

Perttu Lamminmäki

**OHJELMISTOTUOTANTOPROSESSI PIENISSÄ  
OHJELMISTOALAN YRITYKSISSÄ**

Tietojärjestelmätieteen  
kandidaatintutkielma  
11.4.2008

Jyväskylän yliopisto  
Tietojenkäsittelytieteiden laitos  
Jyväskylä

# TIIVISTELMÄ

Lamminmäki, Perttu

Tietojärjestelmätieteen kandidaatintutkielma / Perttu Lamminmäki

Jyväskylä: Jyväskylän yliopisto, 2008, 45s.

Kandidaatintutkielma

Pienet ohjelmistoalan yritykset ovat viime vuosina muodostuneet tärkeäksi suurtenkin ohjelmistojen kehittäjäryhmäksi. Hyvin suuri joukko ohjelmistoja kehittävästä yrityksistä on nykyään alle 50 työntekijän kokoisia. Pienten yritysten ohjelmistotuotantoa, sen erityispiirteitä ja pienille yrityksille sopivia ohjelmistotuotantoprosesseja on kuitenkin alettu vasta viime vuosina tutkimaan enemmän.

Tässä tutkielmassa luodaan katsaus pienten ohjelmistoalan yritysten ohjelmistotuotantoprosessiin ja sen erityispiirteisiin kirjallisuuden pohjalta. Tutkielmassa kartoitetaan kirjallisuudesta myös miten pienet yritykset voisivat ohjelmistotuotantoprosessiaan kehittää ja erilaisten olemassaolevien ohjelmistotuotannon menetelmien ja prosessien sopivuutta pienille yrityksille. Tutkimusmenetelmänä on käsitteellisteoreettinen tutkimus, jossa pyritään löytämään kirjallisuudesta näkökulmaa pienten yritysten ohjelmistotuotantoprosessiin ja sen kehittämisen aloittamiseen.

Pienille yrityksille suunnattuja ohjelmistotuotannon menetelmiä ja prosesseja on kehitetty lisää viime vuosina. Pienet yritykset ovat kuitenkin monesti erikoislaatuisia ja niiden erityispiirteet on otettava huomioon erilaisia malleja ja menetelmiä sovellettaessa. Pienen yrityksen olemassaolevien käytäntöjen ja prosessien selvittäminen esimerkiksi mallintamalla ja arvioimalla on tärkeää ennen yrityksen prosessin kehittämisen aloittamista. Usein käytettyjä menetelmiä, malleja ja prosesseja täytyy räätälöidä tapauskohtaisesti, jotta ne sopivat parhaalla mahdollisella tavalla kullekin pienelle yritykselle.

AVAINSANAT: Ohjelmistotuotantoprosessi, pieni ohjelmistoalan yritys, arviointimenetelmä, ohjelmistotuotantoprosessin kehittäminen

Ohjaaja: etunimi sukunimi  
Tietojenkäsittelytieteiden laitos  
Jyväskylän Yliopisto

Tarkastaja: etunimi sukunimi  
Tietojenkäsittelytieteiden laitos  
Jyväskylän Yliopisto

# SISÄLLYSLUETTELO

1 JOHDANTO .....	5
2 PIENTEN OHJELMISTOALAN YRITYSTEN OMINAISPIIRTEET .....	8
2.1 Pienten yritysten ohjelmistotuotantoprosessin ominaispiirteet .....	11
2.2 Yrityksen koon vaikutus ohjelmistotuotantoprosessiin ja sen kehittämiseen .....	18
3 SUURILLE YRITYKSILLE KEHITETTYJEN OHJELMISTOTUOTANTOPROSESSIEN JA MENETELMIEN SOVELTUVUUS PIENILLE OHJELMISTOALAN YRITYKSILLE .....	21
3.1 Nykyisten arviointimenetelmien soveltuvuus pienille yrityksille .....	22
3.2 Arviointimenetelmien räätälöinti pienten yritysten tarpeisiin .....	27
4 PIENTEN OHJELMISTOALAN YRITYSTEN OHJELMISTOTUOTANTOPROSESSIN KEHITTÄMINEN .....	30
4.1 Ohjelmistotuotantoprosessin mallintaminen .....	34
4.2 Pienten yritysten ohjelmistotuotantoprosessin kehittämisen arviointia ....	36
5 YHTEENVETO.....	40
LÄHDELUETTELO .....	42

