



KUOPION YLIOPISTO



centek

Tietojenkäsittelytieteen laitos, Ohjelmistotekniikka  
Department of Computer Science, Software Engineering

## Service Oriented Software Engineering (SOSE) - tutkimusprojekti

### TUTKIMUSSUUNNITELMA

#### 1. Tiivistelmä

Tutkimuksen tavoitteena on kehittää Pohjois-Savon alueella toimivien ohjelmistoyritysten ohjelmistotuotantomenetelmiä ja ohjelmistojen hyödyntämismahdollisuuksia asiakasyrityksissä. Erityisesti tarkastellaan **palvelukeskeistä ohjelmistoliiketoimintaa**. Palvelukeskeinen ohjelmistoliiketoiminta on syrjäyttämässä nopeasti perinteisen ohjelmistoliiketoiminnan malleja. Tutkimuksen keskeisenä osana on selvittää projektissa mukana olevien ohjelmistoyritysten ja heidän asiakkaiden vuorovaikutuksen ongelmia perinteisessä ohjelmistotuotannossa, jossa ohjelmistotalo toimittaa standardin ohjelmistotuotteen tai räätälöidyn projektin asiakkaalle, sekä analysoida, miten projektikeskeistä ohjelmistotuotantoa voidaan kehittää kohti **tuote- ja palvelukeskeistä ohjelmistokehitysmallia**. Tuote- ja palvelukeskeinen ohjelmistokehitysmalli mahdollistavat tuotetun ohjelmiston kokoamisen hyvin suunnitelluista rakeisista (ei-monoliittinen) osista (paketit, komponentit ja kehykset). Tämä tukee tuotettujen ohjelmisto-osien laajempaa uudelleenkäyttöä, ohjelmistotuotannon taloudellisuutta ja verkottuneiden toimijoiden yhteistyön kehittämistä prosessi- ja sovellusintegraation avulla.

Tutkimuksen tavoitteena on myös laajentaa Kuopion yliopiston ohjelmistotekniikan tutkimusryhmän toimintaa, kehittää Pohjois-Savon alueella toimivan teollisuuden tietotekniikan hyödyntämistä ja tukea alueella vaikuttavien ohjelmistotalojen kehitystä.

Ohjelmistotekniikan ja teknologian nopea kehittyminen on johtanut siihen, että asiakasyrityksillä ei ole riittävästi asiantuntemusta nykytietotekniikan hoitamiseen ja ohjelmistohankintaan. Tarvitaan tutkimusta, jotta voitaisiin määritellä, suunnitella ja esittää toteutuskelpoinen ratkaisu asiakkaiden käyttöön. Ohjelmistotekniikan tutkimusryhmän toimintaa laajennetaan ei vain terveydenhuoltoon vaan myös teollisuutta palvelevaksi.

Hanke toimii tiiviissä yhteistyössä Ylä-Savon ammattioiston "Laatua ICT-projektin toteuttamiseen" -projektin kanssa, jonka vetäjänä toimii Kari Puumalainen. Projektin tavoitteena on kehittää Pohjois-Savon alueella toimivien pk-yritysten IT-järjestelmähankinnoista vastaavien henkilöiden ohjelmistohankintaosaamista, ohjelmistoliiketoiminnan hallintaa, ostotoiminnan hallintaa ja tietojärjestelmähankkeiden koordinoitua osaamista. Tähän päästään kehittämällä asiakkaiden ja ohjelmistoyritysten kommunikointiedellytyksiä. Lopullisena tavoitteena on ohjelmistoprojektien ja

ohjelmistojen parempi laatu.

SOSE-hanke soveltaa PlugIT-hankkeessa luotuja määrittelyjä, menetelmiä ja toimintatapoja sekä luo niistä teollisuuden tarpeisiin toteutuksia. Erityisesti PlugIT-hankkeessa tehtyä vaatimustenhallinnan, sovellusintegraation ja hajautettujen tietojärjestelmien tutkimusta jatketaan tuloksia muokkaamalla ja edelleen kehittämällä (kts. <http://www.cs.uku.fi/research/Teho>). Hanke toimii kiinteässä yhteydessä PlugIT-hankkeen seuraajaksi ehdotettujen hankkeiden, kuten ZipIT, SerAPI ja ExportHIS kanssa (FinnWell) sekä tietojenkäsittelytieteen laitoksen muun tutkimuksen kanssa.

Tutkimukseen osallistuvat Kuopion yliopiston tietojenkäsittelytieteen laitoksen ohjelmistotekniikan tutkimusryhmä sekä joukko alueella vaikuttavia IT-yrityksiä ja asiakasyrityksiä. Projektin alustavat tutkimus- ja menetelmätuotokset on esitetty Kuvassa 1.

