen liittyvät tietoturvatekijät

Tyypillisesti yrityksissä on toteutettu eri aikoina mahdollisesti eri ohjelmointikielillä sovelluksia kattamaan yksittäisten liiketoimintatarpeiden tietojenkäsittelyvaateita. Pahimmassa tapauksessa sovellukset on toteutettu tukeutuen eri teknologioihin ja eri laitealustoihin.

Sovellusintegraatiossa keskeisenä päämääränä on palveluiden ja niiden käsittelemien tietojen hallinta yrityskohtaisena kokonaisuutena. Dokumentissa Guidelines for Application Integration on otetaan kantaa sovellusintegraatioon liittyviin tietoturvakysymyksiin kappaleessa Security and Operational Considerations.

4.8.3 Tietoturvan varmistaminen Web sovellusten osalta

”Improving Web Application Security” on kattava dokumentaatio koskien Web sovellusten tietoturvan varmistamista ja jo olemassa olevien sovellusten tietoturvan parantamista.

Dokumentti opastaa seikkaperäisellä tavalla miten tietoturvaan liittyvät yksityiskohdat tulee huomioida palveluarkkituurin suunnittelussa ja toteutuksessa. Dokumentissa käydään läpi eri sovelluskerroksiin liittyvät tietoturva-asiat. Erinomaisena yksityiskohtana voidaan mainita osio jossa listaan eri sovelluskerroksiin liittyviä mahdollisia tietoturva uhkia. Tunnistamalla jo määrittely ja suunnitteluvaiheessa käytettävään teknologiaan liittyvät riskit, pystytään välttämään tyypillisimmät ”sudenkuopat”.

4.8.4 Palvelukäyttökirjanpito palveluarkkitehtuurissa

Eräs keskeisin SOA-palveluiden tietoturvaan liittyvä tekijä on tieto palveluiden virheettömästä toiminnasta, mahdollisten virhetilinteiden havainnointi ja hallinta, palveluiden käyttökirjanpitoon liittyvät tekijät sekä palveluiden suojausasetusten hallinta.

AmberPoint mainostaa itseään maailman johtavana SOA – palveluiden hallinta- ja monitorintiratkaisutoimittajana. Sivuston <http://www.amberpoint.com> sisältö on kokonaisuudessaan erittäin mielenkiintoinen.

AmberPoint tukee niin J2EE- (esim. IBM WebSphere, BEA WebLogic, SAP NetWeaver, JBoss) kuin .NET (Microsoft) -alustoilla tuotettujen palveluiden hallintaa. Myös EJB/JMS/JDBC-pohjaiset toteutukset.

AmberPoint SOA Validation System mahdollistaa kehitettävien palveluiden testauksen ja validoinnin ennen niiden tuotantoon siirtämistä.

AmberPoint SOA Management System tarjoaa mm. palveluiden välisten riippuvuuksien selvittämisen, palveluiden ajonaikaisen hallinnan ja monitoroinnin, virhetilanteiden hallinnan, tietoturvaratkaisun ja sovittujen SLA-tasojen varmistamisen.

Monet muut integraatiotoimijat ovat OEM-lisensoineet AmberPointin Management ja/tai Validation-tuotteen rajatun version osaksi omaa tarjontaansa. Microsoft on paketoinut AmberPoint Expressin osaksi Visual Studio 2005:tä.

5 Yhteenveto

Edellä on yhteenveto uudesta tiedosta, jonka opimme opintojakson tehtävässä selvittäessämme ongelmia kahdessa iteraatiossa oman keskittymisalueensa aiheesta. SOA:n käyttöönottamisen ongelmat kiteytyivät siihen, miten SOA Management huomioidaan SOA-projektissa. Varsinaisia aliongelmiä emme esitä tässä raportissa, vaan pystyimme jakamaan tutkittavan ongelman kysymyksiin, joihin lähdimme etsimään vastauksia.

Ohjasimme omaa oppimistamme yhteistoiminnallisesti ja vuorovaikutuksessa muiden oppijoiden kanssa. Asetimme tutkimusongelman, jonka koimme itsellemme merkitykselliseksi. Aluksi jokainen esitti tutkittavasta aiheesta omia käsityksiään, jotka altistetaan koko oppijaryhmän rakentavalle kritiikille. Työskentelyteorian pohjalta ryhdyimme etsimään aiheeseen liittyvää syventävää tieteellistä ja asiantuntijatietoa.

Käytimme tehtävään aikaa seuraavasti (tuntia):

Ryhmän jäsen	Ongelmien tarkentaminen (1. ja 2. kierros)	Omien teorioiden esittäminen	Kriittinen arviointi	Syventävän tiedon esittäminen
Leila	5+5	5+4	5+5	15+10
Virpi	6 + 2	5 + 4	4 + 2	15 + 10
Janne	5+7	14+8	10+5	15+10
Markku	5+5	10+10	5+5	10+10
Pasi				
Sonja	5+5	5+5	5+5	30+20

Tehtävän aikana koimme, että uusi tieto synnytti uusia kysymyksiä ja vaikeiden, laajojen ongelmien käsittelyä helpotti niiden pilkkominen pienemmiksi. Tässä tutkivassa oppimisessä olimme toistemme tiedonlähteitä ja ajatusten testaajia.

6 Termejä

käytettävyys (en usability)

Ominaisuus, joka ilmentää sitä, miten järjestelmä, laite, ohjelma tai palvelu soveltuu suunniteltuun tarkoitukseen tietyille kohderyhmälle.

huomautus

Hyvän käytettävyyden vastakohta on huono käytettävyys tai epäsopevuus.

Tähän käsitteeseen viitataan joissakin yhteyksissä myös yleiskielen ilmauksilla käyttökelpoisuus, käyttöönsoveltuvuus, helppokäyttöisyys ja käyttäjäystävällisyys.

saatavuus (en availability)

Ominaisuus, joka ilmentää sitä, kuinka varmasti järjestelmä, laite, ohjelma tai palvelu on tarvittaessa käytettävissä.

huomautus

Esimerkiksi tietojärjestelmän saatavuutta voivat heikentää järjestelmän virheet, jotka ajoittain estävät järjestelmän käytön.

Palvelun ominaisuuksiin viitattaessa merkintä 24/7 tarkoittaa, että palvelu on saatavilla 24 tuntia vuorokaudessa seitsemänä päivänä viikossa.

WWW-sovelluspalvelu (en web service (2); XML web service)

mieluummin kuin: web-palvelu

Verkkopalvelimessa toimiva ohjelma, joka tarjoaa standardisoitujen internet-yhteykäytäntöjen avulla palveluja sovellusten käytettäväksi

Huomautus

WWW-sovelluspalvelun ja asiakassovelluksen välinen yhteys voidaan toteuttaa SOAP-yhteykäytännön avulla (SOAP on lyhenne sanoista simple object access protocol). SOAP hyödyntää XML-kieltä, ja SOAPin ja XML:n avulla toteutettua sovellusta voidaan käyttää millä tahansa alustalla. Yhdistettäviä sovelluksia voivat olla esimerkiksi internetissä toimiva käännöspalvelu (WWW-sovelluspalvelu) ja hakupalvelu (asiakassovellus). Hakupalvelusovellus hakee palvelun käyttäjän tarvitsemat tekstien käännökset internetissä toimivasta käännöspalvelusta. Käyttäjän kannalta yhdistelyn tulos näyttää yhdeltä palvelukokonaisuudelta.

Englannin termiä Web Service käytetään myös laajemmassa merkityksessä, vrt. verkkopalvelu. (Tietotekniikan termitalkoot. 2002.)

Lähteet

AMR Research. 2008.

Arsanjani, Ali. 2004. Service-oriented modeling and architecture. IBM Corporation. Luettavissa: <http://www.ibm.com/developerworks/library/ws-soa-design1/>

Bentz, Mark. A Holistic Prescription For SOA Management
(<http://www.intelligententerprise.com/showArticle.jhtml?articleID=172302189>)

Bevan Nigel: Quality in use: Meeting User needs for quality, tiedosto: qiuse.pdf

Bloomberg, Jason. 2006. THE SOA MANAGEMENT LANDSCAPE

Business Review Ltd. Oracles Dave Shaffer talks BPM. Luettavissa:
http://www.cbronline.com/article_news.asp?guid=DDF91F7D-F839-42AC-9278-21AD07F0080A

Guidelines for Application Integration

IBM Corporation. 2007. Best Practices for SOA Management.

IBM Corporation. SOA Management and Security. Luettavissa: <http://www-306.ibm.com/software/solutions/soa/mgmtsec/soa-mgmt.html>

IBM Systems Journal Vol 46 No 4. 2007.

Improving Web Application Security

IT Universitas nro 9. Luettavissa: <http://www.yliopistojenit.fi/webblehti/nro9/soa.html>

IT2000-sopimusehdot

ITIL foundation, Service Level Management

Kuopion Yliopisto. 2007. Palveluarkkitehtuurin soveltaminen terveydenhuollossa

Osa 2: prosessien ja palvelujen määrittely ja suunnittelu.

Market-visio. 2007. SOA ohjelmistojen tarjonta Suomessa. s. 3-7. Luettavissa:
<http://www.pori.tut.fi/~hj/for-guests/public/ohjelmistot-seminaari/Sulonen-SOA-tarjonta.pdf>

Papazoglou, M. & van den Heuvel, W. 2006. Service-Oriented Design and Development Methodology.

Pogonoski Paul , SOA Management Challenges - Case Studies on Successful SOA, Japara Solutions Pty Ltd (http://www.open-standards.com/05papers/Paul_Pogonoski.pdf)

Progress Actional. Runtime governance for SOA.

Systeemityöyhdistys, SOASig. Luettavissa:
<http://www.ttlry.fi/yhdistykset/sytyke/osyt/soasig/>

Tietotekniikan termitalkoot. Luettavissa: <http://www.tsk.fi/termitalkoot/>

Tähtinen, Sami. 2005. Järjestelmäintegraatio, tarve, vaihtoehdot, toteutus. Talentum.

Yukyong K. & Kyung-Goo D. 2007. The Service Modeling Process Based on Use Case Refactoring. Springer-Verlag Berlin Heidelberg.

Öhrnberg. Kauppalehti 25/2008.