## 1 JOHDANTO

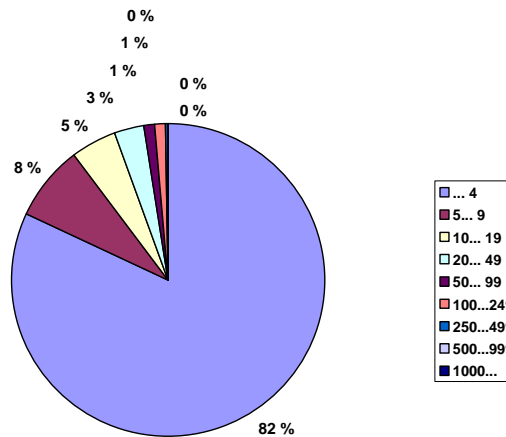
Richardson ja Gresse von Wangenheim (2007) määrittelevät pienen ohjelmistoalan yrityksen alle viidenkymmenen hengen itsenäisesti rahoitetuksi ja organisoiduksi yritykseksi ja heidän mukaansa tällaiset yritykset ovat tärkeitä monien eri kansantalouksien kasvulle. He myös toteavat, että Yhdysvalloissa, Brasiliassa, Kanadassa, Kiinassa, Intiassa, Suomessa, Irlannissa, Unkarissa ja monissa muissa maissa kaikista ohjelmistoalan yrityksistä pienten yritysten osuus voi olla aina 85 prosenttia. Fayad, Laitinen ja Ward (2000) määrittelevät pienen ohjelmistoalan yrityksen sellaiseksi yritykseksi, jossa on viisikymmentä työntekijää tai vähemmän ja heidän mukaansa suurin osa Yhdysvaltojen ohjelmistoalan yrityksistä sopii tähän määritelmään. Myös Horvat, Györkös ja Rozman (2000) tarkoittavat pienellä itsenäisellä yrityksellä yrityksiä, joiden koko vaihtelee tavallisesti viiden ja viidenkymmenen työntekijän välillä. Tilastokeskus (2007) määrittelee tietojenkäsittelypalvelutoimialan seuraavasti: "Automaattiseen tietojenkäsittelyyn liittyvä, asiakkaan laskuun tapahtuva laitteisto- ja ohjelmistokonsultointi, ohjelmistojen suunnittelu, valmistus ja julkaiseminen, tietokone- ja käsittelypalvelutoiminta, tietokantaisännöinti, konttori- ja tietokoneiden korjaus ja huolto sekä muu tietojenkäsittelypalvelu". Tässä tutkimuksessa pieni ohjelmistoalan yritys määritellään sellaiseksi tietojenkäsittelypalvelutoimialan yritykseksi, jossa työskentelee alle viisikymmentä työntekijää.

On kuitenkin huomattava, että pienellä ohjelmistoalan yrityksellä ei välttämättä tarkoiteta juuri perustettua niin kutsuttua start-up yritystä, jolla on myös juuri aloitetulle liiketoiminnalle ominaisia haasteita ja ongelmia. Sutton (2000) toteaaakin, että pienellä jo toimintansa vakiinnuttaneella yrityksellä on erilaiset toimintaedellytykset ja toiminnan vakiintumisesta tulevat edut puolellaan, kun sitä verrataan start-up yritykseen. Suttonin (2000) mukaan tällaisia etuja ovat sisäisen kommunikoinnin ja koordinoinnin parempi toimiminen, suurempi joustavuus ja reagointikyky jo vakiintuneen organisaation ansiosta, valmis perusta tuotteista, kumppaneista ja asiakkaista ja mahdollisesti työntekijöiden pidempi yhteinen historia ja visio. Suttonin (2000) mukaan start-up yrityksillä ja pienillä yrityksillä on joitakin yhteisiä ongelmia, mutta ne kohtaavat erilaisia haasteita erilaisessa toimintaympäristössä, joten pienille yrityksille sopivia malleja ei välttämättä voida suoraan käyttää start-up yrityksissä, vaikka myös start-up yrityksen koko olisi pieni.

Tilastokeskuksen (2007) mukaan Suomessa oli vuonna 2005 yhteensä 4602 tietojenkäsittelypalvelutoimialan yritystä. Tilastojen mukaan (Tilastokeskus 2007) tämän toimialan yrityksistä Suomessa n. 98 prosenttia (KUVIO 1) on kooltaan alle viidenkymmenen hengen yrityksiä. Lähes kaikki Suomessa toimivat tietojenkäsittelypalvelutoimialan yritykset kuuluvat siis tässä tutkimuksessa tarkoitettuihin pieniin ohjelmistoalan yrityksiin, joiden koko on

alle viisikymmentä työntekijää. Myös Suomessa on siis hyödyllistä tutkia pienten ohjelmistoalan yritysten erityispiirteitä, niiden ohjelmistotuotantoprosessin kehittämistä ja arvioimista ja olemassa olevien menetelmien ja prosessien soveltuvuutta pienille ohjelmistoalan yrityksille.

### Tietojenkäsittelypalvelualan yritykset koon mukaan Suomessa vuonna 2005 (Tilastokeskus 2007)



KUVIO 1. Tietojenkäsittelypalvelu toimialan yritykset koon mukaan Suomessa vuonna 2005 (Tilastokeskuksen 2007 tilastojen mukaan).

Ohjelmistoja tuottavien yritysten, jotka haluavat menestyä markkinoilla, pitäisi tuottaa korkealaatuisia ohjelmistoja yrityksen koosta riippumatta. Toisin sanoen jopa hyvin pienen yrityksen, joka kehittää pieniä sovelluksia pienissä ohjelmistoprojekteissa ja vähillä resursseilla, tulisi pystyä varmistamaan tuottamiensa ohjelmistojen ja palvelujen korkea laatu. Suurissa yrityksissä ohjelmistotuotantoprosessimalleja on sovellettu jo pitkään, mutta vasta viime vuosina on tiedostettu ohjelmistotuotantoprosessin kehittämisen tärkeys ja hyöty myös pienissä yrityksissä. Richardsonin ja Gresse von Wangenheimin (2007) mukaan suuret ja pienet ohjelmistoalan yritykset kohtaavat samanlaisia ohjelmistoliiketoiminnan haasteita. Yritysten on kyettävä hallitsemaan ja jatkuvasti parantamaan ohjelmistotuotantoprosessiaan. Niiden pitää pystyä käsittelemään teknologian nopea kehitys, toimimaan globaalissa toimintaympäristössä ja ylläpitämään omia tuotteitaan. Yritysten on myös kehitettävä organisaatiotaan, jotta se pystyy vastaamaan kasvun mukanaan tuomiin haasteisiin. Richardson ja Gresse von Wangenheim (2007) toteavat kuitenkin, että suuret ja pienet yritykset vaativat usein erilaisia lähestymistapoja toimintansa kehittämisessä johtuen yritysten erilaisista liiketoimintamalleista ja tavoitteista. Suurilla ja pienillä yrityksillä on myös

usein erilainen asiakaskunta, niissä on erisuuruiset taloudelliset ja henkilöstöresurssit, yritysten ohjelmistotuotantoprosessi ja hallinto ovat erilaiset ja pienten yritysten organisaatio on matalampi. Pienet yritykset eivät siis ole vain suurten yritysten pienoisversioita, vaan ne kohtaavat erilaisia haasteita oman toimintansa kehittämisessä.

Ohjelmistotuotantoprosessin kehittäminen suurissa yrityksissä on tuottanut monia kehittyneitä malleja prosessin arviointiin kuten esimerkiksi SPICE (SPICE 2004), CMM (Paulk, Weber, Curtis & Chrissis 1995) ja siitä edelleen kehitetty CMMI (Chrissis, Konrad & Shrum 2003). Kun otetaan huomioon pienten yritysten erityispiirteet, voidaan suurissa yrityksissä saatua tietämystä ja kokemuksia soveltaa myös pienten yritysten ohjelmistotuotantoprosessin parantamiseen. Rääätälöimällä ja yksinkertaistamalla olemassa olevia malleja niitä voidaan käyttää myös pienissä yrityksissä (Johnson & Brodman 1997).

Ahosen, Junttilan ja Taskisen (2003) mukaan pienille yrityksille erityisesti suunnitellut ohjelmistotuotannon prosessimallit ovat harvassa. Heidän mukaansa olemassaolevat menetelmät (esimerkiksi OMT++, Unified Process ja Catalysis) sopivat parhaiten keskisuurille tai suurille yrityksille ja olemassaolevien menetelmien muuttaminen pienille yrityksille sopiviksi on hyvin kallista. Nunesin ja Cunhan (2000) mukaan suurille yrityksille kehitetyt mallit eivät ole sopivia pienille ohjelmistoalan yrityksille. Heidän mukaansa ne ovat usein liian monimutkaisia pienten yritysten tarpeisiin ja vaativat räätälöintiä (kts. Johnson & Brodman 1997), joka on useimmiten liian kallista ja aikaa vievää pienen yrityksen ohjelmistotuotantoprosessin parantamiseen varatuilla resursseilla. Cattaneon, Fuggettan ja Lavazzan (1995) mukaan pienet ohjelmistoalan yritykset kuitenkin haluavat parantaa prosessejaan ja tuotteidensa laatua, mutta kohtaavat organisatorisia, kulttuurillisia, rahoituksellisia ja teknisiä esteitä yrittäessään kehittää ohjelmistotuotantoprosessiaan. He toteavatkin, että erityisesti organisatoriset ongelmat nousevat merkittäviksi pienten yritysten yrittäessä kehittää ohjelmistotuotantoprosessiaan. Pienillä yrityksillä on ongelmia löytää resursseja ja muodostaa ryhmiä, joiden tehtävänä on yrityksen prosessin kehittäminen.

## 2 PIENTEN OHJELMISTOALAN YRITYSTEN OMINAISPIIRTEET

Pienen ohjelmistoja kehittävän organisaation tai yrityksen määritelmä ei ole täysin selkeä. Se riippuu paljon siitä minkä tyyppisiä yrityksiä tarkastellaan. Esimerkiksi monikansallisessa yrityksessä yrityksen haaraa tai osastoa, johon kuuluu 100 tai jopa enemmän työntekijöitä, voidaan käsitellä pienenä erillisenä organisaationa tai pienenä itsenäisenä yrityksenä ison yrityksen sisällä. Yleensä kuitenkin tällaisen haaran tai osaston kanssa samalla ohjelmistotuotannon osa-alueella toimivista itsenäisistä ohjelmistoalan yrityksistä hyvin pieni osa on kooltaan sata työntekijää tai suurempia. Joka tapauksessa monet pienet viidestä viitentoista työntekijänkin yritykset ovat tärkeässä roolissa ohjelmistojen kehittämisessä ohjelmistotuotannon eri alueilla. Tällaiset pienet yritykset ovat joustavia ja kehittävät yleensä ohjelmistoja laajemman asiakaskunnan asemasta jollekin tietylle tarkkaan rajatulle alustalle ja tunnetulle asiakkaalle. Pienet yritykset ovat yleensä erikoistuneet jollekin rajatulle ohjelmistotuotannon alueelle kuten vakuutusalan ohjelmistoihin, kuljetusalan ohjelmistoihin tai koulutukseen liittyviin ohjelmistoihin. Tämä varmistaa yritysten menestymisen ja säilymisen markkinoilla. Näiden yritysten erityispiirteet, kuten yritykselle ominaiset työskentelytavat ja työntekijöiden väliset suuria yrityksiä vahvemmat suhteet, vaativat pienille yrityksille sopivaa ohjelmistotuotantoprosessin kehityksen hallintaa. (Horvat, Györkös & Rozman 2000)

Wardin, Laitisen ja Fayadin (2000) mukaan pienten yritysten organisaatio on matala, eikä niiden organisaation rakennetta ole kovinkaan formaalisti määritelty. Heidän mukaansa pienen yrityksen organisaation rakenne muodostuu enemmänkin rooleista ja vastuista ja elää viikoittaisten kriisien mukana. Se ei siis ole tarkasti määritetty, vaan enemmänkin tarpeen mukaan aina itsensä uudelleen järjestävä. Ward ym. (2000) löytävät tällaisesta organisaatiosta kaksi etua: Ensiksi lyhyempi kommunikointietäisyys päättäjien ja työntekijöiden välillä mahdollistaa ohjelmistotuotantoprosessin paremman näkyvyyden ja ongelmiin puuttumisen hyvin nopeasti niiden ilmaannuttua ja toiseksi pienen yrityksen jatkuvan muutoksen tilan, jota voidaan hyödyntää uusia prosesseja ja menetelmiä käyttöönotettaessa.

Wardin ym. (2000) mukaan pienet yritykset yleensä elävät ja kuolevat sen perusteella kuinka hyvin ne saavat palkattua lahjakkaita työntekijöitä. He toisaalta toteavat, että pelkkä lahjakkuus ei yksin riitä, mutta ilman sitä menestyminen on hyvin hankalaa. Pienen yrityksen, jolla ei välttämättä ole kovin suurta varallisuutta, onkin löydettävä keinoja lahjakkaiden työntekijöiden pitämiseen yrityksessä, jotta yritys säilyy kilpailukykyisenä. Wardin ym. (2000) mukaan suuressa yrityksessä, jossa on laajempi työntekijäjoukko, on mahdollista luoda monen vuoden urapolkuja, siirtää lahjakkaita tekijöitä sellaisiin tehtäviin, joissa heitä tarvitaan eniten ja toisaalta



siirtää hankalia ja tuottamattomia tekijöitä sisäisesti muihin tehtäviin. Pienten yritysten on taas hyödynnettävä parhaalla mahdollisella tavalla niitä työntekijöitä, joita on saatavissa ja selvittää mahdollisimman hyvin mitä työntekijät osaavat tehdä parhaiten ja toisaalta mitä he haluavat tehdä. Lahjakkaat työntekijät ovat myös yleensä kiinnostuneita yleensäkin menestymisen mahdollisuudesta joko yritys tai henkilökohtaisella tasolla. He toki haluavat myös kilpailukykyisen korvauksen työstään, mutta se voi olla myös jotain muutakin kuin vain rahallista korvausta. Wardin ym. (2000) mukaan yritysten on kuitenkin hyvä ymmärtää, että pelkästään työntekijän lahjakkuus ei ole ratkaisevaa, vaan myös se kuinka työntekijä on lahjakkuuttaan halukas käyttämään esimerkiksi opastamalla kokemattomampia työntekijöitä tai ottamalla vastuullisempaa roolia projektien johtamisessa.

Ward ym. (2000) toteavat kärjistäen, että pienten ohjelmistoalan yritysten nk. ”pimeä puoli” on kurittomuus. Heidän mukaansa yrityksiin palkatut uudet ohjelmistosuunnittelijat eivät automaattisesti itse kehitä itselleen prosesseja ja laadukkaita käytäntöjä, jos yrityksestä puuttuu hyvin määritetty kehitysohjelma, joka kannustaa ja tukee prosessiosaamisen kehittämistä. He toteavatkin, että kuri tässä yhteydessä tarkoittaa välttämättömien menestymistä lisäävien käytäntöjen löytämistä ja niiden noudattamista. Pienissäkin yrityksissä ei voida unohtaa näitä käytäntöjä, vaikka niissä muuten voitaisiinkin käyttää kevyempää prosessimallia. Heidän kokemuksensa mukaan lahjakkaat ohjelmistosuunnittelijat myös arvostavat tällaista kuria, joka on toteutettu järkevällä tavalla yrityksen ja projektin vaatimuksiin nähden ja auttaa näin ollen yritystä menestymään. Wardin ym. (2000) mukaan pienissä yrityksissä tulisi olla sellainen ohjelmistotuotantojohtaja, joka osaa kouluttaa ja viestiä yrityksen muiden vähemmän teknisten päättäjien kanssa ohjelmistotuotannon realiteeteista ja siitä mikä pitkällä tähtäimellä on kestävään kehitykseen johtavaa toimintaa.

Wardin ym. (2000) mukaan pienissäkin yrityksissä tarvitaan koodin katselmointeja, mutta niistä täytyy tehdä yrityksen kokoon sopivia. Heidän mukaansa normaalit katselmointikäytännöt sisältävät niin monia rooleja, että pienten yritysten resurssit loppuvat, jos käytäntöjä yritetään soveltaa. He toteavatkin, että katselmointikäytännöistä pitää tunnistaa tärkeimmät tavoitteet ja löytää tai tehdä niistä sellaiset karsitut versiot, jotka tukevat parhaalla mahdollisella tavalla näitä tärkeimpiä tavoitteita. Lopuksi he toteavat, että yrityksen koosta huolimatta tärkeimmät asiat projektien läpiviennissä ovat pääosin kuitenkin samanlaisia niin pienissä kuin suurissakin yrityksissä. Esimerkiksi aikatauluissa pysyminen on sekä pienissä että suurissa yrityksissä tärkeää ja siihen voidaan käyttää molemmissa hyvinkin pitkälle samanlaisia keinoja.

Dybån (2000) mukaan pienet ohjelmistoalan yritykset toimivat jatkuvassa toimintaympäristön aiheuttamassa turbulenssissa. Hänen mukaansa pienet

yritykset tarvitsevat sellaisen lähestymistavan kehittymiseensä, joka ottaa huomioon suunnittelun ja toiminnan välisen lyhyen ajan, suunnittelun ja toiminnan suhteen käytännön toteutuksen yksityiskohtiin ja sen, että luovuutta tarvitaan, jotta toimintaympäristöä pystytään hallitsemaan ja ymmärtämään. Dybån (2000) mukaan ohjelmistotuotanto on suurimmaksi osaksi luovuutta vaativaa ja sosiaalista toimintaa, jossa muutoksen hallitseminen on tärkeämpää kuin tasapainotilan saavuttaminen. Hän jatkaa, että todellisuus suurimmassa osassa ohjelmistoja tuottavia pieniä organisaatioita on satunnainen ja monisuuntainen muutoksen virta, johon kuuluu jatkuvia neuvotteluja ja uudelleen neuvotteluja ohjelmistoa muovaavien ryhmien kesken ja välillä. Dybå (2000) toteaa, että kurinalaisuutta ohjelmistojen kehittämisessä ei kuitenkaan saa kokonaan hylätä. Hänen mukaansa ohjelmistotuotannossa on vallittava tasapaino kurinalaisuuden ja luovuuden välillä.

Fayadin, Laitisen ja Wardin (2000) mukaan ohjelmistoteollisuuden kasvu on luonut paljon pieniä yrityksiä ja he toteavatkin, että ohjelmistotuotannon kirjallisuudessa ei ole riittävästi tarkasteltu asioita pienten yritysten näkökulmasta. He listaavat tällaisiksi näkökulmiksi yrityksen koon, ohjelmistojen kehitysmallin, kehitettävän ohjelmiston koon ja kehityksen nopeuden. He toteavat, että monet ohjelmistoja tuottavista yrityksistä ovat pieniä, mutta tuottavat silti merkittäviä ohjelmistoja. Heidän mukaansa pienet yritykset tarvitsevat omiin tarpeisiinsa ja liiketoimintaansa räätälöityjä ohjelmistotuotannon malleja ja käytäntöjä. Heidän mukaansa olemassaolevat käytännöt, jotka on tehty suuria sopimus pohjaisia ohjelmistoja valmistaville yrityksille, eivät toimi parhaalla mahdollisella tavalla, jos niitä yritetään ottaa suoraan käyttöön pienissä yrityksissä.

Horvatin ym. (2000) mukaan pienet yritykset voidaan jakaa kolmeen eri tyyppiin, kun otetaan huomioon ohjelmistotuotantoprosessin kehittämisen vaatimukset ja inhimillisten tekijöiden vaikutus prosessin kehittämiseen.

- Tyyppi A: Suuren ohjelmistoalan yrityksen pieni haarakonttori
- Tyyppi B: Pieni itsenäinen yritys
- Tyyppi C: IT-osasto suuressa ohjelmistoalan yrityksessä

Horvatin ym. (2000) mukaan yksi pienen ohjelmistoalan yrityksen tyyppi (A) on suuren ohjelmistoalan yrityksen pieni haarakonttori ja se voidaan käsittää pieneksi itsenäiseksi yritykseksi. Suurissa yrityksissä työnteon menetelmät, työntekijöiden roolit ja tuotettavat dokumentit on kuitenkin jo tarkasti ennalta määritetty. Tätä vaaditaan, jotta pystytään varmistamaan riittävä kommunikointi ohjelmistotuotantoon osallistuvien yrityksen eri työntekijöiden välillä. Suurissa yrityksissä ohjelmistotuotantoprosessia on väistämättä jouduttu yrityksen kasvaessa kehittämään ja niissä on jo valmiiksi saavutettu ohjelmistotuotantoprosessin taso, joka mahdollistaa työntekijöiden välisen toimivan kommunikaation. Suuren yrityksen olemassa olevaa

ohjelmistotuotantoprosessia käytetäänkin yleensä myös yrityksen haarakonttoreissa. Jo toiminnassa olevan yrityksen on helpompi perustaa haarakonttoreita, kuin kokonaan uuden yrityksen aloittaa puhtaalta pöydältä. Samoin haarakonttoreissa on myös helpompi kehittää ja käyttää emoyritykseltä valmiiksi saatua ohjelmistotuotantoprosessia kuin aloittaa ohjelmistotuotantoprosessin kehittäminen kokonaan alusta. Suuressa yrityksessä on siis olemassa valmis ohjelmistotuotantoprosessi, johon löytyy osaamista ja jonka haarakonttori voi lähes suoraan ottaa käyttöönsä. Tästä johtuen haarakonttorin työntekijöiden määrä ei vaikuta niin paljon ohjelmistotuotantoprosessin kehittämiseen kuin se vaikuttaa kahdessa muussa yritystyyppissä.

Toinen pienten yritysten tyypeistä (B) on Horvatin ym. (2000) mukaan pieni itsenäinen yritys, joka toimii markkinoilla itsenäisesti, ja joka ei siis ole suoraan riippuvainen toisista yrityksistä. Tällaisella yrityksellä on omat toimintatavat, menetelmät ja organisaatio.

Kolmas pienten yritysten tyyppi (C) on Horvatin ym. (2000) mukaan suuren yrityksen IT-osasto. IT-osastot suurissa eri teollisuushaarojen yrityksissä, kuten esimerkiksi lääketeollisuudessa ja pankkitoiminnassa, voidaan käsittää itsenäisiksi yrityksiksi suurten yritysten sisällä, koska niillä on paljon yhtäläisyyksiä tyyppin B yritysten kanssa. Suurten yritysten IT-osastojen tärkeä erityispiirre on kuitenkin se, että niiden asiakkaita ovat pääasiassa tai kokonaan yrityksen toiset osastot, jotka ostavat yrityksen sisäisesti tietojenkäsittelypalvelut IT-osastolta.

Myös Brodmanin ja Johnsonin (1994) mukaan haarakonttorit ja erilliset IT-osastot suurissa yrityksissä toimivat samalla tavalla ja kohtaavat samanlaisia ongelmia kuin pienet itsenäiset yritykset. Heidän mukaansa haarakonttori tai IT-osasto käyttäytyy pienen yrityksen tavoin, kun se toimii suuren yrityksen sisällä omassa kustannuspaikassaan. Brodman ja Johnson (1994) toteavat että myös haarakonttoreiden ja IT-osastojen projektit ovat pääsääntöisesti pieniä, samalla tavalla kuin pienissä itsenäisissä yrityksissäkin.

## **2.1 Pienten yritysten ohjelmistotuotantoprosessin ominaispiirteet**

Myös pienten itsenäisten ohjelmistoalan yritysten päätavoitteena ohjelmistotuotantoprosessin kehittämisessä on parantaa yrityksen kaupallista menestystä ja optimoida yrityksen työskentelytapoja mahdollisimman kustannustehokkaiksi. Pienten yritysten ohjelmistotuotantoprosessin erityispiirteet on otettava huomioon, jotta prosessin kehittäminen onnistuisi mahdollisimman hyvin.

Fayadin, Laitisen ja Wardin (2000) mukaan ohjelmistotuotannon kirjallisuudessa yleensä oletetaan ohjelmistojen tuottamisen perustuvan

sopimusmalliin, jossa ohjelmiston tekemisestä on olemassa selkeä sopimus, johon kaikki osapuolet ovat sitoutuneet. Heidän mukaansa näin ei kuitenkaan aina ole, vaan pienet yritykset tai IT-osastot saattavat tehdä suurille yrityksille sisäisiä koko yrityksen laajuisia sovelluksia, joilla ei ole tarkkaan määritettyä asiakasta, eikä silloin myöskään välttämättä selkeää sopimusta ohjelmiston kehittämisestä. He toteavat myös, että loppuun asti tuotteistettuja sovelluksia rakentavat pienet yritykset eivät noudata kirjallisuudessa oletettua sopimusmallia ohjelmistojensa kehityksessä.

Ohjelmistotuotannon muuttuminen yhä enemmän kilpailuksi toimialaksi aiheuttaa paineita ohjelmistojen nopeammasta kehittämisestä ja uudenlaisten kehitysmallien käyttöönottamisesta. Tämä taas aiheuttaa muutoksia sovellettaviin ohjelmistotuotannon käytäntöihin. Kun ohjelmiston kehittämisen nopeus on tärkeintä, niin uudelleen käyttöön tähtäävää kehitystä voidaan jopa tietoisesti välttää. Myös kehitettävien ohjelmistojen koko kasvaa jatkuvasti ja näyttää siltä, että yhä pienemmät ryhmät kehittävät yhä suurempia sovelluksia. Tästä voidaan tulla siihen johtopäätökseen, että pienempien ryhmien on otettava käyttöön menetelmiä, jotka on alun perin kehitetty suuremmille projekteille ja ryhmille. Pienet ryhmät eivät kuitenkaan tarvitse kaikkia suurten projektien menetelmiä. Merkittävä ongelma suurille yrityksille ja projekteille suunnatuissa menetelmissä on se, että ne eivät skaalaudu kovinkaan hyvin alaspäin. Onkin yllättävää kuinka vähän organisaation tyyppiin ja kokoon kiinnitetään huomiota kirjallisuudessa, kun tutkitaan ohjelmistotuotantoprosessia ja sen kehittämistä. (Fayad ym. 2000).

Fayad ym. (2000) listaavat asioita, jotka aloittelevissa pienissä ohjelmistoalan yrityksissä tehdään yleensä eri tavalla kuin suurissa yrityksissä. Heidän mukaansa pienissä yrityksissä ei välttämättä ole järkevää keskittyä uudelleen käyttöön tähtäävään kehitykseen, vaan tärkeintä on saada ohjelmisto mahdollisimman nopeasti valmiiksi. He toteavat kuitenkin, että yrityksen alkaessa menestyä uudelleen käytöstä tulee tärkeämpää ja alussa nopeasti tehtyjen ratkaisujen muuttaminen voi olla silloin kallista. He viittaavatkin tässä Brooks (1995, 115-123) "Plan to throw one away" -suositukseen. Fayad ym. (2000) listaavat myös pienten aloittelevien ohjelmistoalan yritysten ongelmiksi, että dataa edellisistä projekteista ei ole saatavilla ja se hankaloittaa projektien kustannusten ja niiden vaatiman ajan arvioimista. Myöskään selkeitä ja muuttumattomia vaatimuksia ei ole olemassa ja resursseja on saatavilla hyvin vähän, jolloin esimerkiksi laadunvarmistukseen ei ole osoittaa omia vastuuhenkilöitään. Inkrementaalinen kehitysmalli ja osittain valmiiden versioiden julkaisu aiheuttaa sen, että sovelluksen analyysi ja suunnitteluvaihetta ei välttämättä ole vielä viety loppuun asti, kun sen ensimmäinen versio on jo toimitettu asiakkaalle. Fayadin ym. (2000) mukaan yhden koodirivin kustannusta ei ole välttämättä järkevää arvioida pienissä yrityksissä. Niille on yleensä tärkeämpää mitata tavoitteeseen pääsemiseksi tarvittavaa aikaa ja resursseja ja saada ohjelmisto mahdollisimman nopeasti

markkinoille. Ohjelmiston kustannus on silloin vasta toissijaisena tavoitteena. He jatkavatkin, että pienet yritykset pitävät usein tärkeämpänä valmiiden ja saatavilla olevien ulkoisesti tuotettujen komponenttien ja viitekehysten kustannuksia ja niiden maksimaalista hyödyntämistä, kuin yhden koodirivin kustannuksen laskemista.

Russin ja McGregorin (2000) mukaan projektin pienuus voidaan määrittää neljän eri ympäristötekijän perusteella. Näitä ovat:

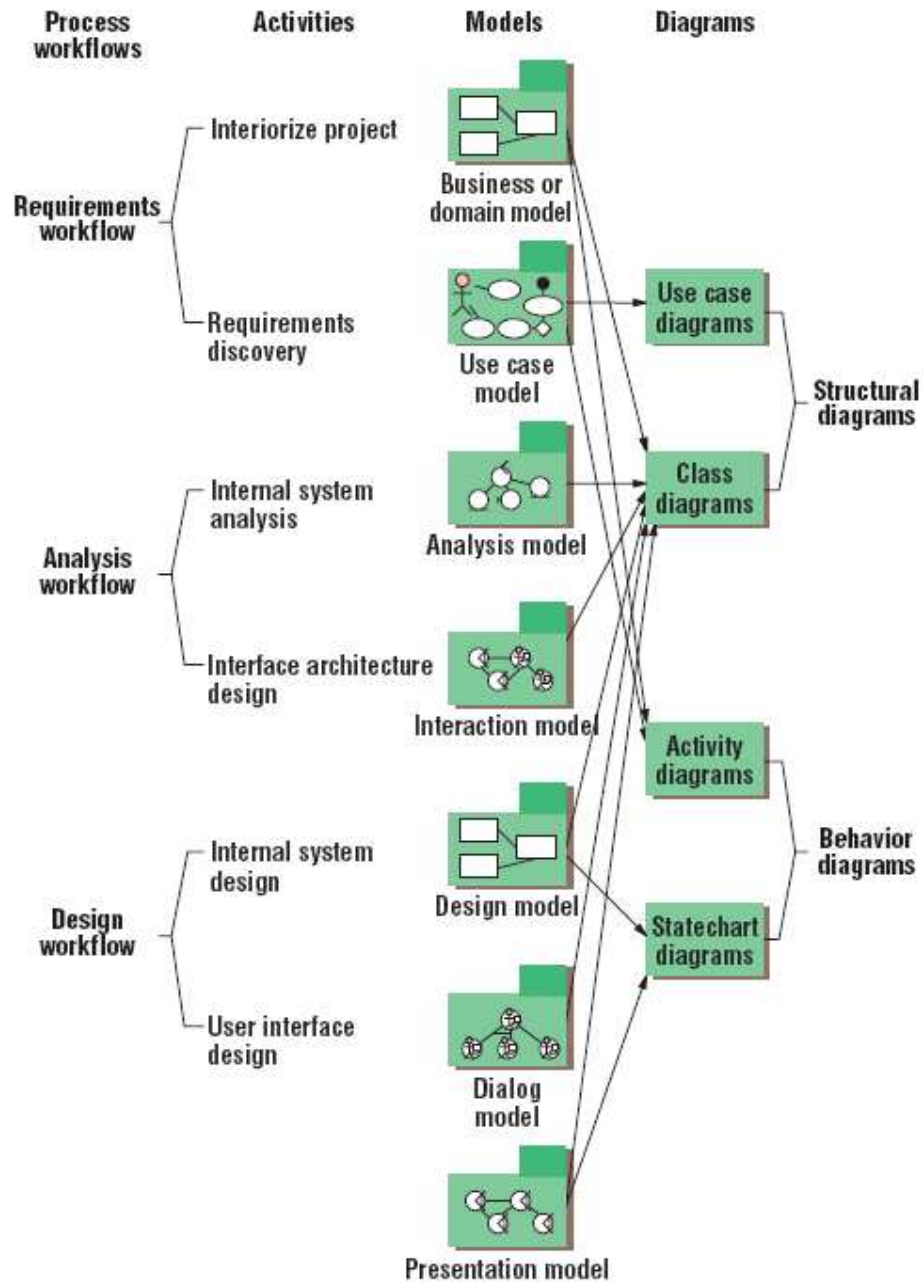
- Kehitysorganisaation koko
- Projektin monimutkaisuus
- Laadun mittaaminen
- Henkilökunnan vuorovaikutus

Pieni projekti on suuressa organisaatiossa eri asemassa kuin pienessä organisaatiossa. Suurempi kehitysorganisaatio pystyy tarjoamaan projektille paljon enemmän erilaisia tukitoimintoja, jotka eivät ole mahdollisia pienemmissä organisaatioissa. Projektin monimutkaisuus nostaa sen projektiryhmälle asettamia vaatimuksia ja mitä monimutkaisempi projekti ja sen vaatimukset ovat, sitä paremmin määritellyn ja formaalimman prosessin projektin läpivienti vaatii. Pienten projektien onkin usein helpompi saavuttaa riittävä laatu. Monimutkaisemmissa projekteissa prosessin on tuettava laadun ja sen eri osatekijöiden mittaamista. Näitä ovat esimerkiksi luotettavuus, turvallisuus ja tehokkuus. Pienissä projekteissa henkilökunnan vuorovaikutus on hyvin vaapaamuotoista ja niitä tehdään matalissa organisaatioissa, jolloin reagoiminen muutoksiin ja uusiin ongelmiin on huomattavasti nopeampaa kuin isoissa projekteissa, joissa päätöksenteko on hitaampaa. Projektiryhmän koon kasvaessa myös tarvittava vuorovaikutuksen määrä kasvaa.

Nunesin ja Cunhan (2000) mukaan pienten yritysten on vaikeaa löytää resursseja ja muodostaa ryhmiä, joiden tehtävänä on yrityksen ohjelmistotuotantoprosessin kehittäminen. Uudet käytännöt pienissä yrityksissä voivat vaikuttaa jo olemassa olevien tuotteiden ylläpitoon ja asiakkaiden kysyntään. Yritysten johto pelkää kustannusten nousua, tuotteiden markkinoille saamisen hidastumista, ja sitä että prosessin parantamiseen käytetyille varoille ei saada vastinetta. Yleensä myös ohjelmistosuunnittelijat vastustavat uusien metodien, työkalujen, teknologioiden ja standardien käyttöönottoa. Ohjelmistotuotantoprosessin arviointimenetelmien mukaisten kypsyystasojen saavuttaminen (esimerkiksi CMMI) vaatii pitkäaikaista sitoutumista ohjelmistotuotantoprosessin kehittämiseen. Rahoituksellisia ongelmia syntyy, koska resurssien allokointi ohjelmistotuotantoprosessin kehittämiseen ja mahdollisesti tarvittavien ulkopuolisten konsulttien palkkaaminen on kallista. Myös nykyaikaisten metodien ja työkalujen hankinta ja käyttöönotto on kallista ja vaatii työntekijöiden kouluttamista. Se taas

väliaikaisesti laskee pienten yritysten tuottavuutta, joissa resursseja on rajallisesti. Pienet yritykset kärsivät myös teknisistä ongelmista, koska monimutkaisten suurille yrityksille kehitettyjen ohjelmistotuotantoprosessien ja metodien räätälöinti pienten yritysten tarpeisiin ei ole yksinkertaista. (Nunes & Cunha 2000)

Nunesin ja Cunhan (2000) mukaan olemassaolevat suurille yrityksille luodut prosessimallit eivät myöskään tarpeeksi hyvin ota huomioon pienestä organisaatiosta saatavia hyötyjä. He toteavatkin, että pienet yritykset ovat yleensä esimerkiksi joustavampia, hallittavampia ja reagoivat nopeammin muutokseen. Heidän mukaansa isoihin yrityksiin verrattuna myös kommunikaatio on tehokasta pienessä organisaatiossa sekä sisäisesti että ulkoisien kumppaneiden ja asiakkaiden kanssa. Pienillä yrityksillä on rajoitetut resurssit eikä niillä ole varaa epäonnistua projekteissaan, mutta pienten yritysten välillä kaoottisilta näyttävät toimintatavat eivät välttämättä tarkoita sitä, että projektit eivät onnistuisi. Jo ennen kuin ohjelmistotuotannon tutkijat kehittivät spiraalimallin ja iteratiivisen ja inkrementaalisen ohjelmistotuotantoprosessin, pienet yritykset olivat tehneet voittoa toimintatavoilla, jotka muistuttivat näitä malleja ja saattoivat näyttää hyvinkin ennalta määräämättömiltä ja joskus jopa kaoottisilta. Nunesin ja Cunhan (2000) mukaan yleensä pienten yritysten toimintamallina on valmista tuotetta kohti kehittyvien prototyyppien asteittainen kehitys painottuen hyvin vahvasti itse ohjelmiston toteutukseen. Heidän mukaansa tämä on johtanut pienissä yrityksissä projekteihin joilla ei ole varsinaista alku tai päätepistettä, ohjelmistojen matalaan laatuun ja siihen, että jossain yksittäisessä projektissa saavutettua menestystä ja onnistumista ei ole pystytty hyödyntämään yrityksen muissa projekteissa. Nunes ja Cunha (2000) toteavatkin, että johtajien ja projektien toteuttajien näkökulmien yhdistämiseen pienissä yrityksissä tarvitaan malli, joka tukee kummankin osapuolen erityispiirteitä ja tarpeita. Tätä varten he ovat kehittäneet prototyyppilähtöisen Wisdom-mallin (KUVIO 2), joka ottaa huomioon pienten yritysten toteutuslähtöisen näkökohdan ohjelmistojen tuottamiseen ja rakentuu pieniä yrityksiä yksilöivien piirteiden kuten nopean muutokseen reagoinnin, joustavuuden ja toimivan kommunikaation ympärille.



KUVIO 2. Työnkulut, toiminnot, mallit ja diagrammit Wisdom-mallissa (Nunes & Cunