

Riikka Ahlgren

**LIKETOIMINTAPROSESSIA KOORDINOIVA
VUOROVAIKUTUS OHJELMISTOTUOTANTOPROSESSIN
NÄKÖKULMASTA**

Tietojärjestelmätieteen
pro gradu –tutkielma
23.11.2001

Jyväskylän yliopisto
Tietojärjestelmätieteen laitos
Jyväskylä

TIIVISTELMÄ

Ahlgren, Riikka

Liiketoimintaprosessia koordinoiva vuorovaikutus ohjelmistotuotantoprosessin näkökulmasta / Riikka Ahlgren

Jyväskylä: Jyväskylän yliopisto, 2001.

94 s.

Pro gradu -tutkielma

Tässä tutkielmassa käsitellään liiketoimintaprosessia koordinoivaa vuorovaikutusta. Tavoitteena on selvittää, millä prosesseilla tutkittavaa ohjelmistotuotantoprosessia koordinoidaan. Lisäksi tavoitteena on miettiä, missä kohdissa koordinointi vaikuttaa eniten tutkittavan ohjelmistotuotantoprosessin kulkuun ja miten koordinointiin panostamalla voitaisiin ohjelmistotuotantoprosessia tehostaa yleisellä tasolla.

Ohjelmistotuotannon alalla pyritään nopeaan ja laadukkaaseen tuotantoon. Prosessia kehittämällä voidaan kehittää koko ohjelmistotuotteen laatua. Ohjelmistotuotantoprosessia kehitettäessä myös muiden prosessien vaikutus tulisi huomioida. Tässä tutkielmassa aiheen käsittely perustuu tapaustutkimukseen, jota analysoidaan SPICE-mallin avulla. Empiiristä aineistoa täydennetään aihetta käsittelevällä kirjallisuudella.

Tutkielman tuloksena on osoitettu projektinhallinnan ja asiakasvaatimusten selvittämisen olevan tärkeimpiä ohjelmistotuotantoprosessia koordinoivia prosesseja. Lisäksi on havaittu koordinoinnin olevan kriittisintä ohjelmistotuotantoprosessin alkuvaiheissa. Tärkeimpänä johtopäätöksenä on todettu, että analyysi- ja suunnitteluvaiheiden koordinointiin panostamalla voidaan tehostaa ohjelmistotuotantoprosessia kokonaisuudessaan.

AVAINSANAT: Liiketoimintaprosessi, ohjelmistotuotanto, ohjelmistotuotantoprosessi, koordinoointiteoria, prosessien välinen vuorovaikutus

SISÄLLYS

1. JOHDANTO	1
2. LIIKETOIMINTAPROSESSIT JA PROSESSIORGANISAATIO	4
2.1. Liiketoiminnan organisointi.....	4
2.1.1. Organisaatorakenteen tyyppejä.....	4
2.1.2. Prosessiorganisaation hierarkia.....	7
2.2. Tutkielman käsitehierarkia.....	9
2.2.1. Liiketoimintaprosessi.....	9
2.2.2. Organisaation perusyksikkö.....	10
2.2.3. Muita käsitteitä.....	11
2.2.4. Projektin ja prosessin erosta.....	13
2.3. Yhteenveto	15
3. OHJELMISTOPROSESSIT.....	16
3.1. Ohjelmistotuotanto ja ohjelmistoprosessit	16
3.2. Ohjelmistoprosessin yleisiä piirteitä	19
3.3. SPICE.....	20
3.3.1. Kehittämisen prosessiryhmä	22
3.3.2. Asiakas-toimittaja –prosessiryhmä	23
3.3.2.1 Asiakasvaatimusten selvittäminen.....	23
3.3.2.2 Käyttö.....	24
3.3.3. Tuen prosessiryhmä.....	25
3.3.4. Hallinnan prosessiryhmä.....	26
3.3.4.1 Projektinhallinta.....	26
3.3.4.2 Laadunhallinta ja -varmistus	28
3.3.5. Organisaation prosessiryhmä	28
3.4. Yhteenveto	29
4. PROSESSIMUOTOINEN TYÖTAPA JA SEN MALLINTAMINEN.....	31
4.1. Prosessimuotoinen työtap.....	31

4.1.1.	Prosessimuotoisuuden taustaa.....	32
4.1.2.	Informaation prosessoinnin näkökulma.....	33
4.1.3.	Lähtökohdat ja tavoitteet.....	35
4.1.4.	Tunnuspiirteitä	36
4.1.5.	Laatu prosessien näkökulmasta.....	38
4.2.	Prosessien mallintaminen.....	40
4.2.1.	Prosessi- eli vaihejakomallit.....	41
4.2.1.1	Vesiputousmalli	41
4.2.1.2	Prototyypimalli	42
4.2.1.3	Inkrementaalinen malli	42
4.2.1.4	Spiraalimalli	43
4.2.2.	Prosessimallien merkitys.....	44
4.3.	Yhteenveto	44
5.	PROSESSIA KOORDINOIVA VUOROVAIKUTUS.....	46
5.1.	Koordinointiteoria	46
5.1.1.	Yleistä teoriasta.....	47
5.1.2.	Teorian sovelluksia	48
5.1.3.	Riippuvuuksien hallinta.....	49
5.2.	Koordinoiva vuorovaikutus.....	51
5.2.1.	Koordinoivan vuorovaikutuksen käsite.....	51
5.2.2.	Vuorovaikutustilanteita	53
5.2.3.	Epävarmuuden hallinta.....	53
5.2.4.	Vuorovaikutuksen tavoitteet	55
5.3.	Informaation rooli koordinoinnissa.....	55
5.3.1.	Koordinoiva informaatio.....	56
5.3.2.	Informaatiotulvan ongelma	58
5.4.	Yhteenveto	59
6.	TUTKIMUSASETELMA.....	61
6.1.	Tutkimusmalli.....	61

6.2. Tapauksena keskisuuri ohjelmistoyritys	63
6.2.1. Tutkittava ohjelmistoprosessi.....	64
6.2.2. Tutkimusmenetelmä ja -prosessi.....	65
6.3. Tutkimusaineiston kuvailu	68
6.3.1. Koordinoivat prosessit tutkitun ohjelmistoprosessin eri vaiheissa	68
6.3.2. Koordinointikeinot.....	70
6.3.3. Tutkitun ohjelmistoprosessin erityispiirteitä.....	71
6.4. Yhteenveto	73
7. TUTKIMUSTULOSTEN JOHTAMINEN JA ANALYSOINTI.....	74
7.1. Millaista on ohjelmistoprosessia koordinoiva vuorovaikutus?	74
7.1.1. Koordinoivat SPICE:n prosessit.....	75
7.1.2. Koordinointisuhde projektinhallintaprosessiin.....	78
7.2. Johtopäätökset	79
7.3. Tutkimustulosten rajoitteita.....	80
7.4. Yhteenveto	82
8. YHTEENVETO	85
LÄHTEET.....	87

1. JOHDANTO

Ohjelmistotuotantoprosessin etenemiseen vaikuttavat monet asiat, muun muassa käytetty prosessimalli ja muut yrityksen liiketoimintaprosessit. Siksi ohjelmistotuotantoprosessia kehitettäessä tulisi huomioida myös sitä koordinoivat prosessit. Nyt ja tulevaisuudessa ohjelmistoja tarvitaan liiketoiminnan tukemiseen ja uusien tuotteiden kehittämiseen. Kilpailun koventuessa ohjelmistoteollisuuden markkinoilla pitää pystyä nopeisiin toimitusaikoihin ja laadukkaisiin tuotteisiin. Samalla ohjelmistotuotannon kehittäminen tulee olla kokonaisvaltaista, mikä tarkoittaa huomion kiinnittämistä laadun kolmeen eri alueeseen: tuotteisiin, toimintatapoihin eli prosesseihin ja resursseihin. (Nevalainen 1999) Prosessilaatu on juuri nyt ajankohtainen, sillä prosessilaadun parantaminen voi samalla nostaa myös tuotteiden ja resurssien laatua (Nevalainen 1999; Chroust 2000). Prosessia voidaan kehittää panostamalla sen vuorovaikutukseen muiden prosessien kanssa. Prosessin kehittämiseen tehdyt taloudelliset investoinnit saadaan nopeasti takaisin liiketoiminnan hyötyinä. (Nevalainen 1999)

Tutkielman aihe ohjelmistotuotantoprosessia koordinoivasta vuorovaikutuksesta on noussut esiin Jyväskylän yliopiston Tietotekniikan tutkimusinstituutissa käynnissä olevan PISKO-projektin tiimoilta. Kolme vuotta kestäneen PISKOn päätavoitteina on ollut kehittää ohjelmistokomponentin kuvaus-, suunnittelu-, ja toteutusmenetelmä sekä tehostaa hankkeessa mukana olevien yritysten ohjelmistoprosessia komponenttilähestymistavan avulla. PISKOssa on pääosin keskitytty ohjelmistoprosessin sisäisiin tapahtumiin, eikä ohjelmistoprosessin yhteyksiä muihin yrityksen prosesseihin ole huomioitu. Tämän tutkielman myötä ohjelmistoprosessista saadaan entistä kokonaisvaltaisempi kuva, kun sen vuorovaikutussuhteet tulevat paremmin tunnetuksi.

Ohjelmistotuotantoprosessilla (engl. Software engineering process) (myöhemmin ohjelmistoprosessi) tarkoitetaan tässä kaikkia niitä ohjelmiston kehitystehtäviä, joita

tarvitaan tekemään asiakkaan vaatimuksista toimiva tuote (Humphrey 1988; Sommerville 1995).

Prosessia koordinoiva vuorovaikutus on määritelty prosessien keskinäisen tiedon tai tuotteen vaihdoksi, jolla voidaan vaikuttaa prosessin etenemiseen. Tieto tai tuote voi olla prosessin synnyttämä tulos tai sen tarjoamaa palautetta, jota vastaanottava prosessi käyttää syötteenään tai ohjaustietonaan.

Ohjelmistoprosessia kehitettäessä yleensä kiinnitetään huomiota vain prosessin sisäisiin tietovirtoihin ja tehtäviin. Harvoin huomioidaan muiden prosessien tarjoamaa koordinoivaa vuorovaikutusta. Tämä johtuu siitä, että vaikka ohjelmistotuotannosta on kirjoitettu paljon (esim. Pressman 1997; Haikala & Märijärvi 1998; Sommerville 1995) ja ohjelmistoprosessin sisältöön on esitetty monia eri näkökulmia (esim. Osterweil 1999), ei ohjelmistoprosessin yhteydestä muihin yrityksen liiketoimintaprosesseihin ole kirjoitettu tarpeeksi. Ohjelmistotuotannon kirjallisuus ei kerro riittävästi ohjelmistoprosessin yhteyksistä muihin yrityksen liiketoimintaprosesseihin ja prosessia koordinoivasta vuorovaikutuksesta.

Tämän tutkimuksen tavoitteena on selvittää, millä prosesseilla tutkittavaa ohjelmistoprosessia koordinoidaan ja missä ohjelmistoprosessin vaiheissa koordinointi vaikuttaa eniten tutkittavan prosessin etenemiseen. Lisäksi tavoitteena on pohtia, miten koordinointiin panostaminen voi tehostaa ohjelmistoprosessia. Pääpaino tutkielmassa on ohjelmistoprosessin vuorovaikutuksessa. Prosessien uudelleenjärjestelyä, johtamista ja mallintamista sivutaan lyhyesti.

Tutkimuksella etsitään vastauksia seuraaviin kysymyksiin:

- Millä prosesseilla tutkittavaa ohjelmistoprosessia koordinoidaan?
- Missä ohjelmistoprosessin kohdissa koordinointi vaikuttaa eniten tutkittavan ohjelmistoprosessin kulkuun?
- Miten koordinointiin panostaminen näissä kohdissa voisi tehostaa ohjelmistoprosessia yleisesti?

Tutkimusaihetta lähestytään liiketoiminnan organisoinnin ja prosessimuotoisen työtavan kautta. Erityisesti tarkastellaan ohjelmistotuotantoa ja ohjelmistoprosessin suorittamista. Empiirisen tutkimuksen pohjana on Malonen ja Crowstonin (1994) koordinoointiteoria, jonka mukaan koordinointi on riippuvuuksien hallintaa. Teorian avulla riippuvuudet jaotellaan luokkiin ja tutkimuskohdetta rajataan luokkien mukaan. Tutkimus tehdään etsimällä ohjelmistoprosessista kohtia, joissa vuorovaikutusta muihin prosesseihin esiintyy ja tunnistamalla koordinoivat prosessit. Prosessin tehokkuus on tässä määritelty prosessin lyhyemmäksi läpimenoajaksi.

Empiirinen tutkimus suoritetaan kohdeyrityksessä ryhmähaastatteluilla. Haastattelukysymykset pohjautuvat Crowstonin tutkimukseen (1997). Menetelmää täydennetään dokumenttien analyysillä. Aineiston kuvaamisessa käytetään apuna Milesin ja Hubermanin (1994, 112 ja 121) ja Harrisonin (1998) malleja ja tuloksia analysoidaan SPICE:n pohjalta. Tutkielman tärkeimpinä tuloksina havaittiin ohjelmistoprosessin koordinoinnin olevan kriittisintä sen alkuvaiheissa. Tärkeimmät koordinoivat prosessit ovat projektinhallinta ja asiakasvaatimusten selvittäminen. Tutkimuksen johtopäätöksenä todetaan, että panostamalla ohjelmistotuotantoprosessin analyysi- ja suunnitteluvaiheiden koordinointiin ohjelmistotuotantoprosessia kokonaisuudessaan voidaan tehostaa.

Tutkielma jakaantuu kahdeksaan lukuun. Toisessa luvussa tutustutaan liiketoiminnan organisoinnin tapoihin ja erityisesti prosessimuotoiseen organisaatorakenteeseen. Tutkimuksen tärkeimmät käsitteet määritellään. Kolmannessa luvussa tutustutaan ohjelmistotuotantoon erilaisten viitekehysten kautta. Ohjelmistotuotannon prosesseihin tutustutaan ISO –standardiin perustuvaan SPICE –mallin avulla. Neljännessä luvussa käydään läpi prosessimuotoista työskentelytapaa ja sen käsitteitä. Lisäksi sivutaan prosessien mallintamista. Koordinoointiteoriaan ja prosessia koordinoivaan vuorovaikutukseen tutustutaan viidennessä luvussa. Empiirisen tutkimuksen kulku ja käytetyt menetelmät raportoidaan kuudennessa luvussa. Tutkimusaineistoa kuvaillaan lyhyesti. Seitsemännessä luvussa raportoidaan tutkimuksen tulokset. Tuloksista vedetään johtopäätökset ja niitä tarkastellaan kriittisesti. Kahdeksas luku vetää yhteen kaiken edellä kirjoitetun.

2. LIKETOIMINTAPROSESSIT JA PROSESSIORGANISAATIO

Organisaatiot muodostuvat eri tavoin riippuen niiden toimialoista, perinteistä ja toimintatavoista. Tässä luvussa ensin tutustutaan lyhyesti kolmeen eri organisaatorakenteeseen: prosessimuotoiseen, funktionaaliseen ja matriisiorganisaatioon. Prosessiorganisaatiota ja sen muodostumista käsitellään tarkemmin. Sen jälkeen määritetään liiketoimintaprosessi ja muita prosesseihin liittyviä käsitteitä. Erityisesti huomioidaan projektin ja prosessin välinen ero.

2.1. Liiketoiminnan organisointi

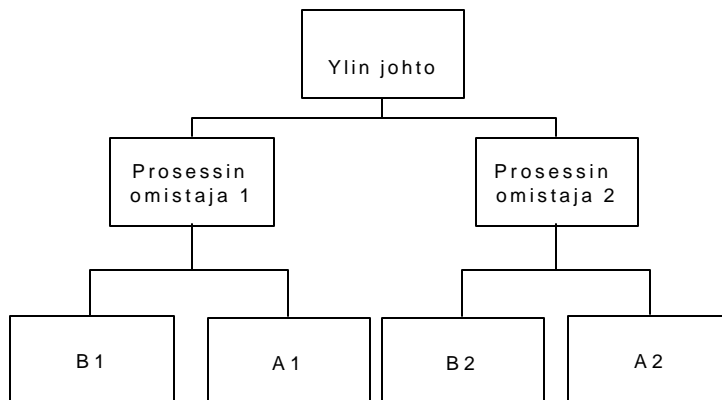
Liiketoiminnan organisoinnilla tarkoitetaan rakennetta, jolla organisaatiossa pyritään liiketoiminnan keskeisiin tavoitteisiin. Organisaatioteoriassa rakenteita on tutkittu jo 1900-luvun alusta alkaen (Taylor 1907, Marchin ja Simonin, 1959, 12, mukaan). Myöhemmin Burns ja Stalker tunnistivat kaksi vallitsevaa tapaa organisoida liiketoimintaa, mekanistisen ja orgaanisen. (Burns & Stalker 1961) Liiketoiminnan organisoinnin perustana on, ettei mikään rakenteista ole toista parempi vaan rakenteen tehokkuuden määrää sen sopivuus organisaatioon ja sen ympäristöön (Galbraith 1973).

2.1.1. Organisaatorakenteen tyyppejä

Organisaation rakenteet vaihtelevat organisaatioiden koon, toimialan ja tehtävien mukaan. Rakenteessa on pohjimmiltaan kyse tehtävien niputtamisesta ensin yksilön työksi ja edelleen hallinnollisiksi yksiköiksi. Nämä yhdistetään suuremmiksi yksiköiksi, jotta saadaan organisaation laajuinen jako. Vaikeus rakenteen muodostamisessa on sellaisen ryhmittelyn toteuttaminen, että myös alimman tason tehtävien suorittaminen on taloudellista. (March ja Simon 1959)

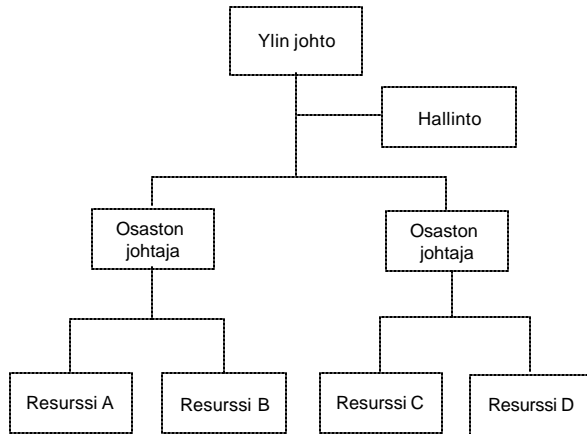
Taloudellisuutta haettaessa tehtävien ryhmittelyssä on yleensä päädytty joko prosessimuotoiseen tai toimintolähtöiseen eli funktionaaliseen malliin. Kuviossa 1 on kuvattu prosessimuotoinen organisaatorakenne, jossa työntekijät sijoittuvat prosessien, ei työtehtävien, mukaan. Rakenteen etuina ovat joustava ja käytännönläheinen toimintatapa ja matala hierarkia (Karvonen 1999).

Prosessiorganisaatiossa työntekijät A1 ja B1 tekevät eri tehtäviä samassa prosessissa ja ovat riippuvaisia toistensa työn etenemisestä. He ovat vastuussa työstään yhteiselle esimiehelle, joka on heidän prosessinsa omistaja. Työntekijä A1 tekee samaa tehtävää kuin A2, mutta he työskentelevät eri prosessissa. He raportoivat kumpikin oman prosessinsa esimiehelle. (Rotemberg 1999) Riippuen liiketoiminnan alasta, yrityksen prosessit ja niiden sisältämät tehtävät voivat olla keskenään hyvin erilaisia.



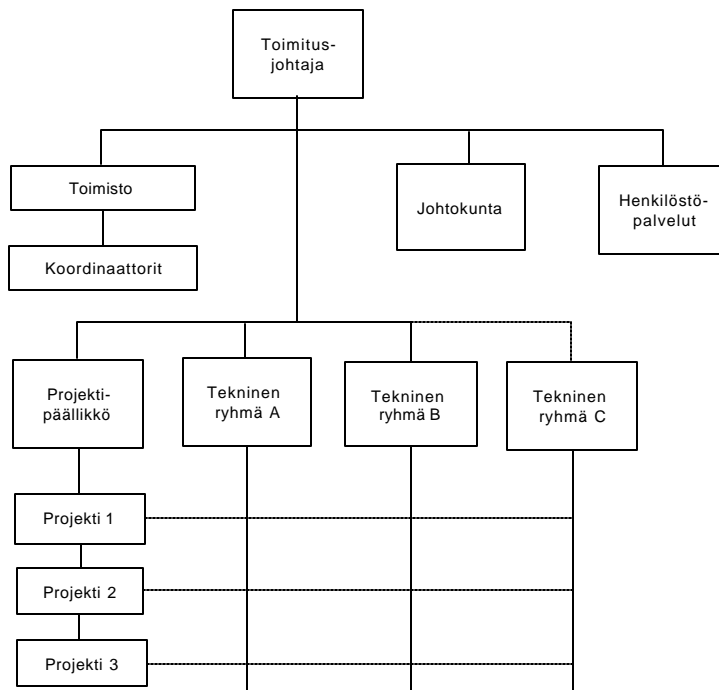
KUVIO 1. Raportointisuhteet prosessiorganisaatiossa (Rotemberg 1999, 458)

Vertailun vuoksi esitetään kuviossa 2 funktionaalinen organisaatorakenne, joka kehitettiin teollisuuden noustessa 1900-luvun alkupuolella (ks. Pandya, Karlsson, Sega & Carrie 1997). Rakenteen tunnuspiirteitä ovat korkea hierarkian taso ja osastojen sisäinen yhteistyö ja koordinointi (McCann & Galbraith 1981).



KUVIO 2. Funktionaalinen organisaatio (Lim 1998, 239)

Kolmas huomionarvoinen organisaation rakenne on matriisiorganisaatio, jossa on pyritty yhdistämään prosessimuotoisen ja funktionaalisen organisaatorakenteen parhaat puolet. Matriisiorganisaatio on esitetty kuviossa 3. Tällainen rakenne saattaa olla vain tilapäinen järjestely, jonka avulla tehdään ainoastaan tietty toimeksianto. Matriisiorganisaation etuna on joustavuus sekä työtehtäviin että ympäristöön sopeutumisessa. (McCann ja Galbraith 1981)



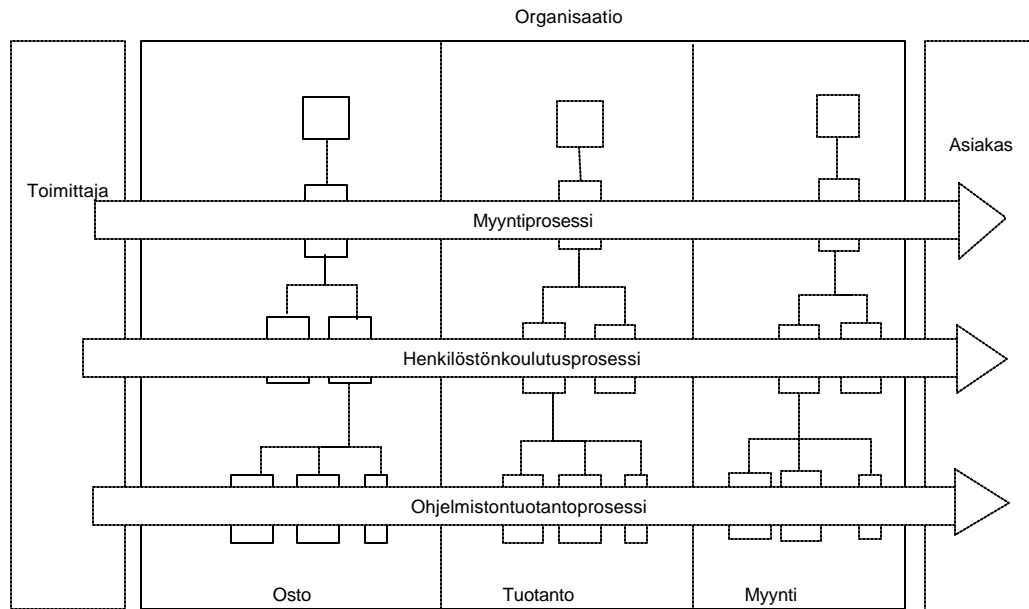
KUVIO 3. Matriisiorganisaatio (McCann ja Galbraith 1981, 62)

Kolmesta esitellystä mallista prosessimuotoinen organisaatorakenne on ajankohtainen ohjelmistotuotannossa, sillä se tukee erityisesti asiakaslähtöistä toimintaa (Karvonen 1999), johon ohjelmistoteollisuudessa yhä enemmän pyritään. Ohjelmistojen tekemisestä tulee koko ajan myös haasteellisempaa kun ohjelmistojen koot kasvavat, monimutkaisuus lisääntyy ja markkinoilletuloajat ja tuotteiden elinkaaret lyhenevät. Prosessilähtöisellä työtavalla toimintaa on yritetty saada läpinäkyvämmäksi ja samalla paremmin hallittavaksi. (ks. Saukkonen & Oivo 1998, 11)

Jokelan mukaan (1999) prosessimuotoinen työn organisointi on yrityksen liiketoimintatapojen ja -mallien yhtenäistämistä ja kehittämistä asiakkaan näkökulmasta käsin. Tällöin ollaan kiinnostuneita nimenomaan siitä, *miten* työ tehdään. Vaikkei prosessimuotoisuus ole uusi asia, vielä muutama vuosi sitten useat eurooppalaiset yritykset perustivat organisaatorakenteensa funktionaalisen mallin mukaan (Pandya ym. 1997). Davenport tiivistää (1993), että yritys tuottaa lisäarvoa asiakkailleen juuri prosessirakenteella ja sen mukanaan tuomalla joustavalla työtavalla. Pohjimmiltaan prosessiorganisaatiossa on kyse päätäntävällän siirtämisestä hierarkian alimmalle mahdolliselle tasolle (Galbraith, 1973, 12).

2.1.2. Prosessiorganisaation hierarkia

Matala ja joustava hierarkia mahdollistaa nopean päätöksenteon ja sen myötä vaakasuuntaisen toimintatavan. Kuviosta 4 näkyy, miten yrityksen liiketoimintaprosessit lävistävät koko organisaation poikittain koskien useita yrityksen toimintoja. Usein prosessit jatkuvat tavarantoimittajan ja asiakkaan organisaatioihin. Hierarkialtaan prosessiorganisaatiot ovat prosessien ympärille rakentuneita.



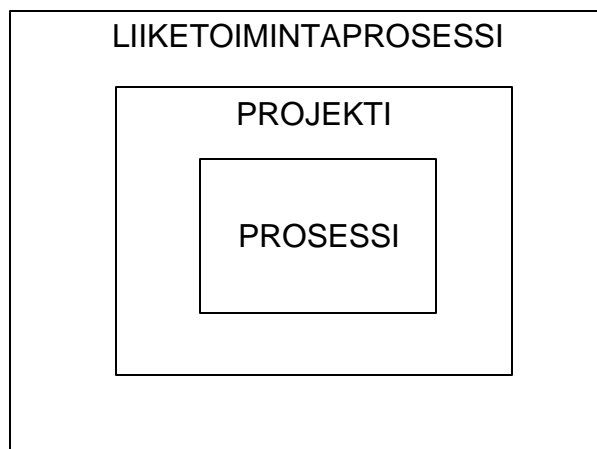
KUVIO 4. Prosessiorganisaation hierarkia (muokattu Laamasen & Tinnilän 1998, 7 mukaan)

Yhdessä prosessissa työskentelee yhteistyössä useiden eri alojen asiantuntijoita ja kukin prosessi muodostaa oman palvelu- tai tuotantoyksikkönsä. Samalla vanhojen toimintorajojen ylittäminen tuo vaikeuksia (Pandya ym. 1997), joten myös yrityksen johdon tulee osallistua prosessirakenteen kehittämiseen. Tällöin prosessit saadaan kauttaaltaan yrityksen linjan ja toimintatapojen mukaisiksi. Kokonaisnäkömyksen muodostamiseen tarvitaan prosessijohtamista, jonka tavoitteena on tuotteisiin tai palveluihin liittyen funktio- tai osastojaon läpi vaakasuoraan kulkevien prosessien kokonaisvaltainen hallinta (Laamanen & Tinnilä 1998).

Prosessijohtaminen mahdollistaa ennen kaikkea resurssien tarkemman kohdistamisen. Laadun ja määrän hajontaa pystytään paremmin pienentämään ja organisaation ongelmakohdat ja vahvuudet on helpompi paikallistaa, toimintojen kehittämisestä tulee luontevaa. Kun prosessimuotoinen työtapa viedään loppuun asti, tuloksena on organisaatio, jossa pystysuora tehtävähierarkia on korvattu vaakasuoralla prosesseihin perustuvalla rakenteella (Karvonen 1999).

2.2. Tutkielman käsittehierarkia

Kuvion 5 käsittehierarkiassa liiketoimintaprosessi on yläkäsite, joka kattaa kaikki yrityksen prosessit. Laamasen ja Tinnilän mukaan *liiketoimintaprosessi* (engl. Business process) on joukko loogisesti toisiinsa liittyviä toimintoja ja niiden toteuttamiseen tarvittavia resursseja, joiden avulla toteutetaan liiketoiminnan tulokset. Kuvion keskellä oleva prosessi voi olla mikä tahansa yrityksen liiketoimintaprosesseista tai niiden aliprosesseista, esimerkiksi ohjelmistoprosessi. Tässä tutkielmassa *prosessi* määritellään useiden tehtävien sarjaksi, jonka lopputuloksena muodostuu ennalta päätetty tulos (Karvonen 1999). Muita prosesseja ovat esimerkiksi markkinointiprosessi ja laskutusprosessi. Prosessit voidaan toteuttaa *projektissa*, mihin palataan luvussa 2.2.4.



KUVIO 5. Tutkielman käsittehierarkia.

2.2.1. Liiketoimintaprosessi

Organisaatioista ja yrityksistä puhuttaessa prosessista käytetään usein sanaa liiketoimintaprosessi. Tällä halutaan painottaa sekä prosessin kriittisyyttä yrityksen toiminnalle että prosessin ulottumista vaakasuunnassa poikki koko organisation, toimittajalta asiakkaalle. (Karvonen 1999)

Kirjallisuudessa ja tieteessä liiketoimintaprosessi -käsitettä käytetään monissa eri yhteyksissä, usein hieman eri merkityksillä. Käsitteeseen liitettyjä muita termejä ovat esimerkiksi liiketoimintojen uudelleenjärjestely (engl. Business Process Re-engineering, (BPR)), kokonaisvaltainen laatujohtaminen (engl. Total Quality Management (TQM)) ja just-in-time –ajattelu (JIT) (ks. esim. Pritchard & Armistead 1999). Operaatioiden johtamisella (engl. Operation Management) on myös oma käsityksensä liiketoimintaprosesseista (ks. Armistead, Harrison & Rowlands 1994), samoin käsite esiintyy organisaatioteorian alalla (ks. esim. Thompson 1974; Mangham 1978). Joskus liiketoimintaprosessin määrittelyissä painotetaan osasto- ja funktiorajojen ylittämistä ja asiakkaan tarpeista lähtemistä. Prosessin lopputuloksena tuotetaan asiakkaalle tuote, tulos tai palvelu. Tällöin liiketoimintaprosessi noudattaa U-käyrän muotoa, sillä se päättyy omaan alkupisteeseensä, eli asiakkaaseen, jolta saadaan toiminnan syöte ja jota varten tulos tuotetaan. (Karvonen 1999)

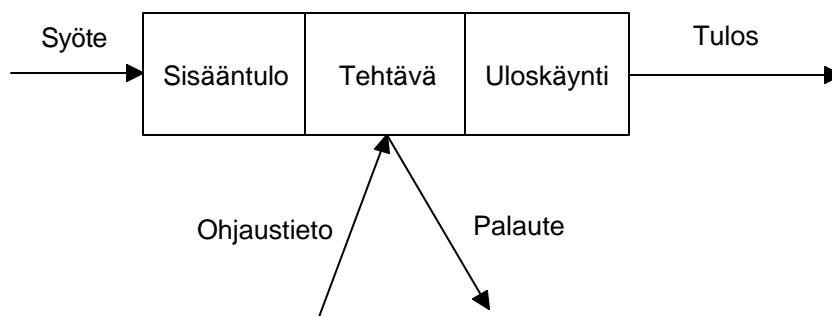
2.2.2. Organisaation perusyksikkö

Prosessi on prosessiorganisaation perusyksikkö ja pienin tuotantoyksikkö (Kock & McQueen 1996). Prosessi voi olla synonyymi liiketoimintaprosessille tai se voi olla yksi liiketoimintaprosessin osa. Suurin osa prosesseista käsittää prosessin tuotteita, tavarantoimittajia, asiakkaita, työtehtävissä käytettäviä välineitä ja työtehtävien välisiä suhteita (Kock & McQueen 1996).

Prosessi -sanaa voidaan ylipäättään käyttää melkein mistä tahansa toiminnosta, jolla on jokin syöte ja jossa on sarja toimintoja, joiden aikana saavutetaan jonkinlainen tulos. Curtisin, Kellnerin ja Overin mukaan (1992) prosesseille on ominaista ihmisen toiminta, ei niinkään koneen. Prosessi voi olla näkyvää, esimerkiksi talon maalaus, tai näkymätöntä kuten ajattelu. Tärkeää on, että prosessin päättyessä on saavutettu asteittaista edistymistä verrattuna prosessin aloitusajankohtaan, että syötettä on prosessoitu tulokseksi.

Hyvä esimerkki syötteen prosessoinnista on minkä tahansa lomakkeen täyttö. Kun asiakas täyttää lomakkeelle pohjatiedot, esimerkiksi omat henkilötietonsa, hän antaa sen syötteeksi käsittelyprosessiin. Virkailija täyttää lomakkeen loput tiedot, esimerkiksi asianumeron. Seuraavaksi lomake toimitetaan hyväksyjälle, joka leimaa lomakkeen hyväksytyksi. Viimeisessä vaiheessa lomake palautetaan asiakkaalle, joka saa eteensä loppuun asti täytetyn lomakkeen. Prosessissa syötteenä oli asiakkaan esitäyttämä lomake ja prosessin aikana kukin toimija täyttää lomakkeesta oman osansa. Tämän prosessin lopputuloksena saadaan täytetty lomake.

Yksittäisen prosessin toimintaa on kuvattu kuviossa 6. Jokainen prosessi tarvitsee käynnistyäkseen alkusysäyksen eli syötteen. Sisääntulossa on prosessin käynnistymisen esiehdot oltava täytetty. Varsinaisen tehtävän aikana prosessiin tulee ohjaustietoja muilta prosesseilta ja samalla prosessi antaa itse palautetta muille prosesseille. Uloskäynnissä on valmiina prosessin tulos, joka voi olla joko konkreettinen tuote tai abstrakti tieto. Yhden prosessin palautteet ja tulokset voivat olla toisen prosessin syötteitä ja ohjaustietoja. (Humphrey 1989) Syötteen, ohjaustiedon, palautteen ja tuloksen lisäksi prosessikirjallisuuteen tutustuva törmää usein myös joukkoon muita käsitteitä, joista keskeisimmät määritellään seuraavaksi.



KUVIO 6. Prosessin rakenne (muokattu Humphreyn 1989, 257 mukaan)

2.2.3. Muita käsitteitä

Jokaisella prosessilla on *rakenne*, joka noudattaa yrityksen omaa malliprosessia eli *prosessimallia*. Prosessimallia muokkaamalla yrityksellä on eri tehtäviä varten

käytössään erilaisia prosesseja, joiden rakenteet poikkeavat hiukan toisistaan. Prosessin *kohde* on alue, jolla se toimii. Ohjelmistoprosessin toimintojen keskus ja kohde on ohjelmistotuotanto, asiakashallintaprosessin kohteena on asiakkaiden hallinta. (Davenport 1993)

Jokaista prosessia tulee voida *mitata* jollakin tapaa, jotta prosessin kehittymistä voidaan seurata. Mittaamisen kohteena voi olla esimerkiksi prosessin käyttämä aika ja resurssit. Prosessin tuloksia voidaan mitata esimerkiksi hyödyllisyyden, virheettömyyden ja keskinäisen yhtenäisyyden mukaan. Joka prosessilla on *omistaja*, joka pääosin huolehtii prosessin suunnittelusta, kehittämisestä ja laadukkaasta lopputuloksesta. Tärkeitä ovat myös *asiakkaat*, joilta prosessi saa tehtävien syötteet ja joille se tuottaa tuloksensa. (Davenport 1993)

Toiminto (= aktiviteetti, toimenpide, työ) on tehtävä- tai toimenpidejoukko tai -ketju, jolla saadaan aikaan tietty lopputulos. Toiminto koostuu joukosta yksittäisiä tehtäviä. *Syöte* on toimittajalta saatavaa tai hankittavaa prosessiin syötettävää tietoa tai materiaalia. Prosessin kuluessa voi syntyä *palautetietoa eli palautetta*, jonka avulla prosessin etenemistä voidaan arvioida. Yhden prosessin tuote tai palaute voi olla toisen prosessin *ohjaustieto*, jota prosessi vaatii edetäkseen tai jonka avulla sen etenemistä voidaan suunnata. Prosessin lopputuloksena syntyy tuote tai muu *tulos*, joka toimitetaan asiakkaalle. Tulos voi olla lopputuote, puolivalmiste, raportti, palvelutositte, sivutuote tai jäte. (Karvonen 1999; ks. myös. Laamanen & Tinnilä 1998)

Prosesseja voidaan luokitella monella tavalla. Yksi jako luokittelee prosessit niiden tuottaman tuloksen mukaan: prosessit tuottavat joko tuotteita, palveluita tai informaatiota (Kock & McQueen 1996). Yleisemmän jaon mukaan prosessit jaetaan ydinprosesseihin ja tukiprosesseihin sen mukaan, miten kriittisiä ne ovat yrityksen menestykselle.

Ydinprosessi on organisaatiossa olennaisin, asiakkaalle lisäarvoa tuova prosessi. Sillä on aina välitön yhteys ulkoiseen asiakkaaseen. Ydinprosessissa jalostetaan tuotetta tai palvelua ja tuotetaan sille lisäarvoa. (Laamanen & Tinnilä 1998) Usein ydinprosessit

koskevat harvempia osastoja kuin tukiprosessit, mutta halkovat poikki koko organisaation (Kock & McQueen 1996). Ydinprosessit ovat prosesseja, joiden tulos muodostaa yrityksen liiketoiminnan perustan. Ohjelmistoja tuottavassa yrityksessä ydinprosessit ovat ohjelmistotuotannon prosessit.

Tukiprosessi on ydinprosessia avustava, sille edellytyksiä luova ja sen mahdollistava toiminto, esimerkiksi sisäinen koulutus (Karvonen 1999). Tukiprosessit ovat yleensä hallinnollisia, kuten palkanlaskenta, tai ne muulla tavoin ovat tukemassa yrityksen varsinaista toimintaa, kuten esimerkiksi laadunvalvonnan ja logistiikan prosessit. Tukiprosessien estymisestä seuraa koko organisaation toiminnan asteittainen jumiutuminen ja lopulta organisaation hidas kuihtuminen (Karvonen 1999). *Avainprosessi* on organisaation menestymisen kannalta tärkeä ydin- tai tukiprosessi. Kirjallisuudessa avainprosessista käytetään myös nimeä pääprosessi. (esim. Karvonen 1999; Laamanen & Tinnilä 1998) Sekä ydin- että tukiprosesseja voidaan analysoida jakamalla ne *aliprosesseihin* ja tarkastelemalla niiden sisältämiä pienempiä tehtäviä ja tehtäväkokonaisuuksia. Esimerkiksi prototyypin valmistaminen on ohjelmistoprosessin aliprosessi.

2.2.4. Projektin ja prosessin eroista

Projekti ja prosessi ovat käsitteinä vanhoja, mutta vasta viime vuosina ne on yritysmaailmassa otettu laajamittaisesti käyttöön (ks. March ja Simon 1959, 177; Brooks 1975; Hammer & Champy 1993). Kyse on ajattelutavan muutoksesta, joka ei ole tapahtunut hetkessä. Prosessimuotoisessa työtavassa yritysjohto ei enää valvo matalan tason työtehtävien suoritusta, sillä työtehtävät määräytyvät kunkin prosessin mukaan. Prosessien suorituksen kannalta olennaisempaa on huolehtia kuhunkin projektiin sellaiset työntekijät, jotka kykenevät itse ottamaan vastuun työtapojensa ja -tehtäviensä valvonnasta ja niitä koskevasta päätöksenteosta. (ks. Galbraith 1973, 13)

Projektin ja prosessin suosioon ohjelmistoteollisuudessa ovat vaikuttaneet monet asiat. Ympäristön taholta tulevat paineet ja muut trendit markkinoilla säätelevät pitkälti

työtapojen ja näkökulmien kehittymistä ohjelmistotuotannon alalla (ks. Frank & Brownell 1989, 82). Käsitteinä projekti ja prosessi on suhteellisen helppo sekoittaa toisiinsa, sillä niillä on yhteisiä sisällöllisiä piirteitä samankaltaisen kirjoitetun ulkoasun lisäksi.

Projekti on määritelty kertaluonteiseksi asetettuihin tavoitteisiin pyrkiväksi kokonaisuudeksi, joka on sekä ajallisesti että resursseiltaan rajattu ja joka toteutetaan tätä varten luodun määräaikaisen organisaation, projektiorganisaation, avulla (Haikala & Märijärvi 1998, 201). Jokaiseen projektiin on kiinnitetty projektiryhmä, joka on vastuussa koko projektin läpiviennistä alusta loppuun. Projektin työntekijöiltä vaaditaan monen alan asiantuntemusta, sillä projektissa työskentelevät henkilöt työskentelevät samalla yhdessä tai useammassa aliprosessissa, esimerkiksi ohjelmiston suunnittelu- ja testausprosessissa. Tällä tavoin vältetään ongelmia työnjaossa ja voidaan vähentää työn koordinoinnin tarvetta (ks. Galbraith 1973, 16).

Yleisesti ottaen voidaan sanoa, että prosessi sisältyy projektiin (ks. kuvio 5 s. 10; Riihinen, Rikkilä & Lassenius 1995). Prosessi kertoo *miten* projektin määräämä asia tehdään ja prosessi toteutetaan usein projektin avulla. Projekti puolestaan määrää prosessin aloittamisesta ja muusta aikataulutuksesta. Projekti määrää kaikki ne muuttujat, joita prosessi ei määrittele, eli projektin sisällön, aikataulun ja resurssit. Tiivistäen voidaan todeta: projekti = prosessi + sisältö + resurssit + aikataulu. (Riihinen ym. 1995) Projektiin voi sisältyä useampia prosesseja, jotka kulkevat ajallisesti rinnakkain tai peräkkäin.

Jokaisella projektin prosessilla tulee olla selkeästi määritellyt tavoitteet ja prosessin oman rakenteen tulee olla sopusoinnussa projektin tavoitteiden kanssa. Prosessia pitää muokata jokaista tuotetta ja jokaista projektia varten, sillä sekä prosessi että projekti ovat ainutlaatuisia. Kaikki mukautukset ja muokkaukset tulee olla dokumentoitu ja katselmoitu. Samalla prosessin pitää olla määritelty jonkin yhteisen käytännön tai standardin mukaan siten, että sitä voidaan muuttaa kun uusia ongelmia ilmenee tai vanhat ongelmat muuttavat luonnettaan. (Humphrey 1989)

2.3. Yhteenveto

Tässä luvussa todettiin, että liiketoiminnan organisointiin on monta tapaa ja että organisaatiomuodon valintaan vaikuttaa muun muassa yrityksen toimiala ja koko. Prosessiorganisaation eduiksi on mainittu sen matala hierarkia ja joustavuus. Organisaation rakenne vaikuttaa organisaatiossa käytettyihin työtapoihin ja siihen, miten organisaatio jakautuu pienempiin osiin. Lisäksi luvussa määriteltiin tutkimuksen kannalta keskeisiä käsitteitä, joista erityisesti projektin ja prosessin ero on tutkimuksen myöhempien vaiheiden kannalta olennainen.

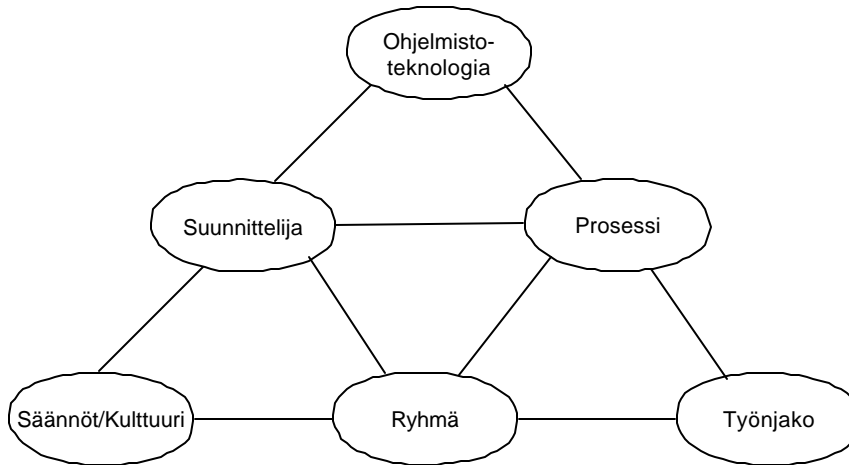
Organisaatorakenteista puhuttaessa tulee samalla muistaa, että prosessien koordinoinnin tutkiminen on samalla organisaation toiminnan tutkimista. Tällaisen tutkimuksen tuloksena organisaatiossa saatetaan huomata, että prosesseja tulee suunnitella uudelleen, mikä saattaa vaikuttaa organisaation rakenteeseen. (ks. esim. Crowston 1997) Muuttunut rakenne taas heijastuu edelleen organisaation muihin osiin, osastoihin tai liiketoiminnan alueisiin.

3. OHJELMISTOPROSESSIT

Monien ohjelmistoyritysten organisaatorakenne on prosessimuotoinen, jolloin ohjelmistoprosessi on yrityksen tärkein liiketoimintaprosessi. Tässä luvussa ohjelmistoprosesseihin ja niiden sisältöön tutustutaan tarkemmin. Ensin kerrotaan ohjelmistotuotannosta yleensä, jotta ohjelmistoprosessi asettuisi oikeaan ympäristöönsä. Sen jälkeen tutustutaan ohjelmistoprosessin yleisiin piirteisiin. Ohjelmistotuotannon prosessien sisältöön tutustutaan ISO:n standardiin perustuvan SPICE:n avulla. SPICEä käsitellään ensin yleisellä tasolla, sitten prosessiryhmittäin, jotta jokaisen prosessiryhmän tehtävät ja tarkoitus selviäisi. Luvun loppuksi tehdään yhteenveto käsitellyistä asioista.

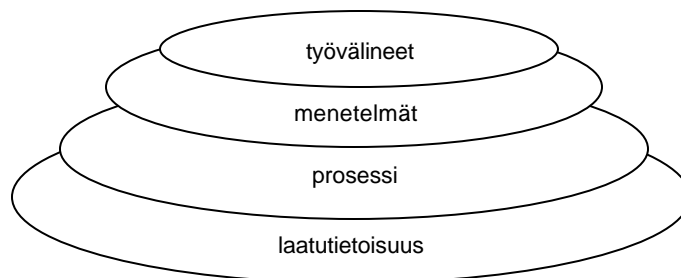
3.1. Ohjelmistotuotanto ja ohjelmistoprosessit

Ohjelmistotuotanto ei ole pelkästään ohjelmistoprosessi. Saukkonen ja Oivo (1998) esittelevät ohjelmistotuotantoa kuvion 7 avulla. Kuviosta ilmenee, että ohjelmistotuotanto koostuu ohjelmistoteknologioista, suunnittelijoista, prosesseista, säännöistä ja vallitsevasta organisaatiokulttuurista, työntekijöiden ryhmistä ja heidän välisestä työnjaostaan. Kaikki ohjelmistotuotannon tekijät vaikuttavat toisiinsa, joten osatekijät ja niiden keskinäinen riippuvuus luo varsinaisen toimintaympäristön ohjelmistoprosessille. Riippuvuuksilla on vaikutusta myös kehittämispyrkimyksissä: kun ohjelmistoprosessia kehitetään, kehittämispyrkimykset ulottuvat kaikkiin muihin ohjelmistotuotannon alueisiin.



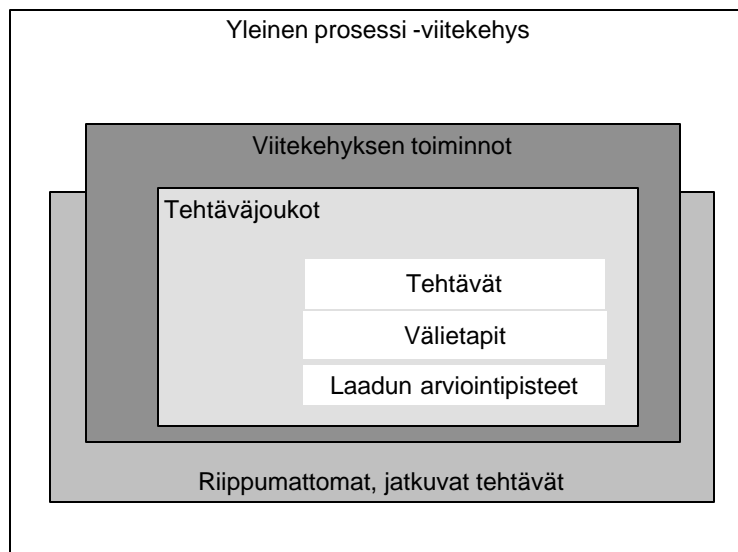
KUVIO 7. Ohjelmistotuotannon osatekijät. (Saukkonen & Oivo 1998, 14)

Saukkosen ja Oivon näkemys ohjelmistotuotannosta on ihmisten toimintaa painottava, kun taas Pressman (1997) painottaa välineellisempää lähestymistapaa. Pressmanin mukaan ohjelmistotuotanto rakentuu kerroksittain alhaalta ylöspäin kuvion 8 esittämällä tavalla. Alimpana, kaiken perustana on organisaation pyrkimys laatuun ja laadukkaaseen tulokseen. Muut ohjelmistotuotannon osa-alueet rakentuvat sen päälle siten, että prosessi-kerros sitoo kaikki muut osa-alueet, eli menetelmät ja välineet, toisiinsa. Prosessi tarjoaa työskentelylle kehyksen ja menetelmät tarjoavat teknisen tietämyksen ohjelmistojen kehittämiseksi. Työvälineet ovat joko täysin automaattisia ohjelmia tai puoliautomaattisia apuvälineitä, esimerkiksi CASE (engl. computer-aided software engineering) –välineitä. (Pressman 1997) Prosessi-kerros on tässä näkökulmassa kokonaisuuden kannalta olennainen, sillä juuri se tarjoaa kullekin ohjelmistotoiminnalle ominaisen kehyksen.



KUVIO 8. Ohjelmistotuotannon kerrokset (Pressman 1997, 23)

Yleistä prosessin viitekehystä tarvitaan yhtenäisyyden luomiseksi erilaisten ohjelmistoprosessien välille (Humphrey 1988). Viitekehys voidaan edelleen pilkkoa osiin ja määritellä kuvion 9 mukaan. Uloin, valkoinen laatikko sisältää yleiset prosessimuotoisen työskentelytavan mallit ja käytännöt ja se pätee minkä tahansa liiketoiminta-alueen prosessiin. Prosessimallit ja käytännöt määritellään tarkemmin tummanharmaassa laatikossa olevien ohjelmistotuotannon viitekehysten toimintojen avulla, jotka soveltuvat kaikenkokoisiin ohjelmistotuotantoprojekteihin, mutta ovat spesifejä juuri ohjelmistotuotannon alalle. Vaaleanharmaan laatikon tehtävajoukkojen avulla viitekehystoiminnot mukautuvat edelleen kunkin yksittäisen ohjelmistoprojektin vaatimusten mukaisiksi ja tekevät kustakin projektista erilaisen. Alimmaisena, harmaalla pohjalla olevat riippumattomat tehtävät voivat olla esimerkiksi ohjelmiston laadun arviointia, konfiguraation hallintaa ja mittamista. Nämä tehtävät eivät ole riippuvaisia mistään viitekehysten muista tehtävistä ja ne jatkuvat läpi koko prosessin kulun. (Pressman 1997, 26) Viitekehysten sisältöä muokkaamalla pystytään säilyttämään ohjelmistoprosessin vertailukelpoisuus muihin ohjelmistoprosesseihin, mutta silti kustakin ohjelmistoprosessista saadaan muunneltua tarkoitukseensa sopiva.



KUVIO 9. Ohjelmistoprosessin viitekehys (Pressman 1997, 26)

3.2. Ohjelmistoprosessin yleisiä piirteitä

Ohjelmistoprosessi on yrityskohtainen ja yhdessä yrityksessä siitä voi olla useita muunnelmia. Kaikkien ohjelmistoprosessien yhteiset piirteet mahdollistavat prosessien hallinnan ja luokittelun. (Humphrey 1989)

Prosessien hallinnan mahdollistamiseksi niiden täytyy sisältää joitakin määreitä, sääntöjä ja käytäntöjä vakauden saavuttamiseksi, esimerkiksi vaatimusten jäädyttämiseksi (Humphrey 1989). Ohjelmistoprosessille tällaisia määreitä on listannut Sommerville (1995), jonka mukaan määreiden ilmenemistapa tekee kustakin ohjelmistoprosessista omanlaisensa. Sommervillen nimeämiä määreitä ovat prosessin määrittelyn *ymmärrettävyys*, etenemisen *näkyvyys* ja välitulosten *selvyys*, prosessin *tuettavuus* käytössä olevilla välineillä, prosessin *hyväksyttävyys* ja *käytettävyys*, mahdollisten *virheiden* aikainen *havaittavuus* ja *välttäminen*, prosessin *kestävyys* ja *ylläpidettävyys* suhteessa ympäröivään organisaatioon ja *nopeus* eli ohjelmiston valmistumisaika. Nämä määreet eivät suinkaan ole ainoa näkökulma ohjelmistoprosessiin (vrt. Pressman 1997; Haikala & Marijärvi 1998), mutta antavat hyvän kuvan siitä, mistä eri ohjelmistoprosessien erot laadussa ja monimutkaisuudessa johtuvat.

Ohjelmistoprosessin ilmenemiseen eli ohjelmiston tuottamistapaan vaikuttaa tuotettavan ohjelmiston monimutkaisuus. Yleensä yksittäisiä sovellusohjelmia on helpompi tehdä kuin kokonaisia järjestelmiä, sillä sovellusohjelmissa on vähemmän muuttumiselle alttiita osia ja toimintoja. (Humphrey 1989) Kokonaisuudessaan ohjelmistoprosessi on joka tapauksessa monimutkainen ja siinä on monia eri tapoja suorittaa samoja tehtäviä. Ylläolevat ohjelmistoprosessin määreet pätevät silti kaikkiin ohjelmistoprosesseihin, vaikka prosessin sisältämät tehtävät saattavatkin vaihdella (Sommerville 1995).

Humphreyn mukaan hyvin määritetyn ohjelmistoprosessin tarkoituksena on auttaa ohjelmistonkehittäjiä tekemään oikeita valintoja. Esimerkiksi vaikeat ongelmat voidaan nopeammin paikantaa ja eristää ennen kuin ne aiheuttavat liikaa kustannuksia. Samalla prosessimäärittely auttaa ohjelmiston kehittäjiä ymmärtämään, mitä heidän itse asiassa

tulisi tehdä, mitä heillä on lupa odottaa toisilta työntekijöiltä ja mitä heiltä odotetaan vastapalveluksena. Hyvin määritetyn prosessin avulla ohjelmiston kehittämisestä tulee vakaampaa, kun ongelmien ennustettavuus ja hallinta paranee. Samalla ohjelmistoprosessin määrittäminen auttaa ymmärtämään ja hallitsemaan sen etenemistä. (Humphrey 1989)

3.3. SPICE

Ohjelmistoprosessin nykytilan ymmärtäminen ja mallintaminen on tärkeää, jotta prosessin kehittämisellä olisi perusta johon kehittämistuloksia verrata. ISO15504 on prosessin laadun mittaamiseen kehitetty standardi, jota tässä tutkielmassa käytetään ohjelmistotuotannon prosessien jaottelun apuvälineenä ja myöhemmin tutkimustulosten analysointivälineenä. Standardista käytetään tässä sen kehittämisprojektin nimeä SPICE (Software Process Improvement and Capability dEtermination). Standardi on käytössä maailmanlaajuisesti ja se tarjoaa kaikille ohjelmistoprosesseille yhteisen perustan (Nevalainen 1999). Tämän kohdan tiedot perustuvat standardista vuonna 1998 julkistettuun versioon (ISO15504-5), jollei muuta lähdettä ole mainittu.

SPICE sisältää ne ohjelmistotuotannon prosessit, jotka ovat välttämättömiä hyvälle ohjelmiston kehitystyölle. Prosessit on jaettu viiteen eri prosessiryhmään, jotka on esitelty ryhmittäin kuviossa 10. Ensimmäisiin elämänsykliprosesseihin kuuluu kaksi prosessiryhmää, asiakas-toimittaja –ryhmä ja kehittäminen prosessiryhmä. Kehittämisen prosessiryhmässä (engl. Engineering (ENG)) on kaksi eri prosessia, kehittäminen ja järjestelmän ja ohjelmiston ylläpito. Asiakas-toimittaja prosessiryhmässä (engl. Customer-supplier (CUS)) prosessit vaikuttavat suoraan asiakkaaseen tukemalla ohjelmiston mukauttamista asiakkaalle sopivaksi ja tarjoamalla apua ja tukea ohjelmiston käytössä. Tukiprosesseja (engl. Support (SUP)) voivat muut prosessit käyttää apunaan elinkaaren eri vaiheissa, ne kuuluvat elämänsykliprosesseja tukeviin prosesseihin. Hallinnan (engl. Management (MAN)) ja organisaation prosessiryhmät kuuluvat organisaation elämänsykliprosesseihin. Hallinnan prosessit sisältävät yleistason työtapoja, joita hyödynnetään sekä projektien että prosessien johtamisessa ja

hallinnassa. Organisaation prosessiryhmän (engl. Organization (ORG)) prosessit puolestaan toteuttavat liiketoiminnallisia tavoitteita ja kehittävät resursseja, tuotteita ja prosesseja yrityksen projektien käyttöön. Prosessiryhmien lisäksi SPICEen kuuluu olennaisena osana prosessin kyvykkyystasot (engl. Process capability levels), mutta niihin ei tämän tutkielman puitteissa puututa.



KUVIO 10. Ohjelmistotuotannon prosessit SPICEen mukaan

3.3.1. Kehittämisen prosessiryhmä

SPICEssa ohjelmiston kehittämisen prosessiryhmän tavoitteena on muuttaa asetetut vaatimukset toimivaksi ohjelmistotuotteeksi tai ohjelmistopohjaiseksi järjestelmäksi, joka vastaa asiakkaan tarpeisiin. Prosessien tuloksena kehitetään valmis ohjelmistotuote ja tuotetaan väliraportit ja muita todisteita vaatimuksiin vastaavuudelle, kuten testausmateriaalia. Sen jälkeen lopputuote asennetaan asiakkaalle ja asiakas hyväksyy sen.

Prosessiryhmän sisältämät aliprosessit kehittävät, määrittävät tai ylläpitävät asiakasdokumentaatiota ja ohjelmistotuotetta ja sen yhteyksiä järjestelmään. Kehittämisen aliprosesseja ovat

1. järjestelmän vaatimusmäärittely ja suunnittelu, jossa järjestelmälle tuotetaan vaatimukset ja arkkitehtuuri
2. ohjelmiston vaatimusmäärittely, jossa tuotetaan vaatimukset ohjelmiston komponenteille
3. ohjelmiston suunnittelu, jonka tarkoituksena on suunnitelman tekeminen, joka toteuttaa ohjelmiston vaatimukset siten, että suunnitelmaa voidaan testata vaatimuksia vasten
4. ohjelmiston toteutus, jossa tuotetaan ohjelmistoyksiköitä ja varmistetaan että ne vastaavat tehtyjä suunnitelmia
5. ohjelmiston integrointitestaus, jossa valmiit ohjelmistoyksiköt liitetään yhteen ja niiden vastaavuus vaatimusmäärittelyn kanssa varmistetaan
6. ohjelmiston kokonaistestaus, jossa ensin kehitetään ne kriteerit, jotka määrittävät toimivan ohjelmistokokonaisuuden, minkä jälkeen integroitu ohjelmisto testataan näiden kriteerien mukaisesti

7. järjestelmän kokoaminen ja testaus, jossa integroidut ohjelmistoyksiköt liitetään yhteen muiden yksikköjen kanssa muodostaen kokonaisen, vaatimusmäärittelyä vastaavan järjestelmän.

Kehittämisen prosessiryhmään kuuluu omana prosessinaan järjestelmän ja ohjelmiston ylläpito. Prosessin tarkoituksena on päivittää ja kehittää asiakkaalle asennettuja ohjelmistoja ja järjestelmää ja huolehtia dokumentointi ajan tasalle. Prosessiin kuuluu myös mahdollinen asiakkaan pyytämä järjestelmän tai ohjelmiston käytöstäpoisto. Kehittämisen prosessiryhmän lisäksi ohjelmistotuotannossa tärkeitä ovat muut SPICE – prosessiryhmät: asiakas-toimittaja, tuki, organisaatio ja hallinta.

3.3.2. Asiakas-toimittaja –prosessiryhmä

Asiakas-toimittaja –ryhmän prosessit vaikuttavat suoraan asiakkaaseen. Ne tukevat ohjelmiston kehittämistä ja siirtämistä asiakkaalle ja välittävät ohjelmistonkehitykseen ohjelmiston käyttötavan ja asiakkaan toivomat toiminnot. Prosessiryhmään kuuluu neljä prosessia aliprosesseineen: hankinta, ohjelmiston toimitus, asiakasvaatimusten selvittäminen ja käyttö.

Hankintaprosessin tarkoituksena on hankkia tai tuottaa palvelu tai tuote, joka tyydyttää asiakkaan tarpeen. Hankintaprosessin aliprosessit huolehtivat hankinnan valmistelusta, toimittajan valinnasta, toimittajan valvonnasta ja asiakkaan hyväksymisestä. Ohjelmiston toimituksesta huolehtii oma prosessinsa, johon kuuluu muun muassa asiakkaan pyyntöön vastaaminen, sopimuksen neuvottelemineen, sopimuksen mukaisen tuotteen kehittäminen, onnistuneen toimituksen tunnistaminen ja tuotteen toimitus ja asennus.

3.3.2.1 Asiakasvaatimusten selvittäminen

Asiakasvaatimusten selvittämisprosessia tarvitaan keräämään, prosessoimaan ja jäljittämään asiakkaan tarpeita ja vaatimuksia koko ohjelmistotuotteen elinkaaren ajan.

Prosessissa muodostetaan vaatimukset tarvittaville ohjelmistotuotteille kommunikoimalla jatkuvasti asiakkaiden kanssa ja määrittämällä sovitut asiakasvaatimukset. Lisäksi muodostetaan mekanismeja, joiden avulla asiakasvaatimukset voidaan liittää aikaisempiin vaatimuksiin ja joilla asiakkaan tarpeita voidaan jatkuvasti valvoa. Muuttuvan teknologian ja muuttuvien asiakastarpeiden myötä syntyvät kehitystarpeet on tärkeää tunnistaa ja hallita. Aliprosesseina asiakasvaatimusten selvittämisessä ovat 1) asiakasvaatimusten hankinta ja selvittäminen, 2) vaatimusten hyväksyminen, 3) vaatimusten rajaaminen projektia ja asiakastarpeiden valvomista varten, 4) asiakkaan vaatimusten muuttumisen hallinta, 5) asiakkaiden odotusten ymmärtäminen ja 6) kysymysmekanismien luominen asiakkaalle, jotta asiakas voi seurata tilannetta ja vaatimusten käyttöä.

3.3.2.2 Käyttö

Käytön prosessin tarkoituksena on käyttää ohjelmistoa sen omassa ympäristössään ja tukea asiakkaita ohjelmiston käytössä. Prosessissa arvioidaan oikeassa ympäristössä ohjelmiston oikeat käyttötavat, käytetään ohjelmistoa sen oikeassa ympäristössä ja opastetaan ja autetaan asiakkaita ohjelmistotuotteen käytössä. Käytön aliprosesseja ovat käyttöpalvelu ja asiakastuki.

Käyttöpalveluprosessin tavoitteena on ohjelmiston käytön riskien tunnistaminen ja niiden lieventäminen, käytön testauksen toteutus ja tyytyväisyyden arviointi. Tämän lisäksi prosessin aikana ohjelmistoa käytetään suunnitellussa ympäristössä, tunnistetaan, priorisoidaan ja ratkaistaan käytön ongelmat, reagoidaan käyttäjien pyyntöihin, dokumentoidaan väliaikaiset ratkaisut ja valvotaan järjestelmän kapasiteettia ja palvelua.

Asiakastuen prosessin tarkoituksena on ylläpitää palvelu hyväksyttävällä tasolla ja tukea asiakasta ohjelmistotuotteen tehokkaassa käytössä. Prosessissa käyttäjät koulutetaan, käyttöohjeet dokumentoidaan ja asiakkaalle muodostetaan tuotetuki josta voi kysyä apua. Ohjelmiston toimintaa valvotaan ja asiakkaan tyytyväisyystaso määritetään ja sitä

verrataan kilpailijoiden tarjoamaan tyytyväisyyteen. Asiakastyytyväisyys viestitään koko organisaatioon.

3.3.3. Tuen prosessiryhmä

Tuen prosessiryhmä kuuluu elämänkaariprosesseja tukeviin prosesseihin. Muut prosessit voivat hyödyntää tukiprosesseja ohjelmiston elinkaaren eri vaiheissa. Tuen prosessiryhmän prosesseja ovat

1. dokumentointi, jonka tarkoituksena on kehittää ja ylläpitää dokumentteja, joihin kirjataan tietoa prosesseista tai niiden toimista
2. kokoonpanon hallinta, jolla hallitaan työtulosten eheyttä kaikissa prosessin vaiheissa
3. laadunvarmistus, jonka tavoitteena on varmistaa, että työtulokset ja prosessi noudattavat määritettyjä vaatimuksia ja ovat suunnitelmien mukaisia
4. todennus, jossa varmistetaan että työn tulokset, palvelut tai prosessit seuraavat niille asetettuja vaatimuksia
5. kelpuutus, jossa varmistetaan että ohjelmisto täyttää sen käytölle asetetut vaatimukset
6. yhteiskatselmus, jolla ylläpidetään yhteisymmärrystä asiakkaan kanssa sopimuksen tavoitteiden saavuttamisesta
7. auditointi, jossa määritetään valittujen tuotteiden ja prosessien yhteensopivuus vaatimusten, suunnitelmien ja sopimuksen kanssa
8. ongelmanratkaisu, jonka tavoitteena on ratkaista ja analysoida jokainen löydetty ongelma ja tunnistaa löydetyt trendit.

3.3.4. Hallinnan prosessiryhmä

Hallinnan prosesseja voi käyttää kuka tahansa projektin tai prosessin johtaja, missä tahansa prosessin elinkaaren vaiheessa. Hallinnan prosessiryhmään kuuluu neljä prosessia aliprosesseineen: johtamiskäytännöt, riskienhallinta, projektinhallinta ja laadunhallinta ja -varmistus.

Johtamiskäytännöt –prosessin tarkoituksena on organisoida, valvoa ja ohjata prosessien tai toimintojen aloittamista ja suorituksia siten, että ne saavuttaisivat tavoitteensa ja organisaation liiketoiminnalliset tavoitteet mahdollisimman tehokkaasti. Prosessiin kuuluu tehtävien ja toimien tunnistaminen, prosessin toteutumisen arviointi, resurssien suunnittelu ja varaaminen, vastuiden jako, tuotosten valvonta ja tarkistus, poikkeamien huomiointi ja työtuloksen esittely.

Riskienhallinnan tavoitteena on tunnistaa ja lieventää projektin riskejä. Prosessissa määritellään riskienhallinnan laajuus ja riskit tunnistetaan, analysoidaan ja priorisoidaan. Riskienhallinnan strategiat määritetään ja ne toteutetaan organisaation laajuisesti tai projektikohtaisesti. Lisäksi riskeille määritetään mittarit, joilla mitataan riskienhallinnan strategioiden tulokset. Tulosten perusteella ryhdytään mahdollisiin korjaaviin toimiin.

3.3.4.1 Projektinhallinta

Ohjelmistoprosessi kulkee usein yhteenkietoutuneena projektinhallintaprosessin kanssa (Toffolon & Dakhli 2000), sillä useimmat ohjelmistot valmistetaan projektissa. SPICEn mukaan projektinhallintaprosessin tarkoituksena on tunnistaa, muodostaa, koordinoita ja valvoa toimenpiteitä, tehtäviä ja resursseja, jotka ovat välttämättömiä projektille, jotta se voi tuottaa tuotteen tai palvelun, joka vastaa vaatimuksia. Projektinhallintaprosessin aliprosesseja ovat

1. työn laajuuden määrittely, jossa työmäärän lisäksi määritetään projektille riittävät resurssit
2. kehittämisstrategian määrittely, jossa arvioidaan projektin toteuttamisen vaihtoehtoja
3. ohjelmiston elinkaaren mallin valinta
4. tehtävien mitoittaminen ja resurssien arviointi
5. projektin tehtävien ja toimitusten ja niiden järjestyksen yhdistäminen resursseihin
6. infrastruktuurin ja inhimillisten resurssien tunnistaminen
7. projektin aikataulun tekeminen
8. vastuiden jako yksilöille ja ryhmille
9. elementtien välillä olevien liitännöiden tunnistaminen, jossa tunnistetaan ja valvotaan elementtien rajapintoja sekä projektin sisällä että organisaation laajuisesti
10. projektisuunnitelmien muodostaminen ja toteuttaminen
11. prosessin etenemisen jäljittäminen esimerkiksi teknisillä tai hallinnollisilla raporteilla
12. häiriöiden korjaaminen silloin, kun projektin tavoitteita ei ole saavutettu. Tähän kuuluu myös projektisuunnitelman säännöllinen päivittäminen.

3.3.4.2 Laadunhallinta ja -varmistus

Laadunhallinta ja -varmistus sisältää prosesseja, joiden avulla valvotaan projektin tuotteiden ja palveluiden laatua ja varmistetaan asiakkaan tyytyväisyys. Prosessissa muodostetaan laadulle tavoitteet asiakkaan vaatimusten ja yleisten laatuvaatimusten pohjalta. Sen jälkeen määritetään kokonaisstrategia, jolla määritetyt tavoitteet pyritään saavuttamaan koko organisaation laajuisesti. Projektin toimien tuloksien mittaamista ja laadun hyväksymiskriteereiden määrittystä varten määritetään erilliset mittarit. Kullekin laatutavoitteelle tunnistetaan ja tuotetaan laatuksentrollit ja varmistustoimet. Laatutavoitteiden saavuttaminen mitataan projektin elinkaressa olevissa tarkistuspisteissä. Korjaaviin toimiin ryhdytään, jos laatutavoitteita ei ole saavutettu.

3.3.5. Organisaation prosessiryhmä

Organisaation prosessiryhmä muodostuu prosesseista, joissa muodostetaan organisaation liiketoiminnan tavoitteet ja kehitetään prosessit, tuotteet ja resurssit niiden käyttöön. Organisaation luoma perusrakenne on tärkeä, sillä ohjelmistoprosessit toteutetaan sen avulla. Organisaation prosessiryhmään kuuluu organisaation johtaminen, parantaminen, henkilöstöhallinto ja koulutus, puitteet, välineet ja infrastruktuuri, mittaustoiminta ja uudelleenkäyttö.

Organisaation johtamisprosessin tavoitteena on kehittää ja viestiä yhteinen näkemys, kulttuuri ja ymmärrys liiketoiminnan tavoitteista. Parantamisprosessissa muodostetaan, arvioidaan, mitataan ja parannetaan ohjelmistoprosessia. Parantamiseen kuuluu organisatoristen prosessien määrittely, kuvaaminen, dokumentointi ja mittaus. Parantamisen aliprosesseja ovat prosessin perustaminen, arviointi ja prosessin parannus.

Henkilöstöhallinto- ja -koulutusprosessin tavoitteena on hankkia organisaatioon henkilöitä, joilla on kyky työskennellä ryhmässä ja riittävästi tietoa ja taitoa suorittaa tehokkaasti heille annetut tehtävät. Prosessiin kuuluu työvoimatarpeen tunnistaminen;

koulutuksen hankkiminen, kehittäminen ja toteuttaminen; henkilöstön rekrytointi; työntekijöiden tulosten arviointi suhteessa tavoitteisiin ja palautteenanto tuotoksista.

3.4. Yhteenveto

Tässä luvussa todettiin, että ohjelmistotuotantoon kuuluu koko se ympäristö, jossa ohjelmistoja tuotetaan. Saukkosen ja Oivon mukaan (1998) ohjelmistotuotannon osatekijät ovat prosessin lisäksi ohjelmistoteknologia, suunnittelija, säännöt ja kulttuuri, ryhmä ja työnjako. Yhden osatekijän kehittäminen heijastuu myös muihin tekijöihin. Pressmanin (1997) näkemyksen mukaan ohjelmistotuotanto rakentuu kerroksittain, jolloin laatutietoisuus muodostaa kerrosten perustan. Prosessi toimii kerrosten liimana ja liittää yhteen käytettävät menetelmät ja työvälineet.

Ohjelmistoprosessi on monimutkainen prosessi, jonka ymmärtämistä voidaan edistää sijoittamalla se omaan viitekehykseensä, jossa ohjelmistoprosessin tehtävät ovat vain yksi osa kokonaisuutta. Eri ohjelmistoprosessit voivat olla erilaisia jopa saman yrityksen sisällä, sillä prosessia tulee muokata tuotettavan tuotteen mukaiseksi. SPICE jakaa ohjelmistotuotantoon liittyvät prosessit viiteen prosessiryhmään, joissa kussakin on omat aliprosessinsa. Prosessiryhmät ovat asiakas-toimittaja, kehittäminen, tuki, hallinta ja organisaatio.

Ohjelmistotuotantoon kuuluvien prosessien kokonaismäärästä on monia käsityksiä, SPICEssa määriteltyjä prosesseja on 36. Saukkosen ja Oivon mukaan (1998) pienessä ohjelmistotuotantoyrityksessä on vain noin 20 prosessia, jotka tulisi määritellä ja joiden kehittämiseen tulisi kiinnittää huomiota, jotta yhteistoiminta ja vuorovaikutus sujuisivat. Davenportin mukaan (1993) suurimmissakin yrityksissä keskeisiä prosesseja on alle 20. Lukujen eroa selittää organisaatioiden näkemys omien prosessiensa tärkeydestä. Liiketoiminnalle epäolennaisia prosesseja ei kannata määritellä, jos siitä aiheutuu enemmän kuluja kuin hyötyä (ks. Saukkonen & Oivo, 16).

Kokonaisuutena SPICE -standardi tarjoaa kattavan paketin ohjelmistotuotannon prosesseja. Standardin jaottelun avulla ohjelmistoprosessi voidaan jakaa aliprosesseihin ja siitä saadaan helpommin käsitettävä, osiin jaettu kokonaisuus. Riippuu pitkälti organisaation rakenteesta ja sen mukanaan tuomista toimintatavoista, miten ohjelmistoprosessi käytännössä toteutetaan, sillä prosessimuotoinen työtapa tuo oman lisänsä ohjelmistotuotantoon ja ohjelmistoprosessin suorittamiseen.

4. PROSESSIMUOTOINEN TYÖTAPA JA SEN MALLINTAMINEN

Prosessimuotoinen työtapa on noussut suuren yleisön tietoisuuteen 1990-luvulla. Sitä ennen prosesseilla ei ole hehkutettu, vaikka niitä on ollut olemassa. Prosessimuotoisen työtavan avulla haetaan kustannussäästöjä järkeistämällä tehtäviä suuremmiksi kokonaisuuksiksi. Työtavalle on ominaista tuotteiden ja palvelujen räätälöinti asiakkaalle eri toimintoketjujen eli prosessien avulla. Tässä luvussa prosessimuotoiseen työtapaan, sen taustaan ja tunnuspiirteisiin tutustutaan. Prosessien kuvaamistapoja käsitellään hiukan ja tutustutaan lyhyesti muutamiin prosessi- eli vaihejakomalleihin.

4.1. Prosessimuotoinen työtapa

Prosessimuotoisen työtavan perusta tulee liiketoimintojen uudelleenjärjestelyn (BPR) alalta, jonka tarkoituksena on nimensä mukaisesti liiketoimintaprosessien perusteellinen ja radikaali uudelleenjärjestely. Karsimalla turhia toimintoja ja etsimällä parempia työtapoja pyritään parantamaan yrityksen kannattavuutta. (Hammer & Stanton 1994)

Työtapaan voidaan suhtautua näkökulmana tai lähestymistapana, jossa yrityksen toiminnot on jaettu liiketoimintaprosesseiksi, jotka koostuvat tehtävien ketjuista. Yhdessä yrityksessä on monia rinnakkaisia ja peräkkäisiä prosesseja, jotka hoitavat oman alueensa tehtävät. Esimerkiksi markkinoinnissa ja tuotekehityksessä on kummassakin omat prosessinsa. Kun yhdessä yrityksessä prosesseja on useampia, prosessien keskinäisen vuorovaikutuksen toimivuus tai toimimattomuus korostuu.

Prosessimuotoisella työtavalla on pyritty hallitsemaan ohjelmistotuotantoprojektien kriittisiä menestystekijöitä ja ennustamaan niiden todennäköisyyksiä. Ydinajatuksena työtavan omaksumisessa on ollut työn järkeistäminen ja sitä kautta tuotantoaikojen lyhentäminen ja sidotun pääoman vähentäminen. Japanin yrityskulttuurissa prosessit ovat olleet vaikuttava tekijä jo vuosikymmenien ajan. (Davenport 1993) Ei ole ihme,

että useilla japanilaisilla yrityksillä on niin hyvä markkina-asema verrattuna moniin eurooppalaisiin ja amerikkalaisiin kilpailijoihinsa.

4.1.1. Prosessimuotoisuuden taustaa

Hammer ja Champy kirjoittavat kirjassaan prosessimuotoisen työtavan alkuaajoista Yhdysvalloissa. Yritykset halusivat olla tehokkaita, joustavia ja pystyviä vastaamaan markkinoiden haasteisiin ja asiakkaiden muuttuviin vaatimuksiin. Uudet työvälineet ja työkäytännöt tarjosivat uudenlaisia mahdollisuuksia liiketoiminnalle. (Hammer & Champy 1993) Tehtävien tehokkaampi ja joustavampi tekeminen vaati entistä enemmän yhteistyötä, joka ylitti perinteiset yksittäisten toimintojen rajat. Enää ei voitu ajatella, että ”tämä on minun tehtäväni, minä teen vain tämän”, vaan liiketoimintaa oli alettava ajatella uudella tavalla. Asiakkaiden tarpeisiin piti pystyä vastaamaan, mikä ei ollut yksinkertaista jatkuvassa muutoksen aallossa ja markkinakilpailussa. Ratkaisuksi nähtiin liiketoiminnan uudelleen järjestäminen prosessimuotoiseksi (Hammer & Champy 1993). Toiminnan tehostamiseen tarvittiin ryhmätyötä, joka taas edellytti toiminnan koordinoitua. Prosessimuotoinen työtapa yhdisti perinteiset yhden henkilön tehtävät tehokkaaksi tehtäväsarjaksi, jonka ryhmä pystyi tekemään yksin tai yhdessä toisten ryhmien kanssa.

Tällä tavoin yksittäisistä tehtävistä tulee osa isompaa prosessia, joka muodostaa toimivan kokonaisuuden tavarantoimittajalta asiakkaalle saakka. Kaikilla prosessin toimilla ja tehtävillä on viime kädessä yhteinen lopputulos, johon jokainen tehtävä osaltaan vaikuttaa. Uudenlaista, prosessimaista näkökulmaa tarvittiin tehtävien päällekkäisyyksien karsimiseen ja niiden yhteensopivuudesta huolehtimiseen. Perustana prosessimuotoisuudessa on kommunikointi- ja päätöksentekohierarkian vaakasuuntaisuus, jota seuraavassa selitetään informaation prosessoinnin näkökulman avulla.

4.1.2. Informaation prosessoinnin näkökulma

Galbraithin esittelemän informaation prosessoinnin näkökulman mukaan organisaatioon kohdistuva ympäristön epävarmuus ja siitä johtuva kommunikoinnin tarve pakottaa organisaation muuttamaan työtapojaan (Galbraith 1973, 9). Tuloksena saadaan vaakasuuntaista kommunikaatiota tukeva prosessimuotoinen työtapo, jossa päätöksentekovastuu on viety mahdollisimman alhaiselle hierarkian tasolle.

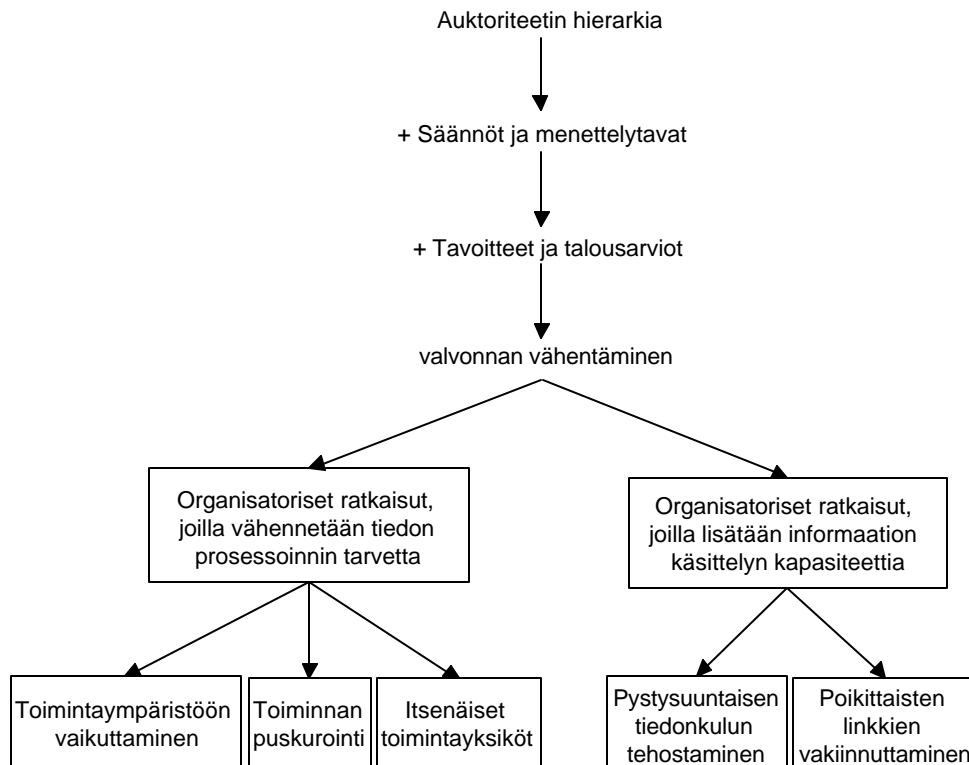
Isojen organisaatioiden toiminnan koordinointiin tarvitaan niin paljon kommunikointia, ettei se ole enää mahdollista kasvotusten. Tällöin auktoriteetin hierarkia muotoilee informaatiokanavat: osastoa koskevaa päätöstä kysytään osaston esimieheltä, koko yritystä koskevaa asiaa kysytään toimitusjohtajalta. Pystysuuntaiset informaatiokanavat kuitenkin tukkeutuvat helposti, jos päätöksentekotilanteet erilaistuvat jatkuvasti. Rutiiniluonteisissa ja toistuvissa päätöksentekotilanteissa voidaan soveltaa sääntöjä ja yhteisesti sovittuja toimintamenetelmiä. Nopeasti kehittyvässä toimintaympäristössä säännöt kuitenkin vanhenevat nopeasti, eikä niistä ole pitkäaikaista hyötyä. (Galbraith 1973)

Kun informaation kokonaismäärä alkaa rasittaa hierarkiaa liikaa, päätöksenteko kannattaa tehdä mahdollisimman alhaisella hierarkian tasolla. Tällöin ongelmana on päätösten erilaistuminen, kun jokainen päättää niin kuin itse parhaaksi katsoo. Ratkaisuna ongelmaan on henkilöstön riittävä koulutus, jotta he osaavat ja haluavat seurata organisaation prosesseja oikein. Tämän askeleen ottaminen ja vastuun jakaminen hierarkian alimmille tasoille saattaa vaatia hyvinkin pitkän ajan. (ks. Galbraith 1973, 13; Hammer & Champy 1993)

Vastuun jakaminen tapahtuu ottamalla käyttöön työtapoja, joilla tiedon prosessoinnin tarvetta voidaan vähentää, ja ratkaisuja, joiden avulla informaation käsittelyä voidaan tehostaa. (Galbraith 1973) Kun toimintaympäristö muuttuu ja uusiutuu jatkuvasti, sitä yritetään kontrolloida ylettämällä omat prosessit myös tavarantoimittajan ja asiakkaan organisaatioihin (ks. kuvio 4 s. 9; Laamanen & Tinnilä, 1998, 7). Toimintaa puskuroidaan pitämällä osa resursseista varalla. Organisaation sisälle luodaan itsenäisiä

toimintayksiköitä, ohjelmistotuotannossa yksiköt ovat projekteja. Informaation käsittelyä pyritään tehostamaan sekä pysty- että erityisesti vaakasuuntaisten informaatiokanavien avulla. Yhdessä organisaatiossa ei välttämättä toteuteta kaikkia näitä keinoja, sillä jokainen niistä aiheuttaa omat kulunsa.

Kuvio 11 selventää informaation prosessoinnin näkökulmaa. Siinä kommunikoinnin tarpeeseen vastataan auktoriteetin hierarkialla, sääntöjen ja menettelytapojen luomisella. Lisäksi toiminnalle asetettujen tavoitteiden ja niiden talousarvioiden noudattaminen mahdollistavat kommunikoinnin vähentämisen ja edelleen päätöksenteon siirtämisen hierarkian alemmille tasoille. (Galbraith 1973) Informaation prosessoinnin näkökulma tuo prosessimuotoiseen työtapaan perustan, mutta työtavan suosion taustoja pitää etsiä muualta.



KUVIO 11. Informaation prosessoinnin näkökulma (vrt. Galbraith 1973, 9-20)

4.1.3. Lähtökohdat ja tavoitteet

Yhtenä syynä prosessimuotoisuuden nousuun ohjelmistotuotannossa on alan kova kilpailutilanne, joka pakottaa yritykset etsimään jatkuvasti uusia pärjäämisen keinoja (Davenport 1993). Suomessa ja Yhdysvalloissa tehtyjen tutkimusten mukaan useissa tietotekniikkaprojekteissa on suuria ongelmia erityisesti tietojärjestelmien tuotantoprosessissa (Pylkkö 1999). Lyytisen (1988) tutkimuksessa yksi ohjelmistotuotantoprojektien epäonnistumiseen painavimmin vaikuttavista tekijöistä oli riittävän kommunikoinnin ja selkeän rakenteen puuttuminen ohjelmistoprosessista. Reel (1999) kirjoittaa, että ongelmat vain lisääntyvät ohjelmistojen monimutkaistumisen myötä. Kaikkiin näihin ongelmiin haetaan ratkaisua prosessimuotoisesta työtavasta. Työtavan toivotaan nopeuttavan tuotantoketjuja ja tuottavan laadukkaampia tuotteita ja tällä tavoin luovan yritykselle kilpailuetua markkinoilla (Davenport 1993).

Markkinoilla pärjäämisen lisäksi toisena motivoijana prosessimuotoiseen työskentelytapaan siirtymisessä ovat asiakkaat ja pyrkimys täyttää heidän vaateensa (Davenport 1993). Jokainen asiakas haluaa hoitaa asiansa nopeasti ja kivuttomasti ja saada kunnollista palvelua. Kukaan ei halua odottaa tilauksen käsittelyn etenemistä päiväkausia tai soittaa monelle eri henkilölle kysyäksään tilauksensa perään. Prosessien avulla voidaan toimintaa tehostaa merkittävästi nimenomaan asiakkaille näkyvällä tavalla.

Taloudelliset seikat ovat kolmas syy, jonka takia prosesseihin yleensä siirrytään. Tilanteessa, jossa yrityksen on pakko vähentää kuluja ja kustannuksia saadakseen esimerkiksi velkansa tai investointinsa maksettua, prosessimainen työtapa on tehokas organisaation sisäinen tapa parantaa yrityksen tuottavuutta. (Davenport 1993)

Toisinaan yrityksellä menee ulkoisesti hyvin, mutta sisäinen koordinointi ja toimintojen hallinta eivät toimi. Tällöin prosessit voivat tuoda yritykselle toimintamallin, jossa toimintojen välisiä rajoja venytetään ja jota seuraamalla eri toiminnot pystyvät paremmin tekemään yhteistyötä. Toiminnan järjeistäminen prosessien avulla parantaa

tuottavuutta edelleen, kun asiakkaiden toiveet välittyvät paremmin koko organisaation poikki. (Davenport 1993)

Prosessimuotoisuudesta on sanottu, että se on paljon enemmän kuin pelkkää järjeistämistä tai yksinkertaistamista ja enemmän kuin maalaisjärki. Se kyseenalaistaa perinteisen osaamisen siitä, mikä on helppoa, taloudellista ja järkevää. Toisinaan se johtaa pikemminkin monimutkaisempiin kuin yksinkertaisempiin prosesseihin, mutta ajansäästö voi silti olla jopa kymmenkertainen. (Davenport 1993) Prosessien monimutkaisuus saattaa myös muuttua ajan kuluessa, sillä prosessit kehittyvät ajan mittaan. Kehittymiseen vaikuttaa muun muassa kilpailu markkinoilla, uudet teknologiat ja uudenlaiset asiakasvaatimukset. (Kock & McQueen 1996) Monimutkaisuutta ei kuitenkaan pidä ajatella pelkkänä negatiivisena asiana, sillä monimutkaisellakin prosessilla voidaan toteuttaa isoja asioita tai tuotteita nopeammin, helpommin ja pienillä kuluilla. Prosessimuotoisessa työtavassa on omat vahvuutensa ja ominaispiirteensä, joita käydään läpi seuraavaksi.

4.1.4. Tunnuspiirteitä

Yksi prosessimuotoisen työskentelyn vahvuuksista on uusiutuminen. Uusiutuminen on välttämätöntä, kun ympäristön jatkuva ja nopea muuttuminen ja uusien teknologioiden syntyminen luovat yrityksille muuttumisen paineita. Prosessimuotoinen organisaatorakenne ja työskentelytapa on yksi tapa vastata uusien uhkien asettamiin haasteisiin, sillä prosesseja tai niiden osia voidaan kohtuullisella vaivalla muokata ja kehittää sitä mukaa kun on tarvetta. (Kock & McQueen 1996)

Asiakaslähtöisyys on toinen prosessimuotoisuuden etu. Prosessit lähtevät liikkeelle asiakkaan tarpeesta tai toiveesta ja prosessin aikana tuotetta tai tulosta pyritään jatkuvasti kehittämään ja parantamaan. Prosessin tuloksena saadaan vastaus asiakkaan tarpeeseen ja jokaisella prosessilla on oma osavastuunsa asiakkaalle. Tästä huolimatta asiakkailla ei useinkaan ole tarpeeksi tietoa niistä prosesseista, joilla heidän ostamansa tuotteet tuotetaan (Kock & McQueen 1996).

Uusiutumisen ja asiakaslähtöisyyden lisäksi prosessimaiselle työtavalle on Karvosen (1999) mukaan tyypillistä

- eteneminen hierarkiassa alhaalta ylöspäin
- sopivan korkea laatu, tuottavuus ja tehokkuus
- kokonaisvaltaisuus ja keskittyminen olennaiseen
- käytännönläheisyys ja joustava toimintatapa
- vapaa kommunikointi, avoimuus, kaksisuuntaisuus ja vastuullisuus
- pienet askeleet ja osaamistason kohottaminen
- jatkuva parantaminen ja oppiminen
- sitoutuminen, yhteisöllisyys ja tiimitaidot
- joustavuus ja reagoitiherkkyys.

Myös jatkuva epätasapaino, valintatilanteet ja muutos ovat ominaisia prosesseille. Näihin sopeutuminen edellyttää yrityksiltä ja organisaatioilta jatkuvaa valppautta ja liikkeelläoloa. (Karvonen 1999) Yritysten uusiutumiskykyä edellyttävät myös Hammer ja Champy (1993), joiden mukaan prosessimuotoiseen työtapaan pätee, että

- useita työtehtäviä yhdistetään yhdeksi työksi
- työntekijät tekevät päätöksiä ja heillä on enemmän valtaa
- prosesseista on monia eri versioita
- työ tehdään siellä missä se on järkevintä
- tarkastuspisteitä ja valvontaa on vähän
- sovittelut ja neuvottelut minimoidaan
- prosessilla on yksi ainoa kontaktihenkilö ulos- ja sisäänpäin
- toiminnot ovat puoliksi hajautettuja ja puoliksi keskitettyjä
- työn yksiköt ovat prosessi-tiimejä
- työtehtävät ovat moniulotteisia
- työhön perehtyminen on pikemminkin koulutusta
- painotetaan tuloksia ja arvostetaan tuottavuutta
- kehittymisen mittarina on taidot ja kyky
- johtajat ovat ”valmentajia” ja yritysjohtajat ovat eräänlaisia ohjaajia (leaders)

➤ organisaatorakenne on tasainen.

(mts., s. 51-73)

Vaikka Karvosen ja Hammerin ja Champyn näkökulmat hieman eroavat toisistaan, ne ovat silti yhdenmukaiset painottaessaan joustavuutta, ryhmätyötä ja työntekijöiden valtaa ja vastuuta. Yhteistä näille listauksille on myös se, että kaikilla toimilla pyritään suoraan tai epäsuoraan vaikuttamaan tuotettavan tuotteen tai palvelun laatuun.

4.1.5. Laatu prosessien näkökulmasta

Ohjelmistojen laatu on avainasemassa niillä aloilla, joilla ohjelmistoja ylipäättään tarvitaan; siis lähes kaikessa teollisuudessa (Chroust 2000). Huippuosaajien palkkaaminen, erilaiset toiminnan organisointitavat ja organisaation sisäinen rakenne ovat tapoja hallita ja vähentää koko ajan lisääntyvää epävarmuutta (ks. esim. Frank & Brownell 1989, 82; Riddle 1987; Lehman 1989; Kahen & Lehman 2000). Vaikeampaa on vastata asiakkaiden asettamiin laatuvaatimuksiin.

Ohjelmistotuotteista, niin kuin kaikista muistakin tuotteista ja palveluista, halutaan yhä laadukkaampia yhä lyhyemmässä ajassa. Laadun käsite ja pyrkimys laatuun on hyvin vahvasti läsnä prosessimuotoisessa työtavassa ja laadukkaan prosessin oletetaan tuottavan laadukkaita lopputuloksia (Karvonen 1999). 15 eurooppalaista yritystä käsittäneen tutkimuksen mukaan prosessimuotoiseen työtapaan siirrytään pääosin kolmesta syystä. Ensinnäkin halutaan parantaa organisaation vastaanottavuutta uusille ideoille, muutokselle ja kehitykselle. Toiseksi prosessimuotoisen työtavan ajatellaan olevan lähes pakollista nykyisessä kilpailutilanteessa. Kolmanneksi tärkein motiivi on halu parantaa yrityksen toiminnan laatua. Vastaavasti kolme suurinta hyötyä, joita prosessimuotoisella työtavalla on saavutettu, ovat parantuneet asiakassuhteet, parempi yhteistyö organisaatioiden eri toimintojen kesken ja organisaatiokulttuurin muutos. (Pritchard & Armistead 1999)

Jotta yhteistyö, muutos ja laatuajattelu eivät jäisi pelkäksi puheeksi, kaikilla prosesseilla tulisi olla selkeästi määritelty omistaja eli henkilö, joka vastaa prosessin kulusta ja kehittämisestä ja prosessin tuottaman tuloksen laadukkuudesta. Prosessien kulku tulee dokumentoida mallintamalla, jotta prosessiin osallistujat saavat tietoa kulloisestakin vaiheesta ja sen syötteistä ja tulosteista. Prosessia mitataan sen tuottamien kulujen, laadukkuuden ja ajankäytön mukaan. (Lee & Dale 1998) Laatuajattelu on kaiken perusta, jota tulee soveltaa kaikkiin toimintatapoihin ja tuotettaviin tuotteisiin (ks. esim. kuvio 9 s.17).

Kaikenlaisia tuotteita tai palveluita ei voida valmistaa yhdellä laadukkaalla prosessilla, sillä vaikka prosessi itsessään olisi laadukas, siis standardoitu, valvottu ja hyvin toteutettu, se ei automaattisesti takaa laadukasta lopputulosta. (Sommerville 1995) Käytettävän prosessin täytyy olla juuri sopiva kutakin eri tuotetta varten, joten prosessin on muututtava kun tuotteen tai palvelun vaatimukset muuttuvat.

Erilaisille prosesseille on tarvetta, sillä tuloksien haluttu laatu vaihtelee. Toisaalta tietynasteisella prosessien yhdenmukaistamisella saavutetaan paljon etuja, ennen kaikkea se mahdollistaa prosessien ja niiden laadun mittaamisen. Mitä yhdenmukaisemmat yrityksen prosessit ovat, sitä yhdenmukaisemmat ovat myös työssä tarvittavat ohjelmistot ja muut apuvälineet. Henkilökunnan koulutusta tarvitaan vähemmän ja yhdessä projektissa opitut prosessikäytännöt voidaan helpommin siirtää muihin yrityksen prosesseihin. Yhdenmukaisuuden ja erilaistamisen yhdistäminen vaatii toki aikaa, taitoa ja vaivaa, mutta onnistuessaan se luo yritykseen toimivan prosessikäytännön. (Humphrey 1989) Täysin yhteneviin prosesseihin tuskin missään yrityksessä päästään, sillä se edellyttäisi täsmällisen prosessimallin seuraamista äärimmäisen tarkasti, mikä ei suinkaan ole prosessimuotoisen työtavan tai prosessien mallintamisen tarkoitus.

4.2. Prosessien mallintaminen

Tehtävien ollessa monimutkaisia, monen ihmisen työpanoksen järjestämiseksi yhdeksi kokonaisuudeksi tarvitaan muodollisia käytäntöjä epämuodollisten tapaamisten lisäksi. Yksinkertaisinkin prosessi voi olla monimutkainen ja siksi tarvitaan prosessimalli kuvaamaan prosessin vaiheita, suhteita ja riippuvuuksia. Prosessien mallintamisella on viisi pääasiallista tehtävää: 1) helpottaa ihmisten keskinäistä kommunikointia ja yhteisen ymmärryksen syntyä, 2) tukea prosessin kehittämistä ja parantamista, 3) tukea prosessin hallittavuutta, 4) automatisoida prosessin ohjaus ja 5) automatisoida toteuttamisen tuki. Mallintamisen päätavoitteena on, että prosessit vastaisivat mahdollisimman hyvin työn oikeata kulkua, eivätkä olisi vain ylemmän tason kuvauksia, joilla ei oikeassa elämässä ole mitään käyttöä. (Curtis, Kellner & Over 1992) Kun mallien avulla voidaan ennustaa prosessin tosiasiallista etenemistä, malleista saadaan suurin hyöty. Myös käytännön prosessista saadaan enemmän hyötyä, kun on olemassa formaali malli, johon prosessin suoritusta ja tavoitteiden toteutumista voidaan verrata. (Cook & Wolf 1999) Prosessimallien määrittämiseen on kehitetty monenlaisia kuvaustapoja.

Prosessien kuvaustavat edustavat yhtä tai useampaa lähestymistapaa: toiminnallista, käyttäytymistä painottavaa, organisatorista tai tiedollista lähestymistapaa. Kuvaustavat eroavat toisistaan muun muassa matemaattisen muodollisuuden, prosessin osien koon ja tarkkuuden ja prosessin ohjailevuuden ja kelpoisuuden suhteen. Eri kuvaustapoja voidaan hyödyntää erilaisissa malleissa, joita ovat ohjelmointimallit, toiminnalliset mallit, suunnitelmapohjaiset mallit, petri-verkko -mallit ja kvantitatiiviset mallit. Kukin malli painottaa prosessin eri puolia ja ominaisuuksia ja niillä on omat hyvät ja huonot puolensa. (Curtis ym. 1992) Näin pystytään tuottamaan erilaisia malleja eri tarkoituksia varten.

Erilaisiin ohjelmistotuotannon projekteihin sopivat erilaiset kuvaustavat riippuen siitä, mitä mallin piirteitä projektissa halutaan painottaa. Prosessimallien vaihteluun vaikuttaa sekä organisaatio että projekti, jossa sitä käytetään. Käytettävää prosessimallia valitessa tulee huomioida henkilökunnan taidot ja kokemus ja käytettävissä olevat apuvälineet,

esimerkiksi CASE –välineet (Humphrey 1989). Määritetty ohjelmistoprosessin malli tarjoaa organisaatiolle pysyvän viitekehyksen ohjelmistojen tekemiselle ja työn tekotapojen parantamiselle (Humphrey & Kellner 1989). Prosessimalleilla on osuutensa myös siihen, millaisena prosessin tulos nähdään prosessin elinkaaren eri vaiheissa.

4.2.1. Prosessi- eli vaihejakomallit

Jokaisella ohjelmistotuotteella on *elinkaari*, joka tarkoittaa aikaa ohjelmiston kehittämisen aloittamisesta siihen asti, kunnes ohjelmisto poistetaan käytöstä. Prosessimallit eli *vaihejakomallit* määrittelevät tavan, jolla elinkaari jaetaan sykleihin tai vaiheisiin. (Haikala & Märijärvi 1998) Malleja on monia ja ne eroavat toisistaan eri vaiheiden painotuksessa ja vaiheiden suorittamisjärjestyksessä: toisissa prosessimalleissa mennään kerralla alusta loppuun, toisissa kehitetään ohjelmistoa vähä vähältä kokonaisuutena. Yksinkertaistettuna, kaikki prosessimallit heijastavat yleistä ongelmanratkaisumallia, jonka vaiheet ovat vallitseva tilanne (status quo), ongelman määrittely, tekninen kehitys ja ratkaisun integrointi. (Pressman 1997, 31-32) Prosessimalleissa nämä vaiheet on nimetty vaatimusmäärittelyn tekemiseksi, ohjelmiston tuottamiseksi, testaamiseksi ja ylläpidoksi (Sommerville 1995). Yleisesti käytössä on neljä mallia: vesiputousmalli, prototyypimalli, inkrementaalinen malli ja spiraalimalli.

4.2.1.1 Vesiputousmalli

Kaikista klassisin ja yleisimmin käytetty ohjelmistotuotannon prosessimalleista on *vesiputousmalli* (engl. waterfall model). Joskus siitä puhutaan myös klassisena elinkaarimallina (engl. classic life cycle model) tai lineaarisena peräkkäismallina (engl. linear sequential model). (Pressman 1997) Vesiputousmallia on käytetty aina 1970 –luvulta alkaen ja siitä on olemassa useita eri muunnelmia (Haikala & Märijärvi 1998). Mallissa prosessin vaiheet etenevät kuin vesiputous: seuraava vaihe alkaa kun edellinen on päättynyt. Vesiputousmallin tärkeimmät vaiheet ovat vaatimusten ja tiedon kerääminen eli esitutkimus, vaatimusten analyysi eli järjestelmän määrittely,

suunnittelu, toteuttaminen, testaus ja ylläpito. Mallia on kritisoitu siitä, että todellisten ohjelmistoprojektien vaiheet harvoin noudattavat niin peräkkäistä järjestystä kuin malli olettaa (Pressman 1997). Vesiputousmallissa oletetaan myös, että kaikki ohjelmiston vaatimukset saadaan selville kerralla ja heti prosessin alussa. Työnkulku ei mallissa ole aina paras mahdollinen, sillä vaiheiden peräkkäisyys saattaa aiheuttaa seisokkiaikoja osalle työntekijöistä, joka osaltaan aiheuttaa sen, että vesiputousmallin avulla on hidasta tuottaa ensimmäistäkään toimivaa versiota asiakkaalle (Pressman 1997).

4.2.1.2 Prototyypimalli

Prototyypimalli sopii erityisesti sellaisten projektien prosessimalliksi, joissa asiakkaalle on nopeasti saatava jokin valmis tulos. Mallissa edetään iteraatiokierroksittain, joihin jokaiseen kuuluu määrittely-, suunnittelu-, toteutus- ja testausvaiheet. Prototyypimallin seuraamisessa on kaksi pääsuuntausta, joista toisessa prototyyppiä käytetään vain mallina varsinaiselle tuotteelle ja toisessa valmiin prototyypin kehittämistä jatketaan ja sen ominaisuuksia täydennetään valmiiksi tuotteeksi (Haikala & Märijärvi 1998). Mallin yhtenä etuna pidetään sitä, että vuorovaikutus asiakkaan kanssa on tiivistä, jolloin asiakkaan vaatimukset on helppo kerätä vähä vähältä prototyypin edetessä. Ongelmana on pidetty muun muassa sitä, että asiakas saattaa luulla keskeneräistä tuotetta valmiiksi (Haikala & Märijärvi 1998). Toisaalta valmista prototyyppiä halutaan usein vähän vielä parantaa, jolloin iteraatiokierroksia tulee aina vain lisää ja prosessin päättäminen saattaa venyä (Pressman 1997).

4.2.1.3 Inkrementaalinen malli

Hiukan prototyypimallin kaltainen *inkrementaalinen eli evolutionaarinen malli* on käytännössä useimpien tuotekehityshankkeiden toteutuksen mallina (Pressman 1997). Gilbin (1988) mukaan mallin idea perustuu siihen, että ohjelmistotuotteet ja niiden vaatimukset joka tapauksessa kehittyvät ajan mittaan ja prosessin pitäisi pystyä suuntautumaan aina uudelleen muuttuvien vaatimusten ja uusien tavoitteiden

mukaisesti. Evolutionaarinen malli koostuu peräkkäisistä vesiputousmallin mukaisista sarjoista, joissa tuotteeseen lisätään pikkuhiljaa uusia ominaisuuksia (Gilb 1988). Se on tavallaan vesiputousmallin ja prototyypimallin välimuoto. Mallin tavoitteena on rakentaa tuotteesta ensin pelkkä ydin, jota sitten kehitetään edelleen seuraavissa projekteissa (Haikala & Märijärvi 1998, 30-31). Erona prototyypimalliin on, että jokaisen iterointikierroksen jälkeen tuote on tarpeeksi toiminnallinen, jotta se voidaan julkaista seuraavana versiona. Malli on erityisen käyttökelpoinen esimerkiksi silloin, kun kaikkia tarvittavia resursseja ei ole kerralla saatavilla. Kerrostamalla tuotteen kehittämistä myös tarvittavia resursseja voidaan paremmin aikatauluttaa. (Pressman 1997, 39-40)

4.2.1.4 Spiraalimalli

Spiraalimalli muistuttaa evolutionaarista mallia, sillä spiraalimallikin etenee kerroksittain, joissa samat vaiheet toistuvat. Se on kohtuullisen uusi prosessimalli, se on kehitetty vasta vuonna 1988 (Boehm 1988). Riskien hallinta korostuu spiraalimallissa, se on oma vaiheensa joka kierroksella. Malli on yleensä jaettu kolmesta kuuteen alueeseen, joita toistetaan joka kierroksella. Alueet ovat 1) kommunikointi asiakkaan kanssa, 2) suunnittelu, 3) riskien analysointi, 4) toteuttaminen, 5) kokonaisuuden muodostaminen ja sen julkaisu ja 6) asiakkaan suorittama arviointi. Mallin mukaisen prosessin ensimmäisten kierrosten tuloksena ei välttämättä ole läheskään toimintavalmis tuote, vaan tulokset näkyvät ehkä vasta paperilla. Spiraalimallia käytetään yleensä isojen järjestelmien kehittämiseen, sillä jatkuva riskien analysointi auttaa sekä kehittäjää että asiakasta hallitsemaan prosessia. Siitä huolimatta mallin ongelmaksi on nähty uskottavuuden luonti asiakkaan suuntaan. Samalla mallin käyttö vaatii käyttäjältään sekä kykyä että halua panostaa riskienhallintaan. (Pressman 1997; Haikala & Märijärvi 1998)

4.2.2. Prosessimallien merkitys

Prosessi- eli vaihejakomallien avulla ohjelmistoprosessia yritetään ymmärtää ja samalla kontrolloida (Notkin 1988). Prosessimalli toimii projektiryhmän yhteisenä viitekehyksenä, jonka avulla työtä jaksotetaan ja suunnataan. Onnistunut prosessimallin valinta auttaa työntekijöitä luomaan mielikuvan työn etenemisestä ja motivoi työhön ehkäisemällä turhia vaiheita.

Prosessin mallintamiselle ja prosessimalleille on tarvetta, sillä kun yhden tavoitteen toteuttamiseksi tarvitaan monta ihmistä, he tarvitsevat jonkun yhteisen koordinoitavan, jolloin prosessimalli voi toimia koordinoinnin apuna. Humphrey ja Kellner (1989) vertaavat ohjelmistoprosessia jalkapallon pelaamiseen: joukkueet, joilla on parhaiten harjoiteltu pelitapa ja jotka osaavat soveltaa sitä erilaisissa tilanteissa, pärjäävät kauden aikana parhaiten. Ohjelmistoprosessin suorittamista harvoin harjoitellaan etukäteen, muuten analogia toimii: ohjelmistojen tekeminen sujuu sitä laadukkaammin mitä jalostuneempi on yrityksen ohjelmistoprosessi. Ohjelmistoprosessin määrittäminen on samalla tekninen viitekehys ja johtamisen malli apuvälineiden ja menetelmien käytölle ja ihmisten työskentelylle ja oppimiselle (Humphrey & Kellner 1989).

4.3. Yhteenveto

Lyhyesti sanottuna, prosessimuotoisen työtavan tavoitteina on kulujen vähentäminen, ajan säästäminen, tuotteen laadun parantaminen ja työn mielekkyyden lisääminen (Davenport & Short 1990) informaation prosessointia hallitsemalla. Harvoin kuitenkaan lähdetään tavoittelemaan parannusta kaikilla näillä osa-alueilla. Tavoitteiden valinnassa joudutaan käymään eräänlaista vaihtokauppaa, sillä keinot vaikuttavat kaikki toisiinsa: se, mikä säästää aikaa prosessissa, saattaa samalla tuoda siihen lisää kustannuksia, kun taas kustannuksien säästö saattaa huonontaa tuotoksen laatua. Parhaan tuloksen saavuttamiseksi tavoitteiden asettamisessa täytyy tehdä kompromisseja (Davenport &

Short 1990). Eri tavoitteisiin pääsemiseksi hyödynnetään prosessimuotoisen työtavan moninaisuutta ja painotetaan prosessien eri puolia ja vahvuuksia.

Tässä luvussa prosessimuotoisen työtavan piirteitä käytiin läpi ja erityisinä vahvuuksina todettiin olevan asiakaslähtöisyys ja laatuun panostaminen. Prosessien vuorovaikutuksen kannalta tärkeää on yrityksen prosessien keskinäinen yhdenmukaisuus tai erilaisuus, joka määräytyy pitkälti käytettyjen prosessimallien mukaan. Prosessimallit ovat abstraktin tason kuvauksia suoritettavista tehtävistä ja ne tarjoavat toiminnalle viitekehyksen. Yhteisen viitekehyksen mukaan eri prosesseja voidaan muokata ja soveltaa kuhunkin projektiin sopiviksi. Erilaisista prosessien vaihejakomalleista esiteltiin yleisimmät ja niiden sopivuutta erilaisiin projekteihin arvioitiin.

Prosessien vuorovaikutuksen kannalta on tärkeää ymmärtää peruskäsitteet ja prosessimallien käsite, sillä ne ovat tutkimuksen kuvausvälineitä. Silti tulee muistaa prosessien erilaistuminen, joka sallii malliprosessin muokkaamisen tilanteeseen sopivaksi ja joka samalla erilaistaa vuorovaikutustilanteita.

5. PROSESSIA KOORDINOIVA VUOROVAIKUTUS

Ohjelmistotuotannossa tyypillisiä ongelmakohtia ovat muun muassa käyttäjätarpeiden ymmärtäminen, projektin laajuuden määrittäminen ja projektin muutosten hallinta (Reel 1999). Sommerville (1995) listaa tärkeimmiksi ongelmiksi niin ikään toimintojen integroinnin, projektin hallinnan ja kommunikaation. Phanin tutkimusryhmän (Phan, Vogel & Nunamaker 1995) raportoimassa tutkimuksessa vuorovaikutus ja siihen liittyvä koordinointi oli järjestyksessä kolmantena parannuskohtana (Toimintojen heikko koordinointi on ongelmien aiheuttajien kärkisijoilla myös Krautin ja Streeterin listalla (1995). Huomion kiinnittäminen ohjelmistoprosessin vuorovaikutukseen muiden prosessien kanssa ja sen kehittäminen voisi auttaa hallitsemaan näitä koordinoinnin puutteesta johtuvia ongelmia.

Tässä luvussa tutustutaan empiirisen tutkimuksen pohjana olevaan koordinointiteoriaan ja selitetään mitä prosessien vuorovaikutus on. Sen jälkeen tutustutaan tilanteisiin, joissa koordinoivaa vuorovaikutusta voi esiintyä ja pohditaan vuorovaikutuksen tavoitteita ja keinoja. Luvun lopuksi mietitään informaation roolia koordinoinnissa ja selitetään informaatiotulvan synty ja seuraukset.

5.1. Koordinointiteoria

Koordinointiteorian käsitteellä viitataan teorioihin koordinoinnin ilmenemisestä erilaisissa järjestelmissä. Se ei ole valmis teoria siinä mielessä kuin tieteessä yleensä teoriolla tarkoitetaan. Pää tarkoituksena teorian kehittämisenä on ollut tutkia informaatioteknologian aiheuttamaa ja edistämää yhteistyön muutosta ja toimintojen koordinointia erilaisissa työympäristöissä. Koordinointiteorian kehityksessä ovat olleet mukana monet tutkijat, joista eniten Malone ja Crowston (1990; 1994; Crowston 1997; Malone, Crowston, Lee, Pentland, Dellacoras, Wyner, Quimby, Osborn, Bernstein, Herman, Klein, & O'Donnell 1999). Teoria on poikkitieteellinen, sillä sen kehityksessä on käytetty hyväksi sekä tietotekniikan, taloustieteen että organisaatioteorian aiempaa tutkimusta ja käsitteistöä. (Malone & Crowston 1994)

5.1.1. Yleistä teoriasta

Koordinointia tarvitaan prosessien sujuvaan etenemiseen ja päällekkäisyyksien välttämiseen eli vuorovaikutuksen toteuttamiseen. Koordinointi on määritelty organisaation eri osien yhdistämiseksi yhteisen tehtäväjoukon tekemiseksi (Van de Ven, Delbecq & Koenig 1976, Kraut & Streeterin 1995, mukaan) Tämän tutkielman aiheeseen sopii paremmin Malonen ja Crowstonin (1990) kapeampi määrittely, jonka mukaan *koordinointi* on riippuvuuksien hallintaa niiden tehtävien välillä, joita tarvitaan tavoitteen saavuttamiseksi.

Taulukosta 1 käy ilmi koordinointiin kuuluvat osat, joita ovat tavoitteet, tehtävät, toimijat ja riippuvuudet. Niitä vastaavat koordinointiprosessit ovat tavoitteiden tunnistaminen, tavoitteiden pilkkominen tehtäviin, toimijoiden valinta ja riippuvuuksien hallinta. (Malone & Crowston 1990)

Tämän tutkielman puitteissa keskitytään vain viimeiseen alueeseen eli riippuvuuksien hallintaan, sillä riippuvuuksien hallinta edellyttää prosessien välistä vuorovaikutusta, kun taas tavoitteisiin, tehtäviin ja toimijoihin liittyvä koordinointi on enimmäkseen prosessin sisäistä toimintaa.

TAULUKKO 1. Koordinoinnin osat (Malone & Crowston 1990)

Osa	Esimerkki koordinointiprosessista
Tavoitteet	Tavoitteiden tunnistaminen
Tehtävät	Tavoitteiden pilkkominen tehtäviksi
Toimijat	Toimijoiden valinta, tehtävien jakaminen toimijoille
Riippuvuudet	Riippuvuuksien hallinta

Riippuvuuksia voidaan luokitella sekä yleisellä että kohdealueen tasolla (Malone & Crowston 1990). Riippuvuuksien päätyypit on esitelty taulukossa 2 ja ne ovat jaetut resurssit, tuottaja-kuluttaja –suhteet, samanaikaisuuden rajoitteet ja tehtävä-alitehtävä –suhteet. Tässä tutkielmassa tutkitaan tuottaja-kuluttaja –suhteiden riippuvuuksia, jotka

voidaan edelleen jakaa kolmeen osaan: edeltävyyden rajoituksiin, siirtämiseen ja käytettävyyteen. Näistä keskitytään tutkimaan edeltävyyden rajoituksia. Käytettävyyttä ja siirtämistä sivutaan vain tiedon siirtämisen osalta, sillä fyysisten tuotteiden osalta riippuvuuksien tutkiminen edellyttäisi syvällisempää tutustumista materiaalinhallintaan, mikä ei ole tämän tutkielman kannalta olennaista.

TAULUKKO 2. Esimerkkejä riippuvuuksista ja niiden koordinoinnista. (muokattu Malone & Crowstonin, 1994 mukaan)

Riippuvuus	Esimerkki koordinoitikeinosta
Jaetut resurssit	Ensin tullutta palvellaan (first come/first serve)
Tuottaja/kuluttaja suhteet	
Edeltävyyden rajoitukset	Ilmoittaminen, järjestykseen laittaminen, seuranta
Siirtäminen	Materiaalinhallinta (esim. just-in time), tiedonvälitys
Käytettävyys	Standardointi, käyttäjiltä kysyminen
Samanaikaisuuden rajoitukset	Aikatauluttaminen, synkronointi
Tehtävä/alitehtävä	Tavoitteen valinta, tehtävien osiin hajottaminen

5.1.2. Teorian sovelluksia

Koordinoititeoriaa on aiemmin sovellettu lukuisissa yhteyksissä. Malone ja Crowston esittelevät artikkelissaan (1990) lyhyesti koko joukon aiempia tutkimuksia, jotka on tulkittu koordinoititeorian sovelluksiksi. Tutkimukset sivuavat organisaatioteoriaa, ryhmätyötä ja toiminnan koordinoitintia ja ne on tehty jo kauan ennen koordinoititeorian julkaisemista, eikä niissä siten ole käytetty koordinoititeorian käsitteitä.

Myöhemmin koordinoititeoriaa on sovellettu muun muassa organisaatioiden prosessikäsitteiden tekemisessä (Malone ym. 1999). Käsikirja muodostuu tietokannasta, johon on tallennettu eri organisaatioiden prosesseja hierarkkisesti ja erilaisista käyttöliittymistä, joilla prosessien hierarkkinen koostumus pystytään esittämään. Hierarkia on kehitetty oliomallintamisen ja koordinoititeorian pohjalta, jolloin

oliomallintamisesta tuttu periytyminen tuo käsikirjaan selvän prosessien jaottelu- ja luokittelutavan. Koordinointiteorian avulla pystytään tunnistamaan yritysten prosessien heikkouksia ja esittämään niihin parannuksia.

Toisenlaisen lähestymistavan koordinointiin ovat esittäneet Toffolon ja Dakhli (2000). He tunnistivat riippuvuuksien esiintymisen sekä prosessin sisällä että prosessien välillä ja ovat Malonen ja Crowstonin kanssa samoilla linjoilla siitä, että koordinoinnilla voidaan hallita ohjelmistoprosessin eri osapuolten välillä olevia riippuvuuksia vastaamalla kysymyksiin kuka tekee mitä ja milloin.

Poiketen Malonen ja Crowstonin esittämästä koordinointiteoriasta, Toffolon ja Dakhli (2000) esittävät, että koordinointi koostuu viidestä spiraalista: neljästä prosessin sisäisestä ja yhdestä prosessien välisestä. Sisäisistä spiraaleista kukin vastaa omasta alueestaan: ongelmien, arkkitehtuurin, tuottamisen ja käytön alueista. Spiraalit ovat toisistaan riippuvaisia ja niillä on kullakin kaksi puolta, muodollinen ja epämuodollinen puoli. Ohjelmistoprosessissa koordinointi kuvailee molempien puolien sellaisia käytäntöjä, joilla sopimuksia tekijästä, tekemisen kohteesta ja ajasta (kuka, mitä, milloin) toteutetaan.

Toffolon ja Dakhli (2000) testasivat teoriaansa ranskalaisessa vakuutusyhtiössä, jonka ohjelmistoprosessin yhteyteen he toteuttivat kuvailemansa koordinointiprosessin. Tutkimustuloksena he totesivat, että koordinoinnin on oltava ainakin osittain automatisoitua toimiakseen mahdollisimman tehokkaasti. Automatisoinnilla, esimerkiksi työnkulku (engl. workflow) –ohjelmistoilla, voidaan tukea nimenomaan spiraalien muodollisia puolia ja tällä tavoin hallita riippuvuuksia ja tehostaa prosessien vuorovaikutusta.

5.1.3. Riippuvuuksien hallinta

Prosessit ovat riippuvaisia toisistaan vuorovaikutuksessa liikkuvan tiedon välityksellä. Kun riippuvuuksien esiintyminen tunnetaan, niitä voidaan myös paremmin välttää.

Vaatimusmäärittelyjen muuttuminen on esimerkki riippuvuuksien syntyisestä: kun vaatimusmäärittelydokumenttia päivitetään, täytyy myös suunnitteludokumenttia päivittää: suunnitteludokumentti riippuu vaatimusmäärittelydokumentista.

Joskus riippuvuuksia halutaan luoda tarkoituksella (Crowston 1994). Esimerkiksi, jotta kahden komponentin liitos onnistuu, niiden täytyy olla riippuvaisia toisistaan. Perustana vuorovaikutuksen onnistumiselle on prosessien välinen työn organisointi, koordinointi ja kommunikointi.

Suunnittelua voidaan hyödyntää koordinoitikeinona tilanteissa, joissa prosessien riippuvuusuhdet ovat molemminpuolisia, esimerkkinä ohjelmistoprosessin ja projektinhallintaprosessin välinen riippuvuusuhde. Tällöin kullekin suhteen osapuolelle, esimerkissä sekä hallinnollisille että ohjelmistokehityksen tehtäville, asetetaan aikataulut ja toimintaa ohjataan niiden avulla. Suunnittelu sopii dynaamisiin, alati muuttuviin ympäristöihin. (Thompson 1974, 74)

Koordinointi voi tapahtua myös *molemminpuolisen sopeutumisen* avulla. Erityisesti silloin, kun toimintatilanteet muuttuvat jatkuvasti eikä niitä pystytä etukäteen kunnolla ennustamaan, tarvitaan molempien osapuolien aktiivista joustamista ja sopeutumista. Kolmantena koordinoitikeinona, ja samalla riippuvuuksien hallintakeinona, on *standardointi*, joka tarkoittaa käytännössä yhteisten sääntöjen ja rutiinien luomista ja noudattamista. Jotta standardointi onnistuisi, täytyy toimintatilanteiden olla suhteellisen samankaltaisia keskenään, eivätkä ne saa muuttua liian nopeasti tai liian usein. (Thompson 1974, 74-75; ks. myös Galbraith 1973, 10)

Erilaisia koordinoititapoja voidaan käyttää rinnakkain tukemaan prosessien vuorovaikutusta erilaisissa ja muuttuvissa tilanteissa. Eroista huolimatta koordinoitikeinoilla on yhteisiäkin piirteitä: kaikissa niissä korostuu kommunikaation ja päätöksenteon tärkeys. Jokainen koordinoititapa aiheuttaa myös kustannuksia ja joskus kustannukset voivat olla hyvinkin huomattavia (Thompson 1974, 75), joten on järkevää miettiä mikä tapa tai menetelmä olisi paras ja edullisin kulloisessakin vuorovaikutustilanteessa. Kustannuksia arvioidessa tulee huomioida, että

vuorovaikutustavoilla ja menetelmillä voi olla vaikutuksia muihinkin prosesseihin kuin vain välittömiin vuorovaikutuksen osapuoliin. (ks. Kaltio 1995).

5.2. Koordinoiva vuorovaikutus

Prosessien vuorovaikutus tarkoittaa prosessien välistä tiedonvaihtoa. Tietoa voi olla erityyppistä ja tiedonvaihtoa prosessien välillä voi tapahtua monesta syystä. Koordinoivalla vuorovaikutuksella tarkoitetaan vuorovaikutuksella olevan mahdollisuus vaikuttaa prosessin etenemiseen. Kahden prosessin keskinäisen vuorovaikutuksen voidaan ajatella tehostavan molempien prosessien toimintaa kokonaisuutena.

5.2.1. Koordinoivan vuorovaikutuksen käsite

Prosessia koordinoivaa vuorovaikutusta voi tapahtua kaikissa prosessin eri vaiheissa. Vuorovaikutuksen tapoja on monia, joista tuottaja-kuluttaja –suhteet on yksi tavallisimmista. (ks. esim. Kahen & Lehman 2000) Tiedon vaihdolla pyritään tekemään vuorovaikutuksen molempien osapuolten työstä sujuvampaa ja vähentämään epätietoisuutta ja työn seisonta-aikaa. Tarve vuorovaikutukselle syntyy aina, kun prosessit ovat jollakin tavalla toisistaan riippuvaisia, esimerkiksi kun yksi prosessi tarvitsee edetäkseen tuotetta tai ohjaustietoja toiselta prosessilta. Näitä riippuvuuksia hallitaan koordinoinnilla. Sisäisten riippuvuussuhteiden lisäksi ohjelmistoprosessin laajuus, sen etenemisen ennustamisen vaikeus ja kommunikoinnin epämuodollisuus tekevät ohjelmistoprosessia koordinoivasta vuorovaikutuksesta elintärkeää. (Kraut & Streeter 1995) Ohjelmistoprosessin koordinoinnin tarvetta lisää prosessin keskeisyys yrityksen liiketoiminnassa (ks. Galbraith & Nathanson 1978, 59).

Koordinointi on iteratiivista niin kuin ohjelmistojen kehitysikin, jolloin koordinoinnin luonne muuttuu käytetyn prosessimallin ja organisaation yleisten toimintamallien mukaan. Koska koordinointi tukee ohjelmistoprosessia, sen täytyy olla tiukasti integroituna ohjelmistoprosessiin. Tämä tarkoittaa, että koordinointi vaihtelee

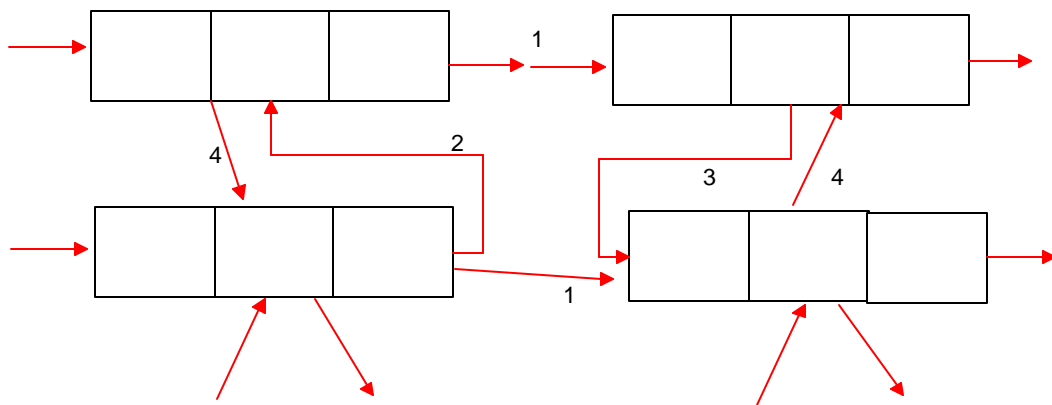
ohjelmistoprosessissa käytetystä vaihejakomallista riippuen, (Toffolon & Dakhli 2000) jolloin myös vuorovaikutuksen keinot vaihtelevat.

Prosessien välistä, koordinoivaa vuorovaikutusta on tutkittu aiemmin vain vähän ja usein hiukan eri näkökulmasta, kuten Krautin ja Streeterin koordinoititutkimuksessa (1995). Siinä tutkittiin ohjelmistoprojektiryhmien välisten koordinoitikäytäntöjen ohella sellaisia projektien rakenteellisia piirteitä, joilla voisi olla vaikutusta koordinointiin ja lisäksi projektien menestymistä eri asteikoilla. Tutkimus tehtiin survey-kyselynä 65:lle projektiryhmälle ja sen tuloksina saatiin muun muassa, että henkilökohtainen kommunikointi yli tehtävärajojen on tärkeä työn koordinoitikeino projektissa työskenteleville. Varsinkin suurten projektien kohdalla painotettiin myös muodollisten menetelmien tärkeyttä. Tutkimuksesta voidaan vetää johtopäätös, että henkilökohtainen tiedonvaihto on olennaista myös yksittäisen projektin ulkopuolella, koko ohjelmistoprosessia koordinoivassa vuorovaikutuksessa.

Kaltion (1995) raportoimassa tutkimuksessa tutkittiin kahden eri aliprosessin välistä vuorovaikutusta ohjelmistoprosessin sisällä. Tutkimuksessa käytettiin SPICE:n mukaista ohjelmistoprosessin kuvausta ja keskityttiin mallintamaan kokoonpanon hallintaprosessin vuorovaikutusta muiden aliprosessien kanssa. Varsinaista kokeellista tutkimusta ei tehty lainkaan vaan tutkimustulokset perustuivat prosessin mallintamisen hyödyntämiseen. Tutkimustuloksena löydettiin useita vuorovaikutussuhteita kokoonpanon hallintaprosessin ja muiden aliprosessien, esimerkiksi testauksen ja projektinhallinnan, kanssa. Vuorovaikutussuhteista todettiin, että ne tulee ottaa huomioon suunniteltaessa muutoksia kokoonpanon hallintaprosessiin, sillä muutokset todennäköisesti vaikuttavat myös toisiin vuorovaikutussuhteiden osapuoliin. Tutkimustulosten perusteella tällainen vuorovaikutuksen heijastuminen toisiin prosesseihin on todennäköistä myös ylemmän tason prosessien välillä.

5.2.2. Vuorovaikutustilanteita

Kuviossa 12 on esitetty, mitä prosessien välinen vuorovaikutus voisi olla. Kuvio on yhdistetty ja muokattu Humphreyn (1989) ohjelmistoprosessista ja Kahenin ja Lehmanin (2000) esittämistä ohjelmistoprosessin palautesilmukoista. Nuolet kuviossa kuvaavat prosessien välistä vuorovaikutusta. Kuvioista näkyy, että vuorovaikutusta tarvitaan monissa eri tilanteissa. Tyypillisimpiä vuorovaikutustilanteita on neljä: 1) prosessin tulos on toisen prosessin syöte tai 2) tulos on toiselle ohjaustieto tai 3) toisen antama palaute on toiselle syöte ja 4) toisen antama palaute on toisen prosessin ohjaustieto. (vrt. esim. Kahen & Lehman 2000) Vuorovaikutuksessa käytettävät syöte, tulos, palaute ja ohjaustieto voivat olla sekä fyysisiä esineitä että abstraktia tietoa, joiden liikkumisesta voi olla kiinni koko prosessien eteneminen. Prosessien lisäksi vuorovaikutuksella voi olla vaikutuksia myös organisaation muihin osiin.



KUVIO 12. Prosessien vuorovaikutus (vrt. Humphrey 1989, 257; Kahen & Lehman 2000)

5.2.3. Epävarmuuden hallinta

Organisaatioiden täytyy olla jatkuvassa vuorovaikutuksessa sekä ympäristönsä että omien sisäisten toimijensa kanssa, sillä organisaation kohdistuvat muospaineet tulevat sekä ulkoa että sisältä. Nämä muospaineet aiheuttavat epävarmuutta. (Rogers & Agarwala-Rogers 1976, 64; Frank & Brownell 1989, 82) Organisaation sisäisen

vuorovaikutuksen rooli korostuu, kun epävarmuutta pyritään edes osittain hallitsemaan muutostöiden avulla.

Vuorovaikutuksessa tietoja vaihdetaan, jolloin kumpikin osapuoli saa itselleen arvokasta tietoa asioiden tilasta ja etenemisestä. Tieto muuttaa eri epävarmuustekijöiden esiintymisen todennäköisyyksiä (Rogers & Agarwala-Rogers 1976, 64), joten tiedon ja vuorovaikutuksen avulla voidaan hallita epävarmuutta ja tällä tavoin hallita prosessin kulkua. Tieto voi myös lisätä vaihtoehtojen määrää, joten organisaation sisäinen epävarmuus kokonaisuutena ei ehkä vähene, mutta sen muoto voi muuttua paremmin hallittavaksi.

Rogers ja Agarwala-Rogers määrittelevät epävarmuuden käsittämään vaihtoehtojen lukumäärän suhteessa tapahtuman esiintymiseen, jolloin kunkin vaihtoehdon esiintymisen todennäköisyys tulee huomioida (Rogers & Agarwala-Rogers 1976, 64). Koordinoinnin aihealueeseen sopii paremmin Galbraithin määritelmä, jonka mukaan *epävarmuus* on tietyn tehtävän tekemiseksi vaadittavan tietomäärän ja nykyisen tietomäärän erotus. (Galbraith 1973, 5)

Epävarmuuden määrä on suoraan verrannollinen käsiteltävän informaation määrään, eli mitä enemmän tehtävän suorittamisessa on epävarmuutta, sitä enemmän informaatiota täytyy käsitellä tehtävän suorittamiseksi. (Galbraith 1973, 4) Järjestyksen, ennustettavuuden ja vakauden puute ilmentää epävarmuutta organisaatiossa (Rogers & Agarwala-Rogers 1976, 64), joten näihin panostaminen tuo varmuutta organisaation toimintaan. Keinot epävarmuuden hallintaan ovat organisaation sisäisessä rakenteessa, jolloin prosessimuotoisuuden joustavuus, matala hierarkia ja sen mahdollistama prosessien vuorovaikutus ovat avainasemassa. (ks. Galbraith 1973, 11)

Prosesseilla ja niiden vuorovaikutuksella hallitaan sekä asiakassuhdetta, kuten kohdassa 4.1.3 todettiin, että organisaation muutospainesta johtuvaa toiminnan epävarmuutta. Asiakassuhteiden ja epävarmuuden hallinnan lisäksi vuorovaikutuksella on myös muita tavoitteita.

5.2.4. Vuorovaikutuksen tavoitteet

Koordinoivan vuorovaikutuksen tavoitteena on tehostaa sen molempien osapuolien suoritusta. Esimerkiksi ohjelmistoprosessin tapauksessa, ohjelmistoprosessia koordinoivan vuorovaikutuksen huomioiminen voi tehostaa samalla sekä ohjelmistoprosessia että projektinhallintaprosessia. Tämä seuraa siitä, että vuorovaikutuksen sisältämä koordinointi vaatii informaation käsittelyä, joka vähentää epävarmuutta (Frank & Brownell 1989, 82). Näin vuorovaikutus tuo varmuutta koko ohjelmistoprosessin kulkuun, joka vaikuttaa edelleen siihen kietoutuneen projektinhallintaprosessin kulkuun.

Koko prosessin tehostumiseksi tärkeää on molempien vuorovaikutuksen osapuolten aktiivinen osallistuminen. Kummassakin osallistuvassa prosessissa tulee olla sellaisia tehtäviä, jotka erityisesti mahdollistavat vuorovaikutuksen onnistumisen ja prosessien sujakkaan kulun. Nämä tehtävät ovat sellaisia, että ne voidaan tehdä joustavassa järjestyksessä, jottei turhia odotusaikoja tule. Ne ovat eräänlaisia vara- tai täytetehtäviä ja niitä joudutaan usein varta vasten järjestelemään tai niiden suoritusta muuten seurataan tarkasti, jotta ne valmistuisivat ajoissa toisen prosessin käyttöön. (ks. Malone & Crowston 1994)

Täytetehtävien, mutta myös muiden tehtävien valmistumisesta tiedottaminen on olennaista vuorovaikutuksen onnistumiseksi. Joskus tiedottamiseen liittyy fyysisen tuloksen siirtäminen toiseen paikkaan toisen prosessin käytettäväksi, esimerkiksi prototyyppejä kehitettäessä tai missä tahansa tuotekehitysprosessissa. (Malone & Crowston 1994) Yleensä konkreettisten tuotteiden puute havaitaan helpommin kuin abstraktin informaation puuttuminen.

5.3. Informaation rooli koordinoinnissa

Tarpeellisen informaation puuttumisesta johtuva prosessin epävarmuus vähentää mahdollisuutta suunnitella prosessin suoritusta etukäteen. Suunnitelmallisuuden

puuttuessa tiedonvaihtoa tarvitaan enemmän prosessin suorituksen aikana. Prosessin sisältämän tehtävän tekemiseksi tarvittavan informaation määrä riippuu tehtävässä tuotettavien tuloksien määrästä ja keskinäisestä vaihtelusta, tehtävään tulevien syötteiden määrästä ja tehtävän vaikeusasteesta (Galbraith 1973, 4-5; vrt. Thompson 1974, 75). Tämä tarkoittaa, että informaation rooli prosessien välisessä vuorovaikutuksessa on monimutkainen.

5.3.1. Koordinoiva informaatio

Organisaatioiden sisällä liikkuu informaatiota pääosin kahdentyyppisissä tuotevirroissa. Toinen tuotevirroista käsittää fyysiset, konkreettiset tuotteet ja toinen pitää sisällään abstraktit tietovirrat. (Medina-Mora, Winograd, Flores & Flores 1992) Kockin ja McQueenin tutkimuksen mukaan (1996) materiaalisten tuotevirtojen osuus tutkituissa organisaatioissa oli vain 15% ja loput 85% olivat abstraktimpia tietovirtoja. Tutkitut organisaatiot edustivat monikansallisia teollisuusyrityksiä ja uusiseelantilaisia valtion virastoja.

Tiedolla ja sen liikkumisella on suuri merkitys organisaatioiden toiminnalle. Usein puutteet tiedonkulussa ja vuorovaikutuksessa, niin prosessien välillä kuin muuallakin organisaatioissa, havaitaan vasta sitten kun todellisia koordinoitongelmia esiintyy. Davenportin mukaan (1993) juuri informaatio on se ”liima” joka pitää organisaation prosessirakenteen kasassa.

Tiedonhallinnan kannalta tärkeää on ne organisaation rakenteet, joilla on vastuu erilaatuisesta informaatiosta, esimerkiksi standardeista ja viranomaismääräyksistä. Tiedon pitäisikin olla järjestetty prosessien mukaan eikä esimerkiksi yrityksen eri toimintojen mukaan. Yllättävän suuri rooli vuorovaikutuksessa voi olla niillä henkilöillä, joilla on eri alojen tietoa. Siksi eri prosesseissa työskentelevien pitäisi olla tietoisia toisistaan ja toistensa asiantuntemuksesta. Tällöin he osaisivat etsiä tarvitsemaansa tietoa oikealta henkilöltä epämuodollisia kanavia pitkin. (Davenport 1993)

Epämuodollinen eli epäformaali kommunikointi on merkittävä tiedon tuottamisen tapa ja tiedonkulun kanava. Pikkutiedolta tuntuva palaute ei välttämättä päädy prosessin ohjaustiedoksi virallisia kanavia pitkin, mutta muutaman minuutin epävirallisella juttelulla eri prosesseissa työskentelevät henkilöt voivat saada hyvin olennaista tietoa prosessien etenemisestä ja tuloksista. (ks. esim. Andrews & Herschel 1996; Curtis, Krasner & Iscoe 1988; Toffolon & Dakhli 2000)

Tiedon käytettäväksi tekeminen ottaa kantaa tiedon muotoon. (ks. Malone & Crowston 1994) Karkea esimerkki tästä on dokumentti, joka on kirjoitettu kielellä, jota muut eivät ymmärrä. Dokumentin tieto voi olla vaikka kuinka olennaista ja oikea-aikaista, mutta se ei ole käytettävää jos sen muoto on väärä. Toinen esimerkki on tietokoneen kovalevyllä oleva tiedosto, joka ei ole vastaanottajan käytettävissä ennen kuin se on tulostettu paperille. Tiedon muodon lisäksi tulee huomioida sen välitystapa.

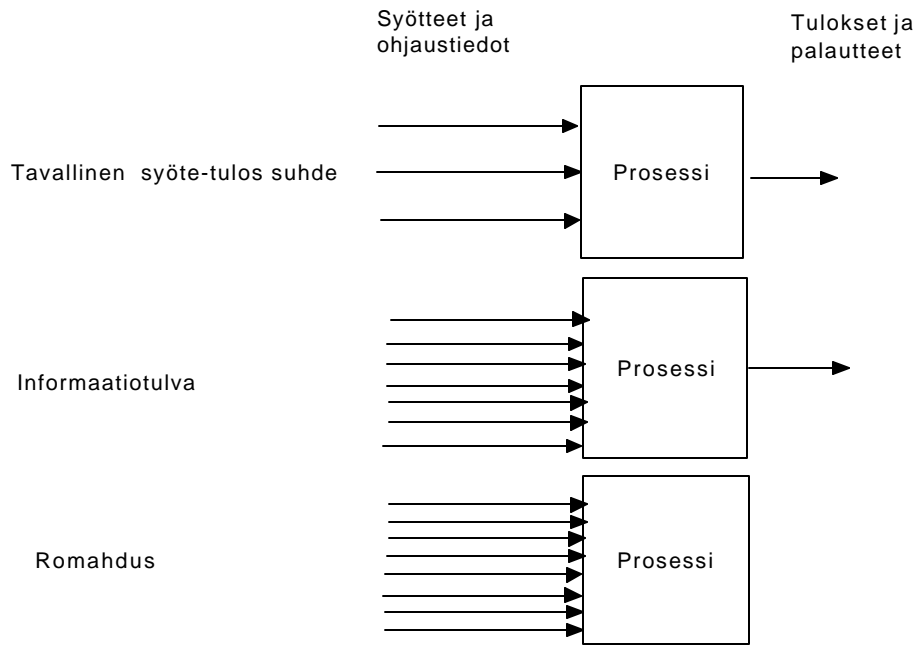
Tiedon välitystavalla on merkitystä yhtäläillä tiedon kulun kuin sen säilytyksenkin suhteen, sillä yhden prosessin keräämä tieto voi olla hyödyksi toiselle prosessille vasta paljon myöhemmin. Tällöin tiedon keräävä prosessi tarjoaa tiedon käytettäväksi osana omia tehtäviään, muussa tapauksessa tietoa tarvitseva prosessi etsii haluamansa tiedon itse. Tässä voidaan erottaa kahdenlaista toimintatapaa, tiedon tarjoaminen (engl. push) ja tiedon pyytäminen (engl. pull). Näistä vain tiedon pyytäminen vaatii tiedon tarvitsijan aktiivista toimintaa. Tiedon tarjoamisen mallissa ongelmana on tiedon säilytys ja löytäminen. (Davenport 1993)

Toisinaan prosessit tarvitsevat vuorovaikutuksessaan tietoa, joka on saatu ulkoa, esimerkiksi tietokannoista (Davenport 1993), ja tiedon määrä voi kasvaa nopeasti yli hallittavien rajojen. Tiedon suodattamiseen tulisi kiinnittää huomiota, sillä tiedon kokonaismäärällä on merkitystä prosessien tehokkuuden kannalta.

5.3.2. Informaatiotulvan ongelma

Prosessi suodattaa ympäristöstään syötteitä ja ohjaustietoja, jotka ovat prosessin toiminnan energiaa. Syöte ja ohjaustieto voivat olla joko fyysisiä raaka-aineita, abstraktia tietoa tai jonkin tehtävän teko. (Rogers & Agarwala-Rogers 1976) Ilman ohjaustietoja toiminta on mahdollista, mutta ilman syötteitä prosessilta loppuu polttoaine. Työnkulkujärjestelmissä esimerkiksi, yhden tehtävän teko ja tieto siitä ovat syötteitä seuraavalle tehtävälle. Seuraavaa tehtävää ei voi tehdä, ennen kuin edellinen tehtävä on suoritettu ja tieto tehtävän suorituksesta on annettu eteenpäin. Tavallaan kyse on viestijuoksusta, jossa kukin suorittaa oman osansa ja siirtää kapulan eteenpäin. Syötteiden laatu voi vaihdella suuresti, mutta kummatkin syötteet, sekä tehtävän teko että tieto sen tekemisestä, ovat yhtä lailla tärkeitä.

Syötteiden, ohjaustietojen ja tulosteiden määrän pitää olla keskenään tasapainossa, jotta prosessien vuorovaikutuksesta olisi todella hyötyä. Jos vuorovaikutus lisääntyy niin, että syötteitä ja ohjaustietoja on huomattavasti enemmän kuin tuloksia, puhutaan informaatiotulvasta. Kuviosta 13 näkyy, miten informaatiotulva syntyy, kun syötteitä ei pystytä käsittelemään ja prosessoimaan riittävästi. Pitkään jatkuessaan tilanne saattaa riistäytyä käsistä, jolloin prosessointi vähenee niin, ettei sen tuloksena synny lainkaan tulosta tai palautetta. (ks. Rogers & Agarwala-Rogers 1976, 90-91) Informaation määrän rajoittaminen koskee myös suullista kommunikointia. Luomalla toimintasääntöjä voidaan suullisen kommunikoinnin tarvetta vähentää ja tällä tavalla keventää informaatiokuormaa. (ks. Galbraith 1973, 10)



KUVIO 13. Informaatiotulva ja prosessoinnin romahdus (muokattu Rogers & Agarwala-Rogers 1976, 90 mukaan)

5.4. Yhteenveto

Kokonaisuutena prosesseja koordinoiva vuorovaikutus on monitahoinen mekanismi, jossa liikkuu monenlaista informaatiota eri tarkoituksia varten. Koordinointi voidaan ajatella myös prosessien keskinäisten riippuvuussuhteiden hallinnaksi, yhtäläillä kuin ympäristöstä ja organisaation sisältä nousevan epävarmuuden hallintakeinoksi. Vuorovaikutuksen osapuolet hyötyvät koordinoinnista sitä enemmän, mitä aktiivisempia ne ovat vuorovaikutussuhteissaan. Tällöin positiiviset vaikutukset saattavat levitä myöhempiin prosessien vaiheisiin muodostaen eräänlaisen ketjureaktion, jolloin koko prosessin toiminta tehostuu.

Kommunikaatiota voidaan pitää yhtenä tuottaja/kuluttaja –suhteiden hallintakeinona tiedon osalta. Koordinointitapoja voidaan muuttaa ja luoda muuttamalla kommunikaation tapoja. (Malone & Crowston 1994) Tiedon käytettäväksi tekeminen on tärkeää, jotta tiedosta on vastaanottajalle todellista hyötyä. Vuorovaikutuksessa tulee ottaa huomioon sekä tiedon välitystapa että sen muoto. Jos tiedon hyödyntäminen on vastaanottajalle mahdotonta tai liian hidasta, tärkeästäkin tiedosta voi tulla arvotonta.

Jossakin määrin tiedonvaihtoa tulee rajoittaa, jotta vältetään informaatiotulvan aiheuttamalta romahdukselta, jossa tietoa ei enää pystytä käyttämään hyväksi prosessissa. Tiedon rooli prosessien vuorovaikutuksessa on sekä tärkeä että monimutkainen ja sen tutkiminen on vaikeaa. Seuraavassa luvussa selostetaan empiirisen tutkimuksen menetelmät ja tutkimuksen kulku.

6. TUTKIMUSASETELMA

Prosessien vuorovaikutuksen tutkimus on prosessin toiminnan analysointia, johon voidaan valita erilaisia näkökulmia. Yksi vaihtoehto on käsittää prosessi organisaation rakenteena, kuten tutkielman toisessa luvussa esitettiin. Toinen näkökulma on tehtävien koordinoinnin näkökulma, josta kerrottiin edellisessä luvussa ja jota sovelletaan tämän tutkielman empiirisessä osassa. Koordinoitinäkökulman avulla pystytään selvittämään, miten prosessit toimivat organisaation sisällä ja miten ne vaikuttavat toistensa etenemiseen. Tässä luvussa selitetään prosessien vuorovaikutuksen tutkimisen tutkimusmalli ja esitellään tapaustutkimuksen kohdeorganisaatio. Lisäksi kerrotaan tiedon keräämisen ja analysoinnin menetelmät ja kuvaillaan kerättyä aineistoa.

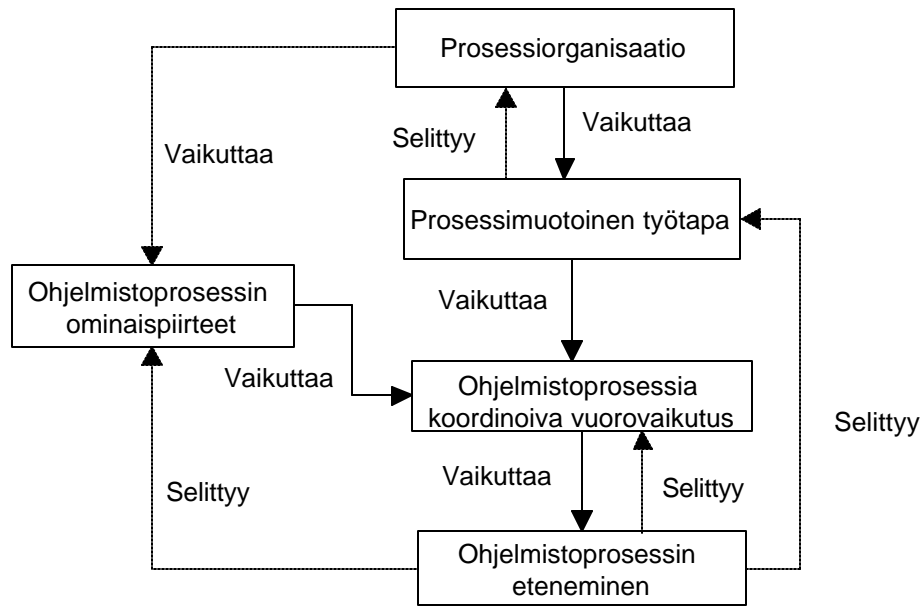
6.1. Tutkimusmalli

Olellista tutkimuksessa on ohjelmistoprosessin etenemisen tarkasteleminen yhtä lailla prosessimuotoisen työtavan, ohjelmistoprosessin ominaispiirteiden kuin koordinoivan vuorovaikutuksen kannalta. Tutkimuksen tavoitteena on kuviossa 15 esitetyn tutkimusmallin avulla vastata seuraaviin kysymyksiin:

- Millä prosesseilla tutkittavaa ohjelmistoprosessia koordinoidaan?
- Missä ohjelmistoprosessin kohdissa koordinointi vaikuttaa eniten tutkittavan ohjelmistoprosessin kulkuun?
- Miten koordinointiin panostaminen näissä kohdissa voisi tehostaa ohjelmistoprosessia yleisesti?

Keskeisin tekijä tutkimuksessa oli ohjelmistoprosessia koordinoiva vuorovaikutus. Sen selvittämiseksi muodostettiin käsitys ohjelmistoprosessin ominaispiirteistä ja koordinoivan vuorovaikutuksen luonteesta. Ohjelmistoprosessin etenemistä selittävää prosessimuotoisesta työtapaa ja sen taustalla olevaa prosessiorganisaation rakennetta jäsennettiin muodostamalla käsitys prosessimuotoisuudesta ohjelmistotuotannossa.

Prosessiorganisaation ja ohjelmistoprosessin suhde on tutkimuksessa tärkeä, sillä tutkimuskohteena olleessa yrityksessä organisaatorakenne on muokattu sulautettujen järjestelmien valmistamiseen sopivaksi, joka näin ollen vaikuttaa ohjelmistoprosessin piirteisiin ja sitä kautta myös sitä koordinoivaan vuorovaikutukseen ja edelleen prosessin etenemiseen.



KUVIO 15. Tutkimuksen tutkimusmalli.

Tutkimuskohde ja -aihe on valittu Jyväskylän yliopiston Tietotekniikan tutkimusinstituutissa käynnissä olleen PISKO-projektin tiimoilta. Kuten tämän tutkielman johdannossa kerrottiin, PISKOssa on kolmen vuoden ajan keskitytty lähinnä ohjelmistoprosessin sisäisiin tapahtumiin, eikä ohjelmistoprosessin yhteyksiä muihin yrityksen prosesseihin ole juurikaan huomioitu. PISKO-projektin kannalta tämän tutkielman myötä ohjelmistoprosessista saadaan entistä kokonaisvaltaisempi kuva, kun sen vuorovaikutussuhteet tulevat paremmin tunnetuksi. Tällöin seuraavissa Tietotekniikan tutkimusinstituutin ohjelmistotuotannon projekteissa osataan paremmin huomioida yrityksen muiden prosessien vaikutus ohjelmistoprosessiin.

Yritykselle tutkimuksesta voi olla monenlaisia hyötyjä. Ohjelmistoprosessin vuorovaikutussuhteiden tutkiminen selventää yrityksen työskentelytapoja, jolloin

prosessien myöhempi kehittäminen on helpompaa (ks. Saukkonen & Oivo 1998, 14). Tutkimuksella voidaan löytää mahdollisia ohjelmistoprosessin ongelmakohtia, jolloin myöhemmissä tutkimuksissa voidaan paremmin keskittyä niiden korjaamiseen. Lisäksi vuorovaikutussuhteiden analyysi voi antaa yritykselle vinkkejä erilaisista vuorovaikutuksen toteuttamisvaihtoehdoista ja niiden seurauksista, jolloin yrityksessä voidaan välttyä liian raskaiden käytäntöjen omaksumiselta.

6.2. Tapauksena keskisuuri ohjelmistoyritys

Empiirinen tutkimus suoritettiin tapaustutkimuksena yhdessä yrityksessä. Tapaustutkimus valittiin tutkimusmenetelmäksi, koska se soveltuu hyvin organisaation toiminnan tutkimiseen, jossa tarkoituksena on kuvata nykyisiä toimintatapoja ja jolloin tutkijan vaikutusmahdollisuudet tutkimustulokseen ovat vähäiset. (ks. Yin 1994) Muilla tutkimusmenetelmillä ei olisi päästy riittävän syvälliseen toiminnan ymmärtämiseen kohtuullisilla resursseilla.

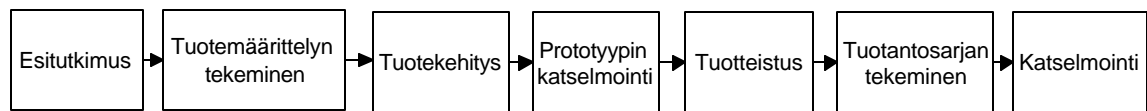
Tapaustutkimuksen kohteena oli keskisuomalainen keskisuuri yritys, joka on yksi neljästä PISKO-projektissa mukana olevasta yrityksestä. Päätoimenaan yritys valmistaa ajoneuvotietokoneita, joilla ajoneuvokalustot voidaan liittää osaksi yritysten tietojärjestelmiä. Ohjelmistot sulautetaan yrityksen valmistamaan päätelaitteistoon. Yrityksen liikevaihtoennuste tälle vuodelle on 45 miljoonaa markkaa ja henkilöstöä on noin viisikymmentä.

Kohdeyrityksen ohjelmistosuunnittelijat ja muut tuotekehitykseen osallistuvat työskentelevät samoissa fyysisissä tiloissa, jolloin vain sermit erottavat työpisteet toisistaan. Myynnin ja markkinoinnin työntekijät työskentelevät Helsingissä. Haastateltavat kuvailivat työympäristöä avoimeksi ja rennoksi, jossa jäykkiä työkäytäntöjä pyritään välttämään milloin se vain on mahdollista. Haastatteluiden perusteella kohdeorganisaatiossa on vallalla prosessimainen työtapo, jota organisaation matala ja joustava hierarkia hyvin tukee. Työskentelyssä painotetaan kommunikointia, käytännönläheisyyttä ja prosessi-tiimien toimivuutta.

Ohjelmistot tuotetaan projektissa, joka seuraa määriteltyä prosessimallia soveltuvin osin. Prosessimalli etenee pääpiirteissään vesiputousmallin mukaisesti. Yrityksen ohjelmistoprosessissa vaiheita on enemmän ja vaiheiden tehtävät poikkeavat hiukan vesiputousmallin tehtävistä, mutta muilta osin prosessin etenemistapa on mallin mukainen. Prosessimallia sovellettaessa ohjelmistoprojekteihin mallia muokataan kutakin tuotettavaa tuotetta varten tuotteen laadun takaamiseksi. Laadun parantaminen, niin tuotteiden kuin prosessinkin laadun osalta, on yrityksessä ollut erityisen huomion kohteena PISKO-projektin aikana.

6.2.1. Tutkittava ohjelmistoprosessi

Tutkimuskohteena oli yrityksen ohjelmistoprosessi, joka on kuvattu kuviossa 14. Se alkaa esitutkimuksesta ja tuotemäärittelyn tekemisestä ja jatkuu tuotekehitykseen ja edelleen prototyypin katselmointiin. Katselmoinnin jälkeisessä tuotteistuksessa huomioidaan erityisesti ohjelmiston tekeminen myyntivalmiiksi. Tuotteistuksen jälkeen seuraa tuotantosarjan tekeminen, minkä jälkeen tuote jälleen katselmoidaan. Prosessin tuloksena on myyntivalmis ohjelmistotuote kaikkine tarvittavine dokumentteineen ja ohjeistuksineen. Ohjelmistoprosessin varrella on neljä eri etappia, joita varten on asetettu tietyt tavoitteet.



Kuvio 14. Kohdeorganisaation ohjelmistoprosessi.

Yrityksessä tuotettavia ohjelmistotyyppisiä on seitsemän erilaista, jotka poikkeavat toisistaan hiukan sekä kooltaan että monimutkaisuudeltaan. Kaikkien ohjelmistotyyppien tuottamiseen käytetään tutkittavana olleen ohjelmistoprosessin sovellettuja versioita. Kohdeyrityksen ohjelmistoprosessille on leimaavaa ohjelmistoprosessin ja laitteistontuotantoprosessien tiivis rinnakkainasettelu, mikä on tyyppillistä erityisesti sulautettuja järjestelmiä valmistavalle yritykselle (Saukkonen &

Oivo 1998, 17). Erityispiirre on tutkimuksessa otettu huomioon toimijoiden määrittelyssä ja prosessin rajaamisessa.

Yrityksen ohjelmistoprosessi on määritelty PISKO-projektin yhteydessä vuonna 2001. Lähteenä käytettävät ohjelmistoprosessin prosessikuvaukset on tehty PISKOssa kehitetyllä seinätekniikalla (ks. Hakulinen, Halttunen, Taskinen & Ahonen 1999) keväällä 2001. Tutkimusmateriaalina käytetyt dokumenttipohjat on tehty PISKOssa aiempina vuosina.

6.2.2. Tutkimusmenetelmä ja -prosessi

Prosessien vuorovaikutuksen tutkimisessa on lähdetty luvussa 5 esitellyn koordinoititeorian (Crowston 1994) pohjalta. Näkökulmassa painotetaan tehtävien tekemistä yhteistyön sujumisen edistämiseksi, jolloin tutkitaan tehtävien välisiä riippuvuussuhteita. Yksinkertaisuuden ja selvyuden vuoksi riippuvuuksia tutkittiin vain ohjelmistoprosessin kannalta.

Riippuvuussuhteet jaoteltiin kolmeen tyyppiin, kuten edellisessä luvussa todettiin (ks. taulukko 2). Näistä riippuvuuksista tutkimuksessa keskityttiin eniten tuottaja/kuluttaja – suhteiden riippuvuuksiin, varsinkin edeltävyyssrajoitteisiin riippuvuuksiin. Siirtämisen riippuvuuksia käsiteltiin vain tiedon siirtämisen osalta, jolloin fyysisten tulosten siirtämiseen ei puututtu lainkaan. Resurssien käyttöä ei tutkittu.

Valittu tiedonhankintamenetelmä pohjautuu Milesin ja Hubermanin (1984) esittelemään ja Crowstonin käyttämään (1997) tapaustutkimukseen soveltuvan laadullisen tutkimuksen menetelmään (ks. myös Harrison 1998). Teemahaastattelun muokkaus perustuu Yinin (1994) teokseen.

Käytettävä tiedonhankintamenetelmä koostui kolmesta osasta. Ensimmäiseksi ohjelmistoprosessin toimijat tunnistettiin, jotta voitiin tunnistaa ohjelmistoprosessin ja muiden prosessien välillä oleva vuorovaikutus. Toisessa vaiheessa ohjelmistoprosessi

rajattiin sen tehtävien avulla erilliseksi kokonaisuudeksi. Tunnistaminen ja rajaaminen tehtiin tutkimalla dokumentteja, joista kävi ilmi toiminnan eteneminen ja luonne. Tällaisia dokumentteja olivat ohjelmistoprosessista aiemmin tehdyt prosessikuvaukset ja organisaatiossa käytössä olevat dokumenttipohjat. Dokumenteista etsittiin mainintoja tehtävien tekijöistä ja erityisesti sellaisista tehtävistä, joissa tarvitaan tietoa prosessin ulkopuolelta. Tämän avulla prosessi pystyttiin alustavasti rajaamaan ja sen toimijat listattiin. Kolmanneksi selvitettiin ohjelmistoprosessin tehtävien eteneminen ja luonne, jolloin erilaiset vuorovaikutustavat ja koordinoitikeinot selvitettiin. Tätä varten prosessissa toimijoita haastateltiin. Kahdessa haastattelussa lisäksi tarkennettiin dokumenttien perusteella tehtyjä oletuksia prosessin toimijoista ja rajauksesta. Molemmat haastattelut tehtiin ryhmähaastatteluina.

Ensimmäisellä kerralla haastateltavina olivat tuotekehityspäällikkö, projektipäällikkö ja vanhempi pääsuunnittelija. Haastattelussa oli esillä seinätekniikalla (ks. Hakulinen ym. 1999) tehty prosessikuvaus kohdeyrityksen ohjelmistoprosessista, joka oli rajattu käsittämään tutkimuksessa käsiteltävän osan. Kunkin prosessin vaiheen kohdalla mietittiin, tapahtuuko vaiheessa vuorovaikutusta muiden prosessien kanssa. Jos vuorovaikutusta havaittiin, edettiin Crowstonin (1997) käyttämän prosessianalyysin mukaan (ks. myös Malone & Crowston 1994; Biazzo 2000), jolloin jokaisen vaiheen kohdalla kysyttiin

1. millaista informaatiota toimijat saivat,
2. keneltä ja mistä prosessista he sen saivat,
3. miten tieto saatiin,
4. miten tietoa käytettiin hyväksi ohjelmistoprosessissa ja
5. kenelle tuloksesta tiedotettiin.

Ensimmäisellä kysymyksellä selvitettiin vuorovaikutuksessa kulkevan tiedon laatu ja luonne. Toinen kysymys paljasti koordinoivan prosessin ja kolmas kysymys koski käytössä olevia koordinoitikeinoja. Samassa yhteydessä keskusteltiin myös kunkin vuorovaikutussuhteen tiiviyydestä. Neljännellä kysymyksellä pyrittiin rajaamaan tutkimuskohdetta siten, että mukaan otettiin vain ne prosessit, joilla on vaikutusta

ohjelmistoprosessin etenemiseen. Rajauksesta keskusteltiin haastattelun aikana ja se selveni haastattelun kuluessa. Viidennellä kysymyksellä selvitettiin vuorovaikutuksen mahdollinen kaksisuuntaisuus.

Kun kysymykset oli käyty läpi, edettiin seuraavaan prosessin vaiheeseen. Tutkija merkitsi vastaukset muistiin etukäteen tehtyyn taulukkoon (ks. Miles & Huberman 1984, 112 ja 122). Osanottajien luvalla haastattelu myös äänitettiin. Ääninauhaa käytettiin tietojen kokoamiseen ja jälkitarkistuksiin.

Toisessa haastattelussa paikalla oli kaksi ohjelmistosuunnittelijaa ja haastattelun vetäjänä toimi toinen tutkija. Haastattelussa tarkennettiin prosessien käymää vuorovaikutusta ohjelmistoprosessissa olevien roolien ja niiden vuorovaikutussuhteiden avulla. Ensin ohjelmistoprosessin roolit listattiin, minkä jälkeen mietittiin niiden vuorovaikutussuhteita muihin prosesseihin. Roolien kautta esiin tuli paljon asioita, jota aiemmassa haastattelussa ei oltu ymmärretty kysyä. Aiheeseen saatiin tuore näkökulma, sillä toisin kuin ensimmäiset haastateltavat, toisessa ryhmähaastattelussa läsnä olleet ohjelmistosuunnittelijat eivät olleet mukana prosessikuvauksia tehtäessä.

Aineiston ja prosessikuvauksen puhtaaksikirjoituksen jälkeen aineistoa tarkasteltiin ohjelmistoprosessin kannalta, keskittyen toiminnan koordinointiin ja sen ohjelmistoprosessia tehostaviin vaikutuksiin. Haastatteluaineistoa verrattiin dokumentteihin ja alkuperäisiin prosessikuvauksiin ja etsittiin epätavallisia poikkeamia, jotka saattavat olla virheitä. Apuna käytettiin haastattelusta tehtyä ääninauhaa. Havaitut epäjohdonmukaisuudet tarkistettiin kohdeyrityksestä ja korjattiin. Haastattelusta ja dokumenteista saatua tietoa analysoitiin etsimällä vuorovaikutuskohdista sekä yhteisiä piirteitä että eroavuuksia.

Kumpaankin ryhmähaastatteluun käytettiin noin kaksi tuntia ja ne tehtiin viikon välein, jolloin ensimmäisen haastattelun aineistoa pystyttiin joiltakin osin tarkentamaan toisessa haastattelussa. Aineistona käytetyt dokumentit olivat PISKO-projektin saatavilla koko tutkimuksen keston ajan. Lopullinen aineisto muodostui kahdesta ryhmähaastattelusta ja noin kymmenestä erilaisesta dokumentista, joilla

ohjelmistoprosessia tuetaan ja ohjeistetaan. Dokumentit olivat pääosin mallipohjia ja muistilistoja eri vaiheiden suorittamista varten. Monet niistä olivat useampisivuisia. Lisäksi aineistoon kuului yrityksen ohjelmistoprosessista aiemmin tehdyt graafinen ja sanallinen prosessikuvaus.

Aineiston purkamisessa ja alustavassa analyysissä käytettiin apuna Milesin ja Hubermanin (1984, 112 ja 122) esittelemiä taulukoinnin ja vuokaavion malleja. Tulokset kuvataan SPICEn avulla, painottaen koordinoinnin merkitystä ohjelmistoprosessille.

6.3. Tutkimusaineiston kuvailu

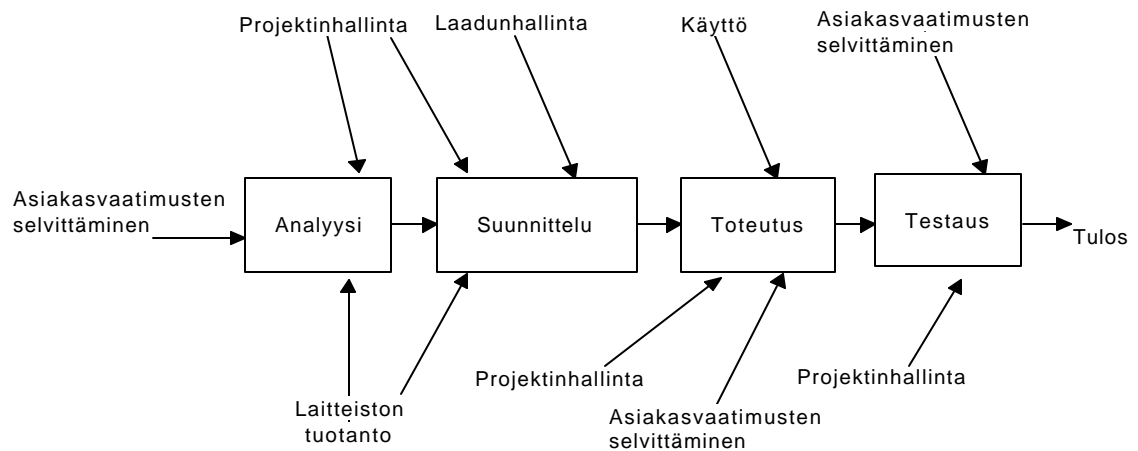
Aineiston kuvailun tarkoituksena on kuvata, millaista empiiristä aineistoa tutkimuksessa kerättiin. Tärkein osa kerättyä tutkimusaineistoa on ohjelmistoprosessin kohdat, joissa koordinointia tapahtuu eniten, sekä koordinoivat prosessit ja niiden käyttämät koordinointikeinot. Näiden lisäksi aineistosta ilmenee haastateltujen näkemyksiä ohjelmistoprosessin etenemisestä ja yrityksen työskentelymenetelmistä.

6.3.1. Koordinoivat prosessit tutkitun ohjelmistoprosessin eri vaiheissa

Tutkimustulosten yleistettävyyden säilyttämiseksi tutkittava ohjelmistoprosessi rajattiin käsittämään vaiheet esitutkimuksesta prototyypin katselmointiin saakka. Tällöin alkuperäisestä yrityksen ohjelmistoprosessista on otettu käsittelyyn ohjelmistoprosessin vesiputousmallia (ks. luku 4.3.1; Pressman 1997) vastaava osa, jolloin tutkimuskohteena olleen ohjelmistoprosessin vaiheiden nimistä huolimatta vaiheet vastaavat sisällöltään vesiputousmallin analysointi-, suunnittelu-, toteutus- ja testausvaiheita.

Kuviossa 16 on tutkittua ohjelmistoprosessia koordinoivat prosessit kuvattuna vaiheittain rajatun prosessin mittaisella aikaulottuvuudella kalanruotomallin (Harrison 1998, 142) avulla. Tutkimustuloksia analysoidaan luvussa 3.3 esitetyn SPICEn avulla,

jolloin tutkimustulosten käsittelyn yhtenäisyyden takaamiseksi ja helppolukuisuuden säilyttämiseksi kohdeyrityksen omat prosessit on nimetty uudelleen vastaamaan SPICE:n prosesseja. SPICE valittiin analysointivälineeksi, koska sen näkemys ohjelmistotuotannosta perustuu prosesseihin ja se on maailmanlaajuisesti käytössä oleva ISO:n standardi (ks. esim. Nevalainen 1999).



KUVIO 16. Tutkimuksessa käsiteltävä ohjelmistoprosessin osa ja sitä koordinoivat prosessit. Koordinoivat prosessit ja ohjelmistoprosessin vaiheet on nimetty uudelleen.

Haastattelujen perusteella yrityksen myynti- ja markkinointiprosessissa tunnistetaan asiakkaiden tarpeet ja kommunikoidaan jatkuvasti asiakkaan kanssa. Näin ollen yrityksen myynti- ja markkinointiprosessi vastaa sisällöltään SPICE-mallin prosessia asiakasvaatimusten selvittäminen. Yrityksen myynninseurantaprosessissa asiakasta opastetaan ja autetaan tuotteen käytössä ja tarvittaessa muodostetaan asiakastuki auttamaan ongelmissa. Samantapaiset tehtävät on SPICE:n käyttö –prosessissa, johon sisältyvät sekä käyttöpalvelu että asiakastuki. Projektinhallinta ja laadunhallinta sisältyvät sellaisinaan SPICEen. Ainoastaan laitteistontuotantoprosessin nimi on säilytetty kohdeyrityksen nimeämiskäytännön mukaisena, sillä laitteistontuotantoprosessi on ominainen juuri sulautettuja järjestelmiä tuottavalle yritykselle, eikä se näin ollen ole SPICE:n mukainen.

6.3.2. Koordinointikeinot

Prosessien käyttämät koordinointikeinot poikkesivat hiukan toisistaan, mikä on tiivistetty taulukkoon 3. Projektinhallintaprosessin ja ohjelmistoprosessin väliset riippuvuudet johtuivat enimmäkseen edeltävyyusrajoitteisista tuottaja-kuluttaja-suhteista, mutta myös samanaikaisuuden rajoituksista ja jaetuista resursseista syntyviä riippuvuuksia esiintyi. Koordinointikeinoina olivat aikatauluttaminen, tehtävien edistymisen seuranta ja tehtävien järjestykseen laittaminen.

Taulukon mukaan asiakasvaatimusten selvittämisprosessin ja tutkitun ohjelmistoprosessin väliset tuottaja-kuluttaja-suhteet olivat siirtämistä. Käytetyt riippuvuuksien hallintamenetelmät olivat enimmäkseen tiedonvälitys ja standardointi. Käytön prosessi koordinoi ohjelmistoprosessia ilmoittamisella ja standardoimisella. Prosessin suhde tutkittuun ohjelmistoprosessiin oli siirtämisen suhde. Laadunhallintaprosessin suhde sitä vastoin oli käytettävyys-suhde, jossa koordinointikeinona käytettiin standardointia. Laitteistontuotantoprosessin tuottaja/kuluttaja-suhde ohjelmistoprosessiin oli sekä edeltävyyusrajoitteinen että siirtämissuhde. Koordinointikeinoina olivat tehtävien seuranta ja tiedonvälitys.

TAULUKKO 3. Koordinoitavien prosessien suhteet tutkittuun ohjelmistoprosessiin ja prosessien käyttämät koordinointikeinot.

Ohjelmistoprosessia koordinoiva prosessi	Tuottaja/kuluttaja-suhde	Prosessin käyttämät koordinointikeinot
Projektinhallinta	Edeltävyys	Aikatauluttaminen, seuranta järjestykseen laittaminen
Asiakasvaatimusten selvittäminen	Siirtäminen	Tiedonvälitys, ilmoittaminen
Käyttö	Siirtäminen	Ilmoittaminen, standardointi
Laadunhallinta	Käytettävyys	Standardointi
Laitteistontuotanto	Edeltävyyden rajoitus, siirtäminen	Seuranta, tiedonvälitys

Tutkimusaineiston mukaan kohdeyrityksessä ohjelmistoprosessin koordinointi toteutetaan pitkälti dokumentoinnin avulla. Osaltaan tähän saattaa vaikuttaa se, että

yrityksessä on äskettäin uudistettu dokumentointikäytäntöjä ja uusia mallipohjia on otettu käyttöön. Dokumenttien ohella koordinoivaa tietoa siirtyy yhteisissä palavereissa. Palaveriin osallistuu ihmisiä aina niistä prosesseista, jotka eniten koordinoivat ohjelmistoprosessia kyseisessä vaiheessa.

Kohdeorganisaatiossa samalla henkilöllä saattaa olla eri rooleja eri prosessien alueelta, suunnittelija voi esimerkiksi olla asiakastuen roolissa. Tällöin prosessien vuorovaikutus on tavallaan automaattista ja vuorovaikutussuhde saattaa jäädä huomaamatta prosessin määrittelyssä. Vaarana tässä on, ettei prosessien vuorovaikutuksesta jää jälkiä, koska sitä ei dokumentoida kuten useimpia muita vuorovaikutussuhteita. Tällöin vuorovaikutuksessa liikkuva prosessia koordinoiva tieto ei välttämättä siirry eteenpäin muille prosessissa työskenteleville.

6.3.3. Tutkitun ohjelmistoprosessin erityispiirteitä

Tutkituista dokumenteista ja haastatteluista ilmeni, että kohdeorganisaatiossa laitteistontuotantoprosessi kulkee hyvin tiiviisti ohjelmistoprosessin rinnalla. Haastateltavien oli vaikea puhua pelkästään ohjelmistonkehityksestä, sillä laitteistontuotanto on olennainen osa yrityksen liiketoimintaa, mikä ilmenee hyvin seuraavasta:

-Vaikee kuvitella niinku meill' ei oikeestaan sellasta pelkkää softaa ole, jota tehtäis ilman rautaa ku me myydään noita laitteita.

-Ni se helposti kääntyy aina siihen [laitteistontuotantoon].

Laitteistontuotantoprosessi yksinkertaistaa ohjelmistoprosessia koordinoivaa vuorovaikutusta huomattavasti. Tutkimusaineiston mukaan laitteiston asettamat vaatimukset on helpompi välittää ohjelmistoprosessiin kuin asiakkaiden asettamat vaatimukset. Laitteiston asettamat vaatimukset ovat haastattelujen perusteella selkeitä ja yksikantaisia, tulkintavaraa ei ole eikä laitteisto muuta vaatimuksiaan kuten asiakkaat saattavat tehdä. Joskus osa laitteistovaatimuksista saattaa jäädä huomioimatta

analyysivaiheen vaatimusten keruussa, jolloin niihin tulee palata ohjelmistoprosessin myöhemmissä vaiheissa.

Osa prosessien välisestä vuorovaikutuksesta on itsestäänselvyys prosessissa työskenteleville, sillä pienessä yrityksessä lähes kaikki osallistuvat ohjelmistonkehitykseen jollakin tapaa. Seuraavasta lainauksesta ilmenee, miten tämä vaikeuttaa vuorovaikutussuhteiden erittelyä ja ilmenemistä haastatteluissa:

-Se tieto kyl menee kaikille, tossa vaiheessa sinänsäkin ne on kaikki jo sotkeutuneet siihen [ohjelmistonkehitykseen] mukaan.

Ohjelmistoprosessin tärkein vaihe on toteutusvaihe. Se kestää kauemmin kuin muut vaiheet ja se sisältää monta alitehtävää. Vaiheen aikana käydään paljon vuorovaikutusta muiden prosessien kanssa, mikä sekä ohjaa että saattaa myös hidastaa vaiheen etenemistä. Haastateltavat pitivät toteutusvaihetta prosessin sekä tärkeimpänä että vaikeimpana vaiheena:

-Toihan on niinku koko ton prosessin haastavin etappi että sitä ei niinku tavallaan taho saada pidettyä, se [ohjelmistonkehitys] ei oo niin pitkällä et sen sais tuotekehityksestä ulos.

-Niin se on, kyllä, koska siellä [toteutusvaiheessa] on niin paljon tehtäviä.

Toteutusvaiheen etenemiseen vaikuttaa analyysivaiheen suorittamistapa. Jos analyysivaihe on tehty huolella, toteutusvaihe nopeutuu. Haastateltujen mukaan toteutusvaiheen ongelmana ovat tuotemäärittelyn ulkopuolelta tulevat ohjelmiston lisävaatimukset, varsinkin silloin kun lisävaatimukset tulevat tiedoksi ilmoitusluontoisesti, kuten seuraavasta ilmenee:

-Sanotaanko et jos toi esitutkimus ja tuotemääritys on oikein hyvin tehty, ni niiten ei tarviis tuolla tuotekehityksessä hirveesti sorkkia sen myynnin ja markkinoinnin.

-Nii, eikä aftersalesinkaan [myynninseuranta], koska siellä on ne kaikki jo määritelty.

-Käytännössä sieltä kuitenkin tulee ainakin ideoita ja kysymyksiä tohon vaiheeseen [toteutusvaihe].

-Toiveita lisäominaisuuksista tietysti saattaa tulla kesken kaiken.

-Tietysti jos näitä lisätoiveita tulee, ja riittävän korkealta ja kovalla äänellä ni mikäpäs siinä sitten.

6.4. Yhteenveto

Tutkimusaineiston yhteenveto on esitetty taulukossa 4. Taulukosta ilmenee ohjelmistoprosessia koordinoivat prosessit kussakin ohjelmistoprosessin vaiheessa, koordinoinnin laji tuottaja/kuluttaja –suhteena ja koordinoivien prosessien yleisimmin käyttämät koordinointikeinot. Koordinoivia prosesseja olivat projektinhallintaprosessi, asiakasvaatimusten selvittäminen, käytön prosessi ja laadunhallintaprosessi. Taulukon mukaan projektinhallintaprosessi koordinoi ohjelmistoprosessia sekä aikatauluttamalla että seuraamalla ohjelmistoprosessin tehtäviä. Muita koordinoivien prosessien käyttämiä koordinointikeinoja ovat tiedonvälitys, ilmoittaminen ja standardointi. Varsinkin standardointi on tärkeää valmistettavan ohjelmiston laadun takaamiseksi. Ilmoittaminen tarkoittaa sitä, että esimerkiksi asiakasvaatimusten selvittämisprosessista saatetaan kertoa ilmoitusluonteisesti asiakkaalta tullutta palautetta, johon ohjelmistoprosessin on mukauduttava.

TAULUKKO 4. Tutkimusaineiston yhteenveto.

Ohjelmistoprosessia koordinoiva prosessi	Ohjelmistoprosessin vaiheet	Tuottaja/kuluttaja –suhde	Koordinointikeinot
Projektinhallinta	Analyysi, suunnittelu, toteus, testaus	Edeltävyyden rajoitus	Aikatauluttaminen, seuranta järjestykseen laittaminen
Asiakasvaatimusten selvittäminen	Analyysi, toteutus, testaus	Siirtäminen	tiedonvälitys, ilmoittaminen
Käyttö	Toteutus	Siirtäminen	Ilmoittaminen, standardointi
Laadunhallinta	Suunnittelu	Käytettävyys	Standardointi
Laitteistontuotanto	Analyysi, suunnittelu	Edeltävyyden rajoitus, siirtäminen	Seuranta, tiedonvälitys

7. TUTKIMUSTULOSTEN JOHTAMINEN JA ANALYSOINTI

Tässä luvussa raportoidaan tutkimuksen tulokset ja niistä vedetään johtopäätöksiä. Tuloksia analysoidaan tarkastelemalla prosesseja SPICE:n avulla. Tulosten pääseikat kootaan yhteen ja asetettuihin tutkimusongelmiin vastataan. Lisäksi pohditaan mikä laajempi merkitys tutkimustuloksilla voisi olla. Tutkimuksen luotettavuutta arvioidaan luvun lopuksi.

7.1. Millaista on ohjelmistoprosessia koordinoiva vuorovaikutus?

Haastatteluissa ilmeni, että tyypillisimpiä vuorovaikutustilanteita tutkitussa ohjelmistoprosessissa ovat tilanteet, joissa toisen prosessin tulosta käytetään prosessin syötteenä. On tärkeää muistaa, että prosessin tuloksia voi olla monenlaisia, esimerkiksi sivutuotteita, puolivalmisteita tai jätteitä (ks. kohta 2.2.3).

Haastattelujen ja tutkittujen dokumenttien perusteella yrityksen ohjelmistoprosessin edeltävyysrajoitteisia riippuvuuksia hallitaan eniten tehtävien etenemisen tarkalla seurannalla, jonka perusteella tehtävien järjestystä saatetaan vaihtaa. Ohjelmistoprosessin suorittamisessa olennaista on hallita prosessien välisiä riippuvuuksia voimassa olevien standardien ja muiden yhteisten säännöksiä avulla. Siten vältetään yllättäviä muutoksilta ja niiden aiheuttamilta ongelmilta.

Tärkeitä koordinoivia prosesseja ovat projektinhallintaprosessi ja asiakasvaatimusten selvittäminen prosessi. Eri prosessien tarjoaman tiedon luonne poikkeaa toisistaan, sillä muut prosessit koordinoivat ohjelmistoprosessin kulkua laadullisella tavalla, mutta projektinhallintaprosessin tarjoama koordinointi on enemmänkin aikatauluihin liittyvää (ks. kohta 6.3.2).

Asiakasvaatimusten selvittäminen prosessi koordinoi ohjelmistoprosessia välittämällä tietoa asiakas- ja markkinatilanteesta. Ohjelmistotuotteen onnistuminen on erityisen

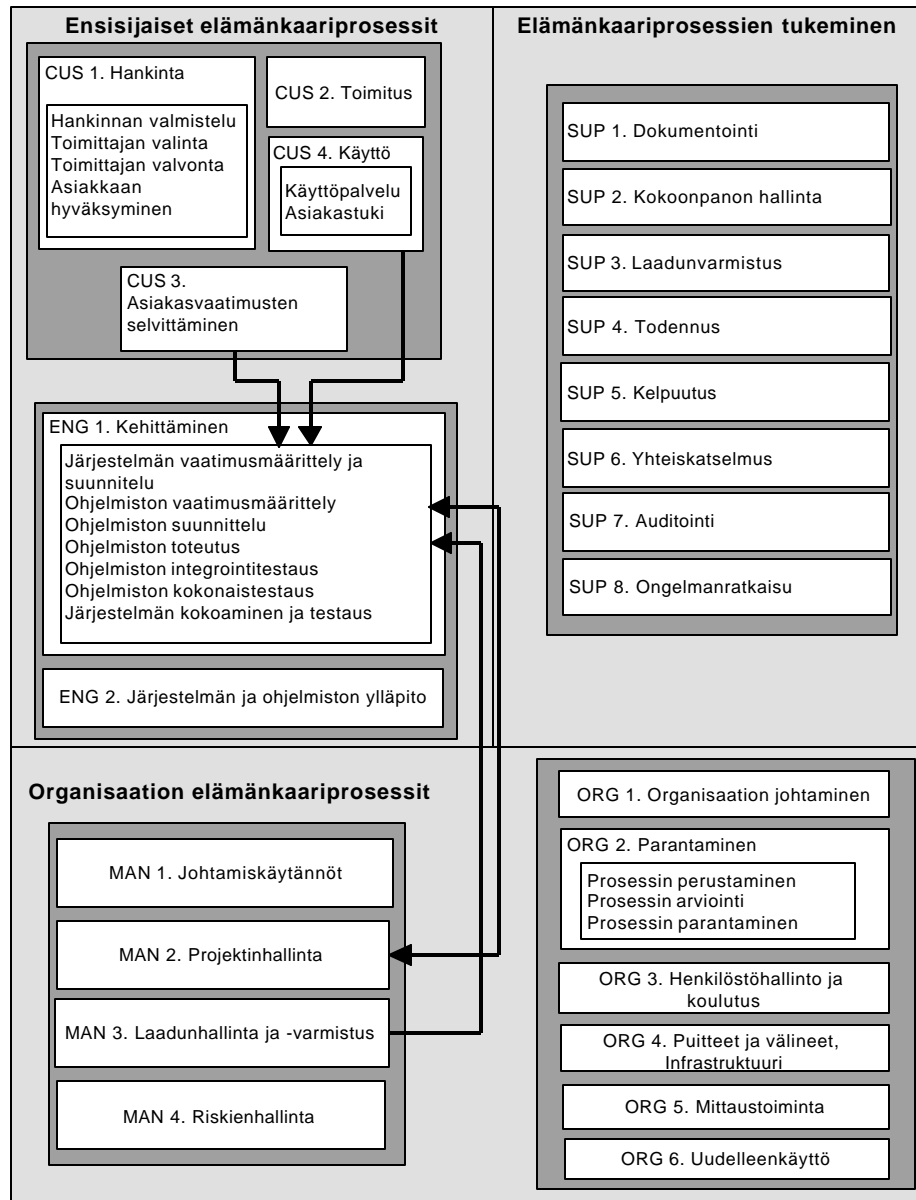
riippuvainen tästä vuorovaikutussuhteesta, sillä suhteessa välittyvän tiedon avulla tuotetta osataan muokata asiakkaalle sopivaksi.

7.1.1. Koordinoivat SPICE:n prosessit

Kuvion 17 mukaisesti ohjelmistoprosessia koordinoivat prosessit jakautuivat SPICE:n kahteen prosessiryhmään: ohjelmistoprosessilla on vuorovaikutusta sekä asiakastoimittaja -ryhmän prosessien että hallinnan prosessiryhmän kanssa. Organisaation prosessiryhmän ja tuen prosessiryhmän prosesseilla ei havaittu olevan suoraa koordinoivaa vaikutusta ohjelmistoprosessiin.

Tuen prosessiryhmän vuorovaikutussuhteiden puuttuminen yllättää, sillä tukiprosessien luonteeseen kuuluu tukea ja avustaa muita prosesseja. Ohjelmistoprosessin tukeminen ei liene mahdollista ilman vuorovaikutussuhdetta, mitä myös Kaltio korostaa omassa tutkimuksessaan (Kaltio 1995). Tutkimustulosta osittain selittää tutkimuksen rajaus, jolla tutkimukseen on otettu mukaan vain ne vuorovaikutussuhteet, jotka koordinoivat ohjelmistoprosessin etenemistä. Sellaisia vuorovaikutussuhteita, joilla ei ole suoranaista vaikutusta ohjelmistoprosessin etenemiseen, ei ole tutkimuksessa huomioitu (ks. kohta 6.2.2). Näin ollen tukiprosessien vuorovaikutussuhteet ohjelmistoprosessiin saattavat olla vähemmän kriittisiä ohjelmistoprosessin etenemiselle.

Toinen vuorovaikutussuhteiden puuttumista selittävä tekijä on kohdeyrityksen prosessien määrittelyn uutuus. Yrityksessä ei vielä ole määritelty perusteellisesti kaikkia sen käyttämiä prosesseja, jolloin osa tukiprosesseista saattaa kulkea niin tiiviisti ohjelmistoprosessin osana, ettei niiden erottelu ollut mahdollista valitulla tiedonhankintamenetelmällä.



KUVIO 17. Ohjelmistoprosessin koordinointiin osallistuvat SPICE:n prosessit.

Yksi tärkeimmistä ohjelmistotuotantoprosesseista koordinoivista prosesseista on asiakas-toimittaja –ryhmän asiakasvaatimusten selvittämisprosessi. SPICE:n mukaan sen tavoitteena on kerätä, prosessoida ja jäljittää asiakkaan tarpeita ja vaatimuksia ja muodostaa vaatimukset tarvittaville ohjelmistotuotteille (ISO 15504-5). Asiakasvaatimusten selvittämisestä ohjelmistoprosessi saa alkusykäyksensä ja suunnan etenemiselleen. Vuorovaikutussuhteessa kulkee muun muassa asiakkaan vaatimukset tuotettavalle tuotteelle ja yleinen myynti- ja markkinatilanne tuotettavan tuotteen osalta.

Näillä tiedoilla asiakasvaatimusten selvittämisprosessi koordinoi etenkin ohjelmistoprosessin analyysi- ja testausvaiheita. Vuorovaikutus jatkuu läpi koko ohjelmistoprosessin, vaikka ohjelmistoprosessia koordinoivan tiedon sisältö muuttuukin matkan varrella. Analyysivaiheessa koordinoiva tieto on ennen kaikkea asiakkaiden vaatimuksia, kun taas toteutus- ja testausvaiheessa vuorovaikutussuhteessa liikkuu enemmän markkinatilanteeseen liittyvää informaatiota. Koordinointitapana käytetään eniten tiedonvälitystä ja ilmoittamista.

Käytön prosessi on toinen ohjelmistoprosessia koordinoiva asiakas-toimittaja -ryhmän prosessi. Käytön prosesseista asiakastuki on olennainen, sen tavoitteena on tuottaa asiakkaalle avustusta ja konsultaatiota ohjelmiston käyttöä varten. Ohjelmistoprosessia koordinoivaa tietoa ovat mahdolliset käytön ongelmat ja esiin tulleet ohjelmiston virheet, joiden selvittäminen antaa ohjelmistosuunnittelijoille arvokasta tietoa ohjelmiston toiminnallisuudesta ja käytettävyydestä. Ilmoittaminen ja standardointi ovat tärkeimmät koordinointikeinot.

Hallinnan prosessiryhmän laadunhallinta- ja -varmistusprosessi on myös merkittävä ohjelmistoprosessia koordinoiva prosessi. Se koordinoi enimmäkseen standardoimalla ohjelmistoprosessin tehtävien suoritusta. SPICEssa laadunhallinta- ja varmistusprosessin tavoitteena on valvoa tuotteiden tai palvelujen laatua ja varmistua asiakkaan tyytyväisyydestä (ISO 15504-5). Laadunhallinnalla on tärkeä rooli tutkitun ohjelmistoprosessin koordinoijana erityisesti ohjelmistoprosessin suunnitteluvaiheessa. Myöhemmissä vaiheissa laadunhallintaa tarvitaan vähemmän, tällöin vuorovaikutus ohjelmistoprosessin kanssa on lähinnä tarkastusluonteista, ei välttämättä toimintaa koordinoivaa. Vuorovaikutussuhteessa kulkee laadunmittauskäytännöt, joilla koordinoidaan ohjelmistoprosessin suorittamistapojen ja prosessissa tuotettavan tuotteen laatua.

Ehkä tärkein koordinoiva prosessi on hallinnan prosessiryhmässä oleva projektinhallintaprosessi. Tutkimuksessa se esiintyi kaikista useimmin ohjelmistoprosessia koordinoivana prosessina.

7.1.2. Koordinointisuhde projektinhallintaprosessiin

Kuten luvussa 3.3.4.1 todettiin, projektinhallintaprosessin tavoitteena on tunnistaa, muodostaa, koordinoida ja valvoa projektille välttämättömiä toimenpiteitä, tehtäviä ja resursseja, jotta se voi toteuttaa vaatimuksia vastaavan tuotteen tai palvelun (ISO 15504-5). Vuorovaikutus ohjelmistoprosessin ja projektinhallintaprosessin välillä on erilaista kuin muiden prosessien kanssa. Riippuvuudet tutkitun ohjelmistoprosessin ja projektinhallintaprosessin välillä olivat sekä edeltävyysrajoitteisia että samanaikaisuuksien ja jaettujen resurssien hallintaa. Tämä on tyypillistä projektinhallintaprosessille sen tehtävien perusteella. Koordinointikeinoina käytettiin tehtävien aikataulutusta ja järjestykseen asettamista ja seuranta.

Lisäksi, projektinhallintaprosessi tarjoaa ohjaustietoa ohjelmistoprosessille hyvin tiheästi ja vuorovaikutuksen laadulle on erityisen leimaavaa sen tiivis kaksisuuntaisuus. Ohjelmistoprosessin tuottamat tulokset ja palautteet välitetään projektinhallintaprosessille, jotta sen tarjoamat ohjaustiedot olisivat mahdollisimman tarkkoja ja ajantasalla. Muiden prosessien vuorovaikutussuhteissa koordinoinnin kaksisuuntaisuus on tutkimuksen mukaan vähäisempää.

Vuorovaikutus projektinhallintaprosessin ja ohjelmistoprosessin välillä on tiiveintä ohjelmistoprosessin analyysi- ja suunnitteluvaiheissa. Kuitenkin vuorovaikutuksen koordinoiva vaikutus on erityisen kriittistä toteutusvaiheessa. Toteutusvaiheeseen kuuluvia tehtäviä saatetaan kiirehtiä, toistaa tai niiden suoritustapaa muuttaa projektinhallintaprosessin tarjoaman tiedon perusteella. Vuorovaikutussuhteessa kulkee projektisuunnitelma ja muuta aikatauluihin ja resursseihin liittyvää tietoa ja niiden muutoksia. Vuorovaikutussuhteen voimakas kaksisuuntaisuus edistää molempien prosessien tavoitteiden toteutumista.

7.2. Johtopäätökset

Tutkimuksen perusteella voidaan olettaa, että tehokkaan vuorovaikutuksen ansiosta myös muissa prosesseissa tiedetään, miten ohjelmistoprosessi etenee ja mitä sen hyväksi tulee tehdä ongelmien välttämiseksi. Tällöin ohjelmistoprosessin etenemisestä tulee jouheaa ja se luo pohjaa entistä paremmille vuorovaikutussuhteille. Edelleen tehokas koordinoiva vuorovaikutus auttaa prosessissa työskenteleviä ymmärtämään toistensa työtä. Tehtyjä oletuksia tukevat Humphrey ja Kellner (1989), joiden mukaan tehokkaan vuorovaikutuksen myötä työntekijöille selviää, mitä he voivat odottaa toisiltaan ja mitä muissa prosesseissa työskentelevät odottavat heiltä. Humphrey ja Kellner tiivistävät tämän osuvasti sanoessaan tehokkaan koordinoinnin selkeyttävän työn tekemisen mallia (ks. myös Curtis ym. 1992).

Tutkitun ohjelmistoprosessin vuorovaikutusta tehostamalla voidaan enemmän sopia sellaisista tehtävistä, joita ei ole määritelty kuulumaan mihinkään prosessiin, mutta jotka silti tulee tehdä työn etenemiseksi. Tutkimuksen perusteella tutkitun ohjelmistoprosessin ja projektinhallintaprosessin välillä saattaa olla tällaisia epäselvyyksiä, esimerkiksi tehtävien jako tekijöille ja niiden valmistumisesta tiedottaminen. Tätä havaintoa tukee Malonen ja Crowstonin tutkimustulokset (ks. Malone & Crowston 1990). Heidän mukaansa näitä epäselvyyksiä esiintyy useimmiten vastamallinnetuissa prosesseissa.

Tutkimuksesta voidaan myös vetää johtopäätös, että koordinoinnin tehostuessa ohjelmistoprosessi voi kehittyä nopeammin, jolloin turhat aikaa vievät sivupolut jäävät pois. Tämän totesivat myös Humphrey ja Kellner (1989) artikkelissaan. Erityisesti tutkitun ohjelmistoprosessin analyysivaiheessa tarvitaan monen prosessin koordinointia. Tiiviillä vuorovaikutuksella voidaan vähentää myöhemmän koordinoinnin tarvetta, mikä nopeuttaa ohjelmistoprosessia toteutusvaiheessa. Myöhemmissä vaiheissa tapahtuva koordinointi saattaa hidastaa ohjelmistoprosessin etenemistä. Tämä koskee ennen kaikkea asiakasvaatimusten selvittämisen ja asiakastuen tarjoamaa koordinointia.

Crowstonin (1997) tutkimuksen mukaan kahden prosessin erityisen tiivis vuorovaikutussuhde voi tarjota vaihtoehtoisen tavan tehtävien suorittamiseen. Tämä voi johtaa siihen, että organisaatiossa syntyy tarve määrittellä prosessit uudestaan, jotta niiden sisältämien tehtävien tekeminen olisi tehokkaampaa.

Ylipäätään tutkitulla ohjelmistoprosessilla on paljon koordinoivaa vuorovaikutusta muiden prosessien kanssa, mutta kohdeyrityksen kokoisessa yrityksessä määrä ei vielä ole sellainen ongelma, että vuorovaikutusta pitäisi alkaa rajoittamaan. Ainoastaan prosessin myöhäisissä vaiheissa, erityisesti toteutusvaiheessa, vuorovaikutusta alkaa kertyä niin paljon, että varsinkin tiedon välitystapaa tulisi miettiä. Vaiheeseen tulevat tiedot pitäisi toimittaa tiedon pyytämismekanismin mukaisesti johonkin yhteiseen paikkaan, josta tietoa tarvitseva voisi sen noutaa sopivana ajankohtana. Tällä tavoin välttäisiin kuormittamasta liikaa toteutusvaiheen toimijoita.

Galbraith mainitsi yhteisten tavoitteiden hyödyntämisen yhtenä tärkeimmistä koordinoitikeinoista (1973, 13). Kohdeyrityksessä tavoitteiden asettamista hyödynnetään ohjelmistoprosessin etappien avulla. Yrityksen ohjelmistoprosessi kulkee projektin sisällä ja nivoutuu yhteen projektinhallintaprosessin kanssa jokaisessa prosessin neljässä etapissa, sillä etappien tavoitteet ovat yhteiset.

7.3. Tutkimustulosten rajoitteita

Tutkimuksen lähestymistapana käytettiin tapaustutkimusta, mikä asettaa suurimmat rajoitteet tulosten yleistettävyydelle. Tutkimusongelmien asettelussa pyrittiin ottamaan huomioon tapaustutkimuksen asettamat rajoitteet, jolloin tutkimustulokset pätevät sellaisinaan vain tutkittuun organisaatioon. Tulosten yleistettävyyttä pyrittiin parantamaan tutkimuskohteen rajauksella.

Tutkimustulosten yleistettävyys kärsii erityisesti siitä, että kohdeorganisaation ohjelmistoprosessi poikkeaa ohjelmistoprosessin yleisestä mallista (ks. esim. Pressman 1997, 33-46), sillä kohdeorganisaatiossa tuotetaan sulautettuja järjestelmiä. Tämä

tarkoittaa, että laitteistontuotantoprosessi kulkee lähes yhteenkietoutuneena ohjelmistoprosessin rinnalla. Tämä vaikuttaa hyvin paljon ohjelmistoprosessin luonteeseen, sillä ohjelmiston vaatimukset saadaan pitkälti juuri laitteistontuotantoprosessista. Laitteistontuotantoprosessi pyrittiin erittelemään tutkimuksessa rajaamalla tutkittavaa ohjelmistoprosessia riittävästi. Rajausta painotettiin myös haastattelujen aikana. Haastattelujen tukeminen dokumenttien analysoinnilla pienentää mahdollisten virheiden todennäköisyyttä. Haastattelussa saadun aineiston vertaaminen dokumentteihin paljasti muutamia epäyhteneväisyyksiä, jotka tarkistettiin ja korjattiin.

Tutkimusmenetelmänä käytetty ryhmähaastattelu saattaa vaikuttaa saatuihin tutkimustuloksiin. Ensinnäkin, tutkijan kokemattomuus kysymysten asettelijana saattaa aiheuttaa, että haastateltavat pitävät haastattelua ajanhukkana, jolloin saadut vastaukset voivat olla epätäydellisiä. Toiseksi, tutkijan johdattelevat kysymykset toisaalta auttavat haastateltavia pysymään aiheessa ja katsomaan ohjelmistoprosessia uudesta näkökulmasta, mutta samalla johdattelu voi vääristää annettuja vastauksia tutkijan toivomaan suuntaan. Kolmas haastattelun ongelma on vastaajien mahdollinen halu mukautua kirjattuihin, virallisiin käytäntöihin. Viralliset käytännöt eivät kuitenkaan aina anna oikeaa kuvaa asioiden todellisesta kulusta ja luonteesta. Toisaalta, haastattelun tarkoitus ja kohderyhmä ja haastattelijan tausta ja asema organisaatiossa oli haastateltavien tiedossa, mikä varmasti vähensi pyrkimyksiä mukautua virallisiin käytäntöihin ja siten kohensi haastattelujen luotettavuutta. Neljänneksi, kaikki tutkimusta varten haastatellut olivat ohjelmistoprosessin toimijoita. Haastattelusta olisi voinut tulla erilaista aineistoa, jos samoja asioita olisi kysytty useammilta toimijoilta ja myös muiden prosessien, esimerkiksi laadunhallintaprosessin edustajilta.

On vaikea erottaa, mitkä tehtävät kuuluvat mihinkin prosessiin, sillä kyse on aina organisaation sisällä tehdyistä valinnoista, joille ei välttämättä löydy järkevää perustetta. Erityisesti projektinhallintaprosessin tehtäviä, ja tässä tutkimuksessa myös laitteistontuotantoprosessin tehtäviä, on vaikea erottaa ohjelmistoprosessin tehtävistä. Tätä tutkimusta varten prosessit rajattiin aikaisemmin tehtyjen prosessikuvausten mukaisesti. Sama ongelma prosessin rajaamisessa esiintyy myös Kaltion

tutkimusraportissa (Kaltio 1995). Toisaalta tutkimuskohteen rajaaminen on aina riski, sillä rajauksen ulkopuolelle saattaa jäädä jotain tutkimusongelmien kannalta hyvinkin olennaista. Tässä tutkimuksessa riskinotto on ollut tietoista ja sen vaarat on tunnistettu.

7.4. Yhteenveto

Ohjelmistoprosessin koordinointi on riippuvuuksien hallintaa. Riippuvuuksia syntyy, sillä ohjelmistoprosessin tekemiseen osallistuu tekijöitä muistakin prosesseista. Ohjelmistoprosessia ja sen etenemistä koordinoivat ne prosessit, jotka toimittavat ohjelmistoprosessille kriittisiä tietoja omilta alueiltaan. Tutkimuksen mukaan koordinointiin osallistuu useita prosesseja eri SPICE:n prosessiryhmistä. Tutkimustulos tukee Kaltion (1995) havaintoja. Merkittävin vaikutus tutkitun ohjelmistoprosessin etenemiseen on projektinhallintaprosessilla ja asiakasvaatimusten selvittämisellä. Kohdeorganisaation prosessimäärittelyjen uutuus ja valittu tutkimusmenetelmä selittää osaltaan tuen prosessiryhmän vuorovaikutussuhteiden puuttumisen.

Koordinoinnin vaikutus tutkitun ohjelmistoprosessin etenemiseen on suurinta ohjelmistoprosessin alkuvaiheissa, eli analyysi- ja suunnitteluvaiheissa. Erot eri vaiheiden koordinoinnin tiheydessä ja määrässä ovat kuitenkin lähinnä vain viitteellisiä, sillä sekä analyysi-, suunnittelu-, että toteutusvaiheessa koordinoivia prosesseja oli kussakin kolme ja testausvaiheessa kaksi. Analyysi- ja suunnitteluvaiheissa koordinointi ja vuorovaikutus oli tutkimuksen mukaan kuitenkin tiiveintä ja vaikutti eniten ohjelmistoprosessin etenemiseen. Ohjelmistoprosessin suoritusta voidaan tehostaa panostamalla varsinkin sen alkuvaiheiden koordinointiin, näin voidaan nopeuttaa ennen kaikkea toteutusvaihetta. Tutkimuksen mukaan asiakasvaatimusten selvittäjien tulisi työskennellä yhteistyössä ohjelmistosuunnittelijoiden kanssa läpi analyysi- ja suunnitteluvaiheiden, jotta asiakasvaatimukset välittyisivät ohjelmistoprosessiin nopeasti ja oikeansisältöisinä. Havainto tukee Pressmanin (1997, 286-289) havaintoa kommunikoinnin tärkeydestä ja ongelmallisuudesta juuri ohjelmistoprosessin aloitusvaiheissa.

Vuorovaikutuksen tallentamisesta tulee huolehtia, jotta tieto toimintaa koordinoivasta tiedosta saavuttaisi mahdollisimman monia ohjelmistoprosessin toimijoita ja jotta sitä osattaisiin hyödyntää prosessin myöhemmissä vaiheissa. Tutkimuksen perusteella työpisteiden fyysinen läheisyys auttaa yhteistyön, epämuodollisen kommunikaation ja siten myös koordinoivan vuorovaikutuksen onnistumista. Tätä havaintoa tukee Krautin ja Streeterin (1995) tutkimuksen tulos epämuodollisen tiedonvaihdon tärkeydestä vuorovaikutuksen menetelmänä.

Laitteistontuotantoprosessi on kriittinen sulautettujen järjestelmien ohjelmistokehityksessä. Sen olemassaolo yksinkertaistaa tutkimuskohteena olevaa ohjelmistoprosessia, sillä ohjelmiston vaatimukset määrittyvät pitkälti laitteistontuotantoprosessin sanelemana. Pelkkiä ohjelmistoja tuottavalle yritykselle laitteistontuotantoprosessia vastaa tiedot asiakkaan omista laitteistoista, jolloin asiakasvaatimusten selvittäminen prosessi tarjoaa koordinoitua enemmän kuin mitä tämän tutkimuksen tuloksena saatiin. Tällöin ohjelmistoprosessin analyysi- ja suunnitteluvaiheet mutkistuvat huomattavasti. Kummassakin tapauksessa on tärkeää, että ohjelmisto noudattaa sille asetettuja vaatimuksia, olivat ne sitten laitteiston tai asiakkaan asettamia.

Lopuksi tulee muistaa, ettei kaikki hyvä tapahdu yhdessä yössä. Tehostuneen vuorovaikutuksen ja tehostuneen ohjelmistoprosessin hyödyt tulevat esille vasta pitkällä tähtäimellä. Esimerkiksi yritysjohdon tuki on tärkeää prosesseja koordinoivan vuorovaikutuksen onnistumiselle ja jatkuvuudelle.

Tämä tutkimus oli yksi yritys huomioida ohjelmistoprosessin yhteydet muihin yrityksen liiketoimintaprosesseihin. Laadukkaiden ohjelmistojen tuottamiseksi on tärkeää, että ohjelmistoprosessin rajapintaa tutkittaisiin enemmänkin. Curtisin ym. (1992) mukaan ohjelmistoyrityksissä ollaan huomattu, että ennen kuin otetaan käyttöön uusia menetelmiä tai toimintatapoja, pitää huolehtia nykyisten menetelmien ja välineiden tuki suunnitelluille käytännöille. Koordinoivien vuorovaikutussuhteiden tutkimus voi vastata kysymyksiin nykyisten käytäntöjen heikkouksista ja vahvuuksista. Jatkossa olisikin aiheellista tarkastella niin ohjelmistoprosessin kuin muidenkin liiketoimintaprosessien

keskinäisiä yhteyksiä yksityiskohtaisemmin esimerkiksi koordinoititapojen kautta, kuten Crowston tutkimuksessaan (1997) esittää. Malonen ja Crowstonin mukaan koordinoinnin mekanismit selvittämällä voidaan riippuvuuksien hallintaa tehostaa vaihtoehtoisilla koordinoititavoilla. (Malone & Crowston 1994) Seuraavassa luvussa tehdään yhteenveto tutkielmassa käsitellyistä asioista, tuloksista ja tärkeimmistä johtopäätöksistä.

8. YHTEENVETO

Tässä tutkimuksessa on tarkasteltu ohjelmistoprosessia koordinoivaa vuorovaikutusta ja sen merkitystä ohjelmistoprosessin etenemiselle. Aihetta käsiteltiin prosessimuotoisen työtavan käsittein. Aluksi tutustuttiin erilaisiin liiketoiminnan organisoinnin tapoihin ja määriteltiin liiketoimintaprosessin ja prosessin käsitteet. Kolmannessa luvussa käsiteltiin ohjelmistotuotantoa ja ohjelmistoprosesseja, neljännessä käytiin läpi prosessimuotoista työtapaa. Viidennessä luvussa käsiteltiin koordinoititeorian pääasiat. Tutkimusmenetelmät ja empiirisen tutkimuksen kulku selostettiin luvussa kuusi. Seitsemännessä luvussa tutkimuksen tulokset raportoitiin ja niitä analysoitiin SPICEn avulla.

Tutkimuksen mukaan ohjelmistoprosessia koordinoivalla vuorovaikutuksella on suuri merkitys ohjelmistoprosessin etenemiseen. Ensimmäisenä ongelmana tutkimuksessa oli selvittää, *millä prosesseilla tutkittavaa ohjelmistoprosessia koordinoidaan*. Ongelmaan vastanneita tuloksia käsiteltiin kohdassa 7.1.1, jossa koordinoivat prosessit jaoteltiin SPICEn prosessiryhmien mukaisesti. Tutkimuksen mukaan ohjelmistoprosessia koordinoivia prosesseja olivat projektinhallintaprosessi, asiakasvaatimusten selvittämisprosessi, laadunhallintaprosessi ja asiakastuen prosessi. Koordinointi oli tiiveintä projektinhallintaprosessin kanssa. Tiiviyden lisäksi vuorovaikutussuhteelle erityistä oli sen kaksisuuntaisuus, joka edistää molempien prosessien tavoitteiden toteutumista. Ohjelmistoprosessin koordinoitisuhdetta projektinhallintaprosessiin käsiteltiin enemmän kohdassa 7.1.2.

Toisena ongelmana oli miettiä, *missä ohjelmistoprosessin kohdissa koordinointi vaikuttaa eniten tutkittavan ohjelmistoprosessin kulkuun*. Ongelmaan vastaamista varten tutkittavana ollut ohjelmistoprosessi rajattiin ja luettavuuden säilyttämiseksi sen vaiheet nimettiin uudelleen. Tutkimuksen mukaan eniten koordinoitua kaikkien prosessien kanssa oli ohjelmistoprosessin analyysi- ja suunnitteluvaiheissa, vaikkakin koordinoitua esiintyi kohtuullisen tasaisesti kaikissa tutkimuskohteena olleen ohjelmistoprosessin vaiheissa.

Kolmantena ongelmana oli selvittää, *miten koordinointiin panostaminen näissä kohdissa voisi tehostaa ohjelmistoprosessia yleisesti*. Ongelmaa käsiteltiin ohjelmistoprosessin vaiheiden ja niissä käytettyjen koordinointimenetelmien analyysillä. Tutkimustuloksena saatiin, että tehostamalla ohjelmistoprosessin analyysi- ja suunnitteluvaiheiden koordinointia ohjelmistoprosessin myöhempiä vaiheita voitaisiin jouduttaa. Tämä tehostaisi samalla koko ohjelmistoprosessia, mutta edellyttää sopivien vuorovaikutuskeinojen miettimistä ja niiden tilanteenmukaista käyttöä.

Tutkimuksessa korostui erityisesti prosessin rajauksen merkitys. Rajaus on aina riski, mutta se on silti tehtävä. Tässä tapauksessa laitteistontuotantoprosessi on rajattu tutkimusaineiston ulkopuolelle. Laitteistontuotantoprosessin vaikutusta ohjelmistoprosessiin ei kuitenkaan voi väheksyä, sillä sulautettujen järjestelmien ohjelmistokehityksessä laitteisto asettaa ohjelmiston vaatimuksista suuren osan.

Tutkimustulosten perusteella ohjelmistoprosessin yhteyksistä muihin prosesseihin saatiin kokonaisvaltaisempi ja selvempi kuva. Koordinoivien prosessien huomioiminen niin ohjelmistoprosessin kuin muidenkin yrityksen liiketoimintaprosessien myöhemmissä kehittämissä yrityksissä auttaa pääsemään totuudenmukaisempiin prosessikuvauksiin ja sitä myötä tehokkaampaan ohjelmistoprosessiin.

Jatkossa olisi hyödyllistä tutkia tietotekniikan mahdollistamia muutoksia prosessien välisessä koordinoivassa vuorovaikutuksessa. Toffolonin ja Dakhlin mukaan (2000) paras mahdollinen toiminnan koordinointi edellyttää koordinoivien vuorovaikutussuhteiden automatisointia ainakin osittain, joten tämän tutkielman jatkoksi sopisi hyvin tutkimus siitä, miten tietotekniikka muuttaa tai tehostaa ohjelmistoprosessia koordinoivaa vuorovaikutusta. Esimerkiksi työnkulku- ohjelmiston käyttöönotto voi muuttaa koordinoinnin luonnetta suuresti.

LÄHTEET

Andrews, P. H & Herschel R. T. 1996. Organizational communication. Empowerment in a technological society. Houghton Mifflin Company, USA.

Armistead, C., Harrison, A. & Rowlands, P. 1995. Business process re-engineering: lessons from operations management. International Journal of Operations & Production Management, Vol.15, No. 12, 46-58.

Biazzo, S. 2000. Approaches to Business Process Analysis: A Review. Business Process Management Journal, Vol. 6, No. 2, 99-112.

Boehm, B. 1988. A Spiral Model of Software development and Enhancement. Computer, Vol. 21, No.5, 61-72.

Brooks, F. P. Jr. 1975. The mythical man-month. Essays on Software Engineering. Addison-Wesley publishing company, USA.

Burns, T. & Stalker, G. M. 1961. The Management of Innovation. Tavistock Publications, Englanti.

Chroust, G. 2000. Workshop: Software Process and Product Improvement. Proceedings of the 26th EUROMICRO conference (EUROMICRO'00), Maastricht, Netherlands, 2000. Saatavilla myös www-muodossa: <<http://www.computer.org/proceedings/euromicro/0780/07800011.pdf>>. Luettu 2.7.2001.

Cook, J. E. & Wolf, A. L. 1999. Software Process Validation: Quantitatively Measuring the correspondence of a Process to a Model. ACM Transactions on Software Engineering and Methodology, Vol. 8, No. 2, 147-176.

Crowston, K. 1994. A Taxonomy Of Organizational Dependencies and Coordination Mechanisms. MIT Center for Coordination Science, Working Paper Series. Saatavilla [www-muodossa: <http://ccs.mit.edu/papers/CCSWP174.html>](http://ccs.mit.edu/papers/CCSWP174.html). Luettu 2.4.2001.

Crowston, K. 1997. A Coordination theory approach to organizational process design. *Organization Science*, Vol. 8, No. 2, 157-175.

Curtis, B., Kellner, M. I. & Over, J. 1992. Process Modeling. *Communications of the ACM*, Vol. 35, No. 9, 75-90.

Curtis, B., Krasner, H. & Iscoe, N. 1988. A field study of the software design process for large systems. *Communications of the ACM*, Vol.31, No. 11, 1268-1287.

Davenport, T. 1993. Process innovation. Reengineering work trough information technology. Ernst & Young, USA.

Davenport, T. & Short, J. 1990. The New Industrial Engineering: Information Technology and Business Process Redesign. *Sloan Management Review*, Vol. 31, No. 4, 11-27.

Frank, A. D. & Brownell, J. L. 1989. Organizational communication and behaviour: communicating to improve performance. Holt, Rinehart and Winston Inc., USA.

Galbraith, J. R 1973. Designing complex organizations. Addison- Wesley Publishing Company, USA.

Galbraith, J. R & Nathanson, D. A. 1978. Strategy Implementation: The Role of Structure and Process. West Publishing, USA.

Gilb, T. 1988. Principles of software engineering management. Addison- Wesley, Iso-Britannia.

- Haikala, I. & Märijärvi, J. 1998. Ohjelmistotuotanto. Suomen Atk-kustannus, Jyväskylä.
- Hakulinen, E., Halttunen, V., Taskinen, S-K. & Ahonen, J. 1999. Pisko-projektin raportti: Sovellettu ohjelmistotuotantoprosessien kuvausmenetelmä seinätekniikkaa hyödyntäen. Jyväskylän yliopisto, Tietotekniikan Tutkimusinstituutti, Jyväskylä 1999.
- Hammer, M. & Champy, J. 1993. Reengineering the corporation. A manifesto for business revolution. HarperBusiness, USA.
- Hammer, M. & Stanton S. A. 1994. The reengineering revolution. A handbook. Harper Business, USA.
- Harrison, A. 1998. Investigating business processes: does process simplification always work? Business Process Management Journal, Vol. 4, No. 2, 137-153.
- Humphrey, W.S. 1988. The Software Engineering Process: Definition and Scope. Proceedings of the 4th international software process workshop on Representing and enacting the software process, Devon, UK, 82-83. ACM Press, USA. Saatavilla myös www-muodossa: <<http://www.acm.org/pubs/articles/proceedings/soft/75110/p82-humphrey/p82-humphrey.pdf>>. Luettu 27.6.2001.
- Humphrey, W. S. 1989. Managing the Software Process. Addison-Wesley Publishing Company, USA.
- Humphrey, W. S. & Kellner, M. I. 1989. Software Process Modeling: Principles of Entity Process Models. Proceedings of the 11th international conference on Software engineering, Pittsburgh, USA, 331–342. IEEE, USA. Saatavilla myös www-muodossa: <<http://www.acm.org/pubs/articles/proceedings/soft/74587/p331-humphrey/p331-humphrey.pdf>>. Luettu 29.6.2001.
- ISO 15504-5 standardi 1998. SPICE –malli ohjelmistotuotannon prosesseista. ISO/IEC TR 15504-5.

Jokela, J. 1999. Prosessimittariston käyttöönottokokemuksia ICL Datassa. *Systemityö* 1/99, 28-29.

Kahen, G. & Lehman, M. M. 2000. [online] A Brief Rview of Feedback Dimensions in the Global Software Process. Position paper in FEAST 2000 International Workshop on Feedback and Evolution in Software and Business Processes, London, UK, 2000. Saatavilla myös www- muodossa: <<http://www.doc.ic.ac.uk/~mml/feast/papers/pdf/658.pdf>>. Luettu 26.6.2001.

Kaltio, T. 1995. Software configuration management process improvement – use of process modeling for describing interactions to other processes. Teoksessa Lassenius, C. & Alho, K. (toim.) *Software Process Modeling: Research and Applications*. Teknillinen korkeakoulu, Helsinki.

Karvonen, O. 1999. *Prosessimalli. Käsikirja prosessien kehittäjälle*. Helsingin kaupunginkanslian julkaisusarjaa A8/1999. Helsingin kaupunginkanslia, Helsinki.

Kock, N. F. & McQueen, R. J. 1996. Product flow, breadth and complexity of business processes. An empirical study of 15 business processes in three organizations. *Business Process Re-engineering and Management Journal*, Vol. 2, No. 2, 8-22.

Kraut, R. E. & Streeter, L. A. 1995. Coordination in Software Development. *Communications of the ACM*, Vol. 38, No. 3, 69-81.

Kreps, G. 1989. *Organizational communication: theory and practice*. 2. painos. Longman, USA.

Laamanen, K. & Tinnilä, M. 1998. *Prosessijohtamisen käsitteet*. 2. painos. Metalliteollisuuden kustannus Oy, Vantaa.

Lim, W. C. 1998. *Managing Software Reuse*. Prentice Hall, Inc. USA.

Lee, R. G. & Dale, B. G. 1998. Business process management: a review and evaluation. *Business Process Management Journal*, Vol. 4, No. 3, 214-225.

Lehman, M. M. 1989. The role of process models in software and systems development and evolution. Proceedings of the 5th international software process workshop on experience with software process models. Kennebunkport, USA, 91-94. IEEE Computer Society Press Los Alamitos, USA. Saatavilla myös [www –muodossa: <http://www.acm.org/pubs/citations/proceedings/soft/317498/p91-lehman/>](http://www.acm.org/pubs/citations/proceedings/soft/317498/p91-lehman/). Luettu 26.6.2001.

Lyytinen, K. 1988. Expectation Failure Concept and Systems Analyst's view of Information System Failures: Results of an Exploratory Study. *Information and Management*, Vol. 14, No. 4, 45-56.

Miles, M. B. & Huberman, M. A. 1994. *Qualitative Data Analysis. An Expanded Sourcebook*. 2. painos. Sage Publications, USA.

March, J. G. & Simon, H. A. 1958. *Organizations*. 2. painos (1959). John Wiley & Sons, USA.

Malone, T. & Crowston, K. 1990. What is Coordination Theory and How Can It Help Design cooperative Systems? Proceedings of the conference on Computer-supported cooperative work, Los Angeles, USA, 1990. ACM, USA. Saatavilla myös [www-muodossa: <http://www.acm.org/pubs/articles/proceedings/cscw/99332/p357-malone/p357-malone.pdf>](http://www.acm.org/pubs/articles/proceedings/cscw/99332/p357-malone/p357-malone.pdf). Luettu 11.4.2001.

Malone, T. & Crowston, K. 1994. The Interdisciplinary Study of Coordination. *ACM Computing Surveys*, Vol. 26, No. 1, 87-119.

Malone, T., Crowston, K., Lee, J., Pentland, B., Dellacoras C., Wyner, G., Quimby, J., Osborn, C. S., Bernstein, A., Herman, G., Klein, M. & O'Donnell E. 1999. Tools for

Inventing Organizations: Toward A Handbook of Organizational Processes. Management Science, Vol. 45, No. 3, 425-443.

Mangham, I. L. 1978. Interactions and interventions in organizations. John Wiley & Sons, Iso-Britannia.

McCann, J. & Galbraith, J. R. 1981. Interdepartmental relations. Teoksessa Nyström, P & Starbuck, W. (toim.) Handbook of organizational design, Volume 2, Oxford University Press, Oxford.

Medina-Mora, R., Winograd, T., Flores, R. & Flores, F. 1992. The Action Workflow Approach to Workflow Management Technology. Conference proceedings on Computer-supported cooperative work, Toronto, Canada, 1992, 281–288. ACM Press, New York. Saatavilla myös www-muodossa: <<http://www.acm.org/pubs/articles/proceedings/cscw/143457/p281-medina-mora/p281-medina-mora.pdf>>. Luettu 11.4.2001.

Nevalainen, R. 1999. Kokemuksia ohjelmistoprosessien arvioinnista SPICE:n avulla. Systemityö 1/99, 2-8.

Notkin, D. 1988. The Relationship between Software Development Environments and the Software Process. Teoksessa Henderson, P. (toim.) Proceedings of the ACM SIGSOFT/SIGPLAN Software engineering symposium on practical software development environments, Boston, USA, 1988, 107-109. ACM Press, USA. Saatavilla myös www- muodossa: <<http://www.acm.org/pubs/articles/proceedings/soft/64135/p107-notkin/p107-notkin.pdf>>. Luettu 26.6.2001.

Osterweil, L. J. 1997. Software Processes Are Software Too, Revisited: An Invited Talk on the Most Influential Paper of ICSE '97. Proceedings of the 19th International Conference on Software Engineering, Boston, USA, 1997, 540-548. Saatavilla post script –muodossa osoitteesta <<http://thunderbird.cs.umass.edu/publications.asp>>. Luettu 26.6.2001.

Pandya, V. K, Karlsson, A., Sega, S. & Carrie, A. 1997. Towards the manufacturing enterprises of the future. *International journal of Operations & Production Management*, Vol. 17, No. 5, 502-521.

Phan, D. D., Vogel D. R. & Nunamaker, J. F. Jr. 1995. Empirical studies in software development projects: Field survey and OS/400 study. *Information and Management* Vol. 28, No. 14, 271-280.

Pressman, R. S. 1997. *Software Engineering. A Practitioner's Approach*. Neljäs painos, eurooppalainen sovitus. McGraw-Hill Companies, USA.

Pritchard, J-P. & Armistead, C. 1999. Business process management – lessons from European business. *Business Process Management Journal*, Vol. 5, No. 1, 10-32.

Pylkkö, J. 1999. Ohjelmistotyön laadun mittaaminen. *Systeemyö* 1/99, 12-13.

Reel, J. S. 1999. Critical success Factors in Software Projects. *IEEE Software*, Vol. 16, No. 3, 18-23.

Riddle, W. E. 1987. Improving the Software process. *Proceedings of the 9th International Conference on software engineering*, California, USA, 1987, 344-344. IEEE Computer Society Press, USA. Tiivistelmä saatavilla www-muodossa: <<http://www.acm.org/pubs/articles/proceedings/soft/41765/p344-riddle/p344-riddle.pdf>> Luettu 26.6.2001.

Riihinen, J., Rikkilä, J. & Lassenius, C. 1995. The entire approach to the ISPW-6 software process example problem. Teoksessa Lassenius, C. & Alho, K. (toim.) *Software Process Modeling: Research and Applications*. Helsingin teknillinen korkeakoulu, Otaniemi.

Rogers E. M. & Agarwala-Rogers R. 1976. *Communication in organizations*. The Free Press, USA.

Rotemberg, J. J. 1999. Process- versus Function-Based Hierarchies. Journal of Economics and Management Strategy, Vol. 8, No. 4, 453-487.

Saukkonen, S. & Oivo, M. 1998. Teollinen ohjelmistoprosessi. Ohjelmistoprosessin parantaminen SIPI –menetelmällä. Teknologia katsaus 64/98. Teknologian kehittämiskeskus, Sipoo.

Sommerville, I. 1995. Software Engineering. Viides painos. Addison-Wesley Publishing Company, USA.

Thompson, J. D. 1974. Miten organisaatiot toimivat. Ekonomia –sarja 28. Weiling & Göös, Tapiola.

Toffolon, C. & Dakhli, S. 2000. A Framework for Studying the Coordination Process in software Engineering. Proceedings of the 2000 ACM symposium on Applied computing, Como, Italy, 2000, Vol. 2, 851 - 857. Saatavilla [www-muodossa: <http://www.acm.org/pubs/articles/proceedings/sac/338407/p851-toffolon/p851-toffolon.pdf >](http://www.acm.org/pubs/articles/proceedings/sac/338407/p851-toffolon/p851-toffolon.pdf) Luettu 2.4.2001.

Yin, R. K. 1994. Case study research. Design and methods. 2. painos. Sage Publications, USA.