## TUTKIMUSKOHEET

### Vuosi 2004:



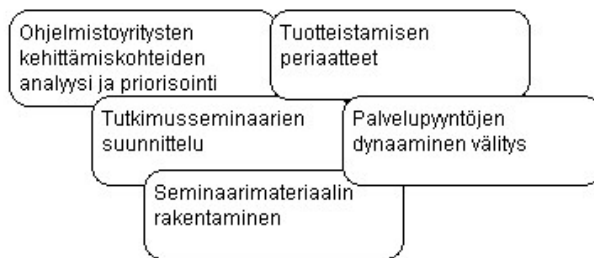
### Vuosi 2005:



### Vuosi 2006:



## TUOTOKSET



**Kuva 1: SOSE-projektin tutkimus- ja menetelmätuotokset**

## 2. Tutkimuksen lähtökohdat

### 2.1. Eräajoista palvelukeskeiseen arkkitehtuuriin

Tietojenkäsittelytehtävien historia voidaan jakaa kolmeen tai neljään jaksoon. Aina 80-luvulle asti tietojenkäsittelytehtävät perustuivat eräajotyypisiin järjestelmiin: Ensin tallennettiin järjestelmään tarvittavat tiedot ja tapahtumat. Tallennuksen jälkeen suoritettiin tarkistusajo ja varsinainen tuotantoajo. Halpojen PC-laitteiden, minitietokoneiden ja mikroverkkojen yleistymisen laajensi tietokoneiden käyttöä ja loi uusia käyttötapoja 80-luvulla. Entistä tehokkaammat mikrotietokoneet ja

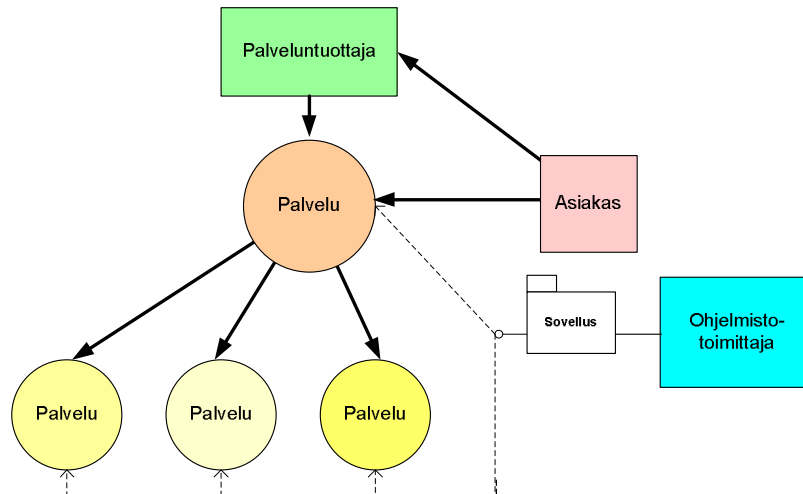
kehittyneet tiedonhallintaohjelmistot tarjosivat käyttöliittymänsä kautta mahdollisuuden tehdä reaaliaikaisia kyselyjä, päivityksiä ja tulostuksia. Tyypillinen tälle ajalle ominainen pk-yrityksen ohjelmistoratkaisu oli se, että toimistopalvelut toimivat käyttäjien mikrotietokoneissa (fat-client). Yhteisiä palveluja olivat minikoneella toimivat tiedonhallinta- ja tulostuspalvelut. Hallinnolliset ja operatiiviset sovellukset (esim. talous-, henkilöstö, materiaalihallinto) hoidettiin pääasiassa minitietokoneiden avulla ja ohjelmistotoimittajat tarjosivat ohjelmistopalveluitaan omilla alustoillaan. Ohjelmistot olivat luonteeltaan monoliittisia. Näin ollen ohjelmistojen kehittäminen vähitellen ja niiden integrointi toisiin sovelluksiin ja mikroverkkoihin oli kallista jopa mahdotonta. Yrityksen toiminnan kasvaessa tai supistuessa ohjelmisto jouduttiin yleensä korvaamaan kokonaan uudella. Tämä vaikutti epäedullisesti ohjelmistojen hyödyntämisen taloudellisuuteen.

90-luvun lopulla suuri osa sovelluksista toimi hajautetusti käyttäjäkerroksessa (User tier). Ongelmaksi muodostui vaatimus ohjelmistojen yhteistoiminnallisuudesta, ylläpidettävyydestä ja järjestelmien hallinnasta. Pyrittiin keventämään käyttäjän työpöytää (thin-client). Tavoitteena oli, että käyttäjän työpöydällä hallitaan sovellusten esittäminen (käyttöliittymä) ja varsinaiset liiketoimintalogiikkaa tukevat työtehtävät keskitetään sovelluspalvelimille, jotka käyttävät tiedonhallintaan siihen erikoistuneita tietokantapalvelimia. Tämä kehityssuunta korostuu organisaatioiden verkottuessa ja siirtyessä hajautettuun toimintamalliin, jolloin myös tietojärjestelmän tulee toimia hajautetusti.

Internetistä on muodostunut kaikkialle ulottuva tietoverkko, joka mahdollistaa globaalien verkkopalveluiden synnyn. Vanha palvelukeskusajattelu uudessa muodossa "palveluoperaattorina" on muodostumassa houkuttelevaksi ja kilpailukykyiseksi ratkaisuksi liikeyritysten miettiessä tietoteknisiä ratkaisujaan. Asiakas ostaa palveluita ja maksaa palveluista käytön mukaan. Palvelut auttavat yrityksiä ulkoistamaan toimintojaan, tällöin yritys voi keskittyä **ydinliiketoimintaansa**. Palvelujen ansiosta yrityksessä ei tarvita vaativaa ICT-osaamista kuten esimerkiksi varmistusten, suojausten ja tietoturvan hallintaa. Varsinkin pienyrityksissä työnjako parantaa toiminnan laatua ja poistaa toiminnan esteitä. Palvelutarjoaja vuorostaan **erikoistuu palvelutuotantoon** ja tarjoaa asiakkailleen parasta mahdollista palvelua. Palvelu koostuu esimerkiksi ohjelmisto-, laite-, ylläpito-, koulutus- ja konsultaatiopalveluista. Palvelu voi siis pitää sisällään myös ihmistyötä ohjelmistojen ja laitekapasiteetin lisäksi. Palvelutarjoaja voi ostaa osan palveluista toisilta palvelutarjoajilta ja niin edelleen. Palvelukeskeinen lähestymistapa (Service-oriented approach) tuottaa yritysten kesken monipuolisia verkostoja, joissa toiminnan tehokkuus saavutetaan erikoistumalla (Kuva 2). Tällaisessa toimijaverkossa ei tarkastella enää pelkästään toimittaja-asiakas suhteita, vaan asiakkaan asiakkaat ja koko toimijaverkon tyytyväisyys ratkaisevat menestyksen. Keskeiseksi kehittämiskohteeksi tällaisessa ympäristössä nousee palvelujen paketointi ja ryhmittely taloudellisesti ja toiminnallisesti tehokkaiksi riittävän turvallisiksi paketeiksi, joiden käytettävyys, käyttökelpoisuus ja toiminnallisuus tukevat asiakasyrityksen liiketoimintaa [Imm03].

## 2.2. Tuote- ja palvelukeskeinen ohjelmistoliiketoiminta

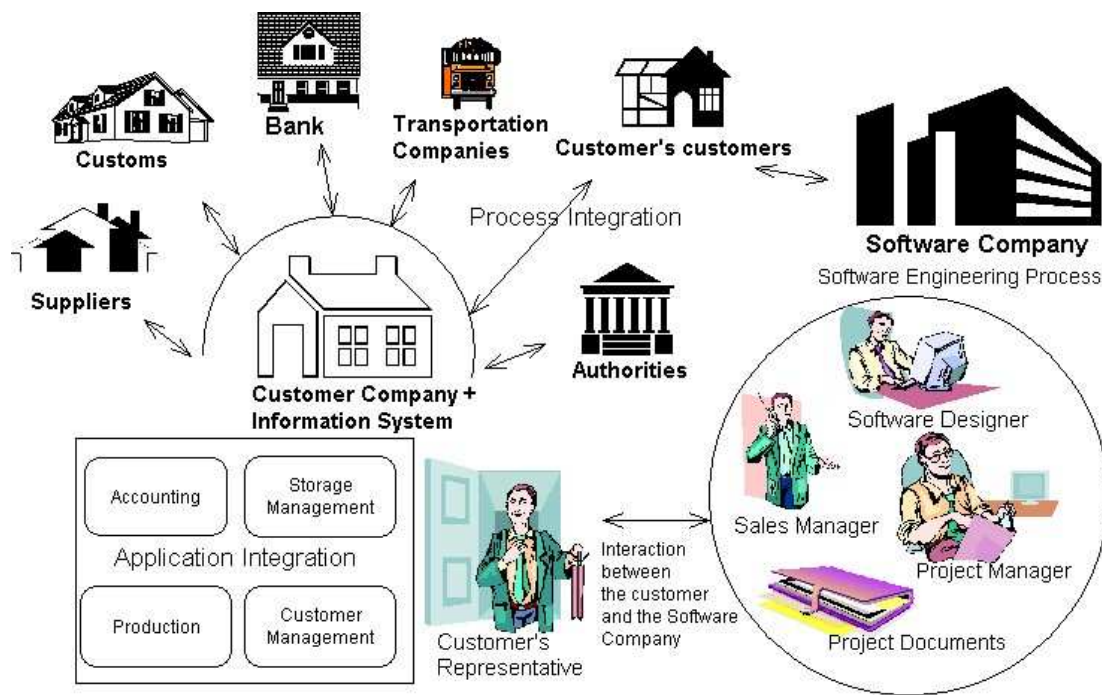
Ohjelmistotuotannon ongelmakenttää on havainnollistettu kuvassa 3. Ohjelmistotalo valmistaa ohjelmistotuotteet asiakkaille joko räätälöityinä projekteina asiakkaan tarkkojen toiveiden mukaisesti tai standardeina hyllytuotteina. Ohjelmistot tuotetaan ohjelmistotalon ohjelmistotuotantoprosessin mukaisesti ja asiakas saa tuotteen ostaessaan lisäksi projektidokumentit järjestelmän rakenteesta ja toiminnasta. On tärkeää, että ohjelmistotuotantoprosessi määritellään, suunnitellaan, kuvataan ja dokumentoidaan. Prosessin tunteminen parantaa ohjelmistokehityksen laatua ja tehostaa kehitysprosessia. Ohjelmistotuotteen rakentamisessa tulee pyrkiä jo rakennettujen ohjelma-osien uudelleenkäyttöön (reuse), jota tukevat komponenttipohjaisen sovellustuotannon menetelmät.



**Kuva 2: Palveluketju ja verkostoituminen**

Asiakasyritys (esim. pk-teollisuusyritys) haluaa uuden tietojärjestelmän tai sen osan, kun yrityksen vanha tietojärjestelmä ei vastaa enää liiketoiminnan vaatimuksia. Vanhan tietojärjestelmän toimintakelpoiset osat halutaan säästää ja pitää edelleen toiminnassa. Uuden tietojärjestelmän suunnittelussa tulee välttää päällekkäisten toimintojen toteuttamista (esim. asiakastietojen kirjaamista moneen kertaan) eli tarvitaan tietoa **järjestelmäintegraation** menetelmistä. Vientiä harjoittava teollisuusyritys on päivittäisessä yhteydessä tavarantoimittajiin, tulliviranomaisiin, pankkeihin, kuljetusyhtiöihin sekä omiin asiakkaisihinsa. Tarvitaan **käytettävyydeltään ja toiminnallisuudeltaan parempia tietojärjestelmiä** tukemaan tuotteiden vientiä. Web-pohjaiset ratkaisut auttavat yritystä jakamaan asiakkailleen tietoa tuotteista, niiden saatavuudesta ja hinnoista ja mahdollistavat **prosessi-integraation** asiakkaan kanssa (esim. nopeamman tilausprosessin, kun tilaukset ovat standardeja).

**Tuotokeskeisessä ohjelmistoliiketoiminnassa** ohjelmistotalo toimittaa asiakkaalle tuotteen, jonka asennuksen suorittaa joko asiakas itse tai ohjelmistotalon työntekijä. Yleensä asiakas joutuu maksamaan lisenssimaksun lisäksi lisähintaa tuotteen asennuksesta ja lisäpalveluista. Asiakas käyttää tuotetta, kunnes se ei enää täytä liiketoiminnan vaatimuksia ja tilaa uuden tuotteen joko samalta ohjelmistotoimittajalta tai eri toimittajalta, mikäli asiakasyritys ei ollut edelliseen tuotteeseen tyytyväinen.



**Kuva 3: Ohjelmistotuotannon kohdealue**

**Ohjelmistojen tuotteistamisessa** ja tuoteperheiden kehittämisessä voidaan hyödyntää tuotelinja-arkkitehtuureja (product line architectures). Tuotelinja-arkkitehtuurissa saman ohjelmistoperheen jäsenille rakennetaan yhteinen tuotealusta, joka sisältää tuotteille yhteiset peruskomponentit. **Tuotelinja-arkkitehtuurien rakentaminen** voidaan jakaa kahteen rinnakkaiseen vaiheeseen 1) arkkitehtuurin rakentamiseen, jossa analysoidaan sovellusaluetta ja tuotetaan analyysin perusteella yhteinen tuoterunko, 2) sovelluksen rakentamiseen, jossa lisätään tuotteeseen asiakkaan toivomat erityispiirteet [Bos00]. Keskeinen vaatimus uuden **tuotteen hyväksynnälle** (acceptability) on ohjelmiston helppo yhteen liittäminen muiden ohjelmistotuotteiden kanssa.

**Palvelukeskeisessä ohjelmistoliiketoiminnassa** ohjelmistoyritysten asiakkaat ovat 1) perinteisiä asiakkaita, jotka ostavat lisenssin ohjelmistotuotteelle (ml. lisäpalvelut kuten ylläpitoa, koulutusta ja asennuspalveluja), 2) palveluntarjoajia (service organizations or operators), jotka markkinoivat ja myyvät palveluja omille asiakkailleen tai 3) sovellusintegraattoreita, jotka kehittävät hankituista ja itse tuotetuista ohjelmistokokonaisuuksista (paketti, komponentti, palvelu) uusia kokonaisuuksia. Myös palvelun tarjoajat voivat integroida omaan palvelupakettiinsa sekä itse tuotettuja että muilta palveluntarjoajilta ostettuja tuotteita ja palveluja. Tätä tarkoitusta varten he tarvitsevat ohjelmiston, johon on helppo yhdistää ominaisuuksia, toimintoja ja tietoa eri lähteistä.

Palvelun hyödyntäjät ostavat haluamansa palvelun palveluntarjoajalta, jonka tuote vastaa parhaiten heidän vaatimuksiaan (toiminnalliset ja laadulliset vaatimukset, toteutettavuus, taloudellisuus ja tuottavuus). He yhdistelevät omaan palveluunsa juuri ne osat, jotka tuottavat heidän asiakkailleen lisäarvoa, mistä seuraa se, että erilaisia palvelukombinaatioita on suuri määrä. Tutkimusprojektin tarkoituksena on kehittää yhteistyöyrityksille **palvelusuuntautunut toimintamalli**, josta hyötyvät sekä ohjelmistotalo että ohjelmistotalon asiakkaat.

Tutkimuksen tavoitteena on parantaa pk-ohjelmistoyritysten ja asiakasyritysten kilpailukykyä kehittämällä ohjelmistotuotteiden ja -palvelujen laatua sekä yritysten verkottumista. Tunnistetaan ja poistetaan ohjelmistoyrityksen ja asiakasyrityksen välisessä yhteistyössä sijaitsevat ongelmakohdat yhteistyötä ja verkottumista kehittämällä.

### 2.3. Ohjelmistotuotannon ongelmat

Asiakkaiden todellisten tarpeiden ja ohjelmistoyrityksen toteuttaman järjestelmän välillä on usein aukkoja. Neuvotteluvaiheessa ohjelmiston ominaisuudet määritellään usein epätarkasti ja odotettavissa olevia hyötyjä painottaen. Ohjelmistotuotetta ikään kuin myydään visioiden avulla. Käyttöön otettu tuote sen sijaan on ominaisuuksiltaan tinkimätön. Puutteet, vajavaisuudet ja käytettävyysongelmat näkyvät konkreettisina liiketoimintaa haittaavina piirteinä. Ohjelmistosta puuttuu olennaisia ominaisuuksia, jotka asiakkaan mielestä kuuluvat järjestelmään automaattisesti, mutta ohjelmistotuottaja vetoaa siihen, ettei ominaisuuksia ole toteutettu, koska niistä ei ole määrittelyissä sovittu. Kun vielä toimintaprosessien kehittämiseen ja ohjelmiston käyttöönottoon panostetaan riittämättömästi, on seurauksena tyytymätön asiakas, joka mielellään vaihtaa ohjelmistotoimittajaa. Ohjelmistotuottajat uupuvat yrittäessään viimehetkellä parantaa tuotetta pikakorjausten avulla. Ohjelmistoprojekteissa painotetaan liikaa toimitusaikaa (dead line, time to market) ja lyhyen tähtäimen taloudellisia tavoitteita (budget) ohjelmistojen pitkätähtäimen kehityksen kustannuksella [Szy99]. Toimintatapa ei mahdollista ohjelmistotalon ja asiakkaan liiketoiminnan ja vuorovaikutuksen kehittymistä. Erityisesti tästä kärsivät pk-yritykset.

Unified Modeling Language (UML) on ohjelmistojen mallinnuskieli, jota useat ohjelmistotuottajat käyttävät vaatimusten mallintamisessa ja järjestelmän rakenteen ja toiminnallisuuden kuvaamisessa. Tyypillisellä asiakkaalla ei ole mitään kokemusta tietojärjestelmien mallintamisesta eikä myöskään mallinnuskielistä, kuten UML. Asiakas antaa ohjelmistovaatimukset epätarkkoina ja osin puutteellisina kuvauksina, joita ohjelmistoyrityksen edustaja (esimerkiksi systeeminmäärittelijä) yrittää tulkita. Mikäli ohjelmistoyrityksellä ei ole kokemusta toimialalta tai sen erityispiirteistä, asiakkaan ja ohjelmistoyrityksen kommunikoinnissa esiintyy väistämättä ongelmia. Toisaalta ohjelmistojen mallinnukseen hyvin sopiva kieli, kuten UML, ei anna riittäviä välineitä ohjelmiston ominaisuuksien havainnollistamiseen asiakkaalle. Tähän tarvittaisiin havainnollistavia kuvia ohjelmistoarkkitehtuurissa noudatetusta ideasta tai jopa mielellään kokeiluympäristöä (demonstration, prototypes).

Projektimuotoinen ohjelmistoliiketoiminta on altis monille yrityksen sisäisille ja ulkoisille riskitekijöille. Tutkimusten mukaan ohjelmistoprojektien tyypillisimpiä ongelmia ovat [Per91, Cra99, HaM97]:

- aikataulun pettäminen
- kustannusten ylittyminen
- ohjelmiston käyttöönottovaiheessa ilmenevät toiminnalliset ja laadulliset puutteet
- asiakkaan tyytymättömyys tuotteeseen
- projektin jälkihoidon valtava määrä
- yritysten osastojen välinen kilpailu
- puutteellinen ohjelmistotuotantoprosessi
- puutteellinen toimialakokemus
- puutteellinen uuden teknologian (esim. komponentit ja suunnittelumallit) osaaminen.

Projektissa kohdattujen ongelmien syitä ovat muun muassa

- puutteellinen määrittely
- virheelliset työmääräarviot
- muutosvastarinta ja muutoshallinta
- valittu liian suuri projekti resursseihin nähden
- muutos yrityksen strategiassa, josta on seurauksena projektin tarpeettomuus

- asiakkaan tai toimittajan asiantuntemattomuus
- suunnittelematon käyttöönotto ja käyttöönottovaiheen riittämätön resurssointi
- henkilöstön vaihtuminen joko asiakkaan tai toimittajan puolella
- projektin yhteistoiminnan ongelmat (projektipäällikkö, projektin jäsenet)
- ongelmat työvälineissä ja laitteissa

### 3. Tavoitteet

#### Keskeiset tavoiteltavat tulokset:

1. Konkreettisimmillaan Sose auttaa tunnistamaan kehityskelpoiset ohjelmistotuotteet tai tuotelinjat sekä tuottaa menetelmän integraatiopisteiden määrittelemiseksi. Kehitetään alustariippumattomat ja alustariippuvat mallit (PIM ja PSM) ohjelmistotuotteiden tuottamista varten. SOSE auttaa ohjelmistoyritystä sen kehittäessä liiketoimintaansa ja rakentaessa kaupallisia ohjelmistotuotteita.
2. Käytännössä tuloksena saadaan testatut menetelmät ja tekniikat **palvelukeskeisen ohjelmistoliiketoiminnan** kehittämiseksi **teollisuuden ja kaupan toimialalle**. Tavoitteena on parantaa Pohjois-Savon alueella vaikuttavien pk-yritysten (ohjelmistotuottajat, palvelutuottajat ja asiakasyritykset) kilpailukykyä ja työllisyyttä. Tekniikan pitää kehittää **verkottuneiden organisaatioiden yhteistoimintaa** vaatimustenhallinnan, prosessi-integraation ja sovellusintegraation sekä ohjelmistoarkkitehtuurin ja suunnittelumallien avulla.
3. Yleisemmällä tasolla tuotetaan käytännössä testatut **menetelmät palvelujen määrittelemiseen, koostamiseen ja integrointiin**. Menetelmien tulee perustua vähittäiseen määrittelyyn, eli tekniset yksityiskohdat ratkaistaan viimeisessä vaiheessa (early design and late binding). Kaikki tulokset ovat mahdollisimman laajalti **yhteensopivia kansainvälisten ja toimialariippumattomien standardien kanssa**. Tulokset tukevat **ohjelmistoliiketoiminnan kannattavuuden, tehokkuuden ja tuottavuuden kehittymistä**.
4. Palvelu- ja komponenttirajapintojen tulee edistää Suomessa tuotettujen sovellusten, komponenttien ja palvelujen kilpailukykyä ja vientiä. Lopputuloksena asiakasorganisaatiot voivat **kehittää kokonaisjärjestelmäänsä vähittäin** uusia palveluja ja komponentteja hankkimalla. Helpompi integrointi kokonaisjärjestelmään kasvattaa kannattavuutta sekä Suomessa että vientimarkkinoilla.

#### Tavoitteet ohjelmistoyritysten kannalta:

- Lähiajan tavoitteena on kehittää tekniikat ja menetelmät tuote- ja palvelukeskeisen arkkitehtuurin ja ohjelmistorakenteen määrittelyyn sekä prosessi- ja sovellusintegraation toteutukseen. Tämä mahdollistaa tuote- ja palvelukeskeisen ohjelmistoliiketoiminnan kehittymisen ja prosessia tukevien palvelujen ”lainaaminen” sovellusten kesken. Edelleen näin voidaan tehostaa ohjelmistotuotantoa, kehittää yhteistyötä ohjelmistoyritysten ja asiakasyritysten muodostamassa toimijaverkossa, helpottaa sovellusten käyttöönottoa, mahdollistaa ominaisuuksien hankkiminen sovelluksiin muiden tuottamina palveluina tai komponentteina, myydä omaa ydinosaa helpommin integroitavina kokonaisuuksina.
- Pidemmän aikavälin tavoitteena on edistää ohjelmistoliiketoiminnan kehittymistä Suomessa sekä vientikelpoisuutta muihin maihin. Viime kädessä näin edistetään yritysten elinkelpoisuutta ja ohjelmistoammattilaisten työllisyyttä Suomessa.

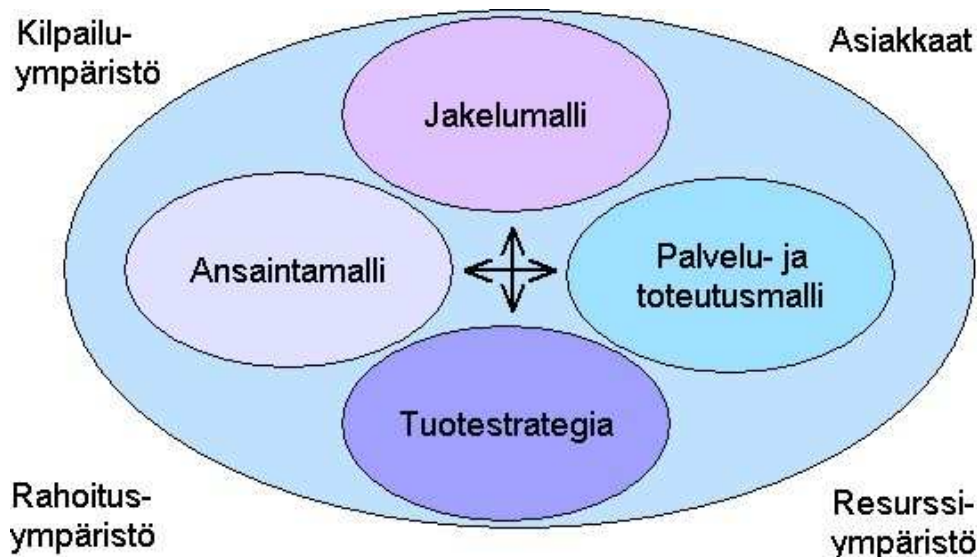
#### Tavoitteet ohjelmistoyritysten asiakasorganisaatioiden kannalta:

- Lähiajan tavoitteena on kehittää verkottuneiden organisaatioiden yhteistoimintaa tietojärjestelmien kehittämisessä ja hankinnassa sekä pienentää uusien sovellusten käyttöönottokustannuksia ja -aikaa.
- Pidemmän aikavälin tavoitteena on luoda riittävän avoimet ja tehokkaat toimijaverkot, jotka yhteistyössä kehittävät liiketoimintaprosesseja ja tietojärjestelmiä. Tämä parantaa työllisyyttä ja kilpailukykyä.

Kaikki tuloksena syntyvät **julkaisut ja menetelmäkuvaukset** ovat julkisia dokumentteja. Tutkimuksen tuloksista julkaistaan vuosittain artikkeleita kansainvälisissä tieteellisissä lehdissä ja konferensseissa, esityksiä kotimaisilla ammatillisilla foorumeilla, sekä raportteja ja opinnäytetöitä. Tutkijat myös **luennoivat** omista teemoistaan perus- ja täydennyskoulutuskursseilla. Tutkimusaineistosta tuotetaan myöhemmin **kaksi väitöskirjaa** tai lisensiaatintyötä.

### Ohjelmistoliiketoiminnan viitekehyksen rakentaminen

Ohjelmistoliiketoiminnan viitekehyksen rakentamisen tavoitteena on tutkimusprojektin tulosten organisointi yhdeksi kokonaisuudeksi. Viitekehyksen pohjana käytetään kuvassa 5 esitettyä ohjelmistoliiketoiminnan mallia [RaR95].



**Kuva 5: Ohjelmistoliiketoiminnan malli** mukaellen [RaR95].

Tuotestrategiassa kuvataan yrityksen ydintuote tai tuotekehityksen painopiste. Tuotekehitysmallissa selvitetään, miten yritys on järjestänyt ydintuotteensa kehittämisen. Palvelun ja toteutuksen malli sisältää sen, miten ydintuote saatetaan loppukäyttäjälle toimivana ratkaisuna. Tässä mallissa keskeinen käsite on palveluaste, jossa matalalla palveluasteella tarkoitetaan itsepalvelua ja korkealla palveluasteella konsultointia ja koulutusta.

Jakelumalli sisältää tuotteiden toimituskanavan lisäksi myös markkinoinnin kanavan, tilauksen kanavan, rahoituksen kanavan ja tiedonsiirron kanavan toimittajan ja asiakkaan välillä. Jakelumalliin kuuluvia markkinointipäätöksiä on muun muassa se, miten tuote- ja palveluyhdistelmän markkinointi ja jakelu on organisoitu ja ketkä ovat kyseisen tarjonnan myyjiä ja markkinoijia. Ansaintamalli kuvaa tavat, joilla yritys muodostaa tulorahoituksensa. Siinä selvitetään, mistä lähteistä ja millä tavoin

ohjelmistoliiketoiminnan voitot ja tulorahoitus hankitaan [Hyv03].

## Viitteet

- [Bos00] Bosch, J.: *Design & Use of Software Architectures Adopting and evolving a product-line approach*. Addison Wesley, 2000.
- [Cra99] Crawford, D.: Software Project Failure Lessons Learned, *Communications of the ACM*, Volume 42, Number 11 (1999), Pages 21-24, 1999.
- [Gib94] Gibbs, W.: Software's Chronic Crisis, *Scientific American*, September 1994, 1994.
- [HeS00] Herzum P., Simms O.: *Business Component Factory*, OMG Press, 2000.
- [Hyv03] Hyvönen, E.: *Ohjelmistoliiketoiminta*. WSOY, 2003.
- [JGP97] Jacobson I., Griss M., Johnson P.: *Software Reuse (Architecture, Process, and Organisation for business Success)*, Addison-Wesley, 1997.
- [Imm03] Immonen M.: *Käytettävyyden suunnittelu ja rakentaminen ohjelmistoprosessissa*, Kuopion yliopisto, 2003
- [Jän03] Jäntti M.: *Testitapausten suunnittelu UML-mallinnuksen avulla*, Kuopion yliopisto, 2003.
- [Kar03] Karhunen H.: *Arkkitehtuurikeskeinen lähestymistapa tavoitteena ohjelmiston parempi laatu*, Kuopion yliopisto, 2003.
- [Lat03] Latikka T.: *Hajautetut järjestelmät*, Kuopion yliopisto, 2003.
- [Lin00] Linthicum, David S. *Enterprise application integration*, Addison-Wesley, 2000.
- [Myk03] Mykkänen J., Tikkanen T., Rannanheimo J., Eerola A. and Korpela M.: *Integration of Health Information Systems – Specification Levels and Collaborative Definition*, MIE 2003.
- [Myk02] Mykkänen J., Porrasmaa J., Korpela M.: A process for specifying integration for multi-tier applications in healthcare. In: [Health Data in the Information Age](#). Proceedings of the [17th International Congress of the European Federation of Medical Informatics MIE 2002](#) (25-29 August 2002, Budapest).
- [Per91] Perry, W.: *Quality Assurance for Information Systems*, QED Technical Publishing Group: Methods, Tools, and Techniques.
- [Puu03] Puumalainen K.: *Tietojärjestelmien integrointi yritysten välisen liiketoiminnan kehittäjänä*, Kuopion yliopisto, 2003.
- [RaR95] Rajala, R., Rossi, M., Tuunainen, V., Korri, S.: Software Business Models: A Framework for Analysing Software Industry, *Tekes, Technology Review* 108/2001, 2001.

- [Ses00] Sessions R.: *COM+ and the Battle for the Middle Tier*, Wiley, 2000.
- [Tim99] Timmons, Jeffrey A. *New Venture Creation; Entrepreneurship for the 21<sup>st</sup> century, 5<sup>th</sup> edition*. McGraw-Hill, 1999.
- [TyW04] Tyrväinen, P., Warsta, J., Seppänen, V.: Toimialakehitys ohjelmistoteollisuuden vauhdittajana, *Tekes, Teknologia katsaus* 151/2004, 2004.
- [Uus95] Uusitalo, L. : *Markkinointi – Johdatus perusteisiin*, OTAVA, Keuruu, 1995, s. 150-157.
- [Szy99] Clemens Szyperski: *Component Software Beyond Object-Oriented Programming*, 1999
- [www001] <http://www.agilemodeling.com/index.htm> (29.1.2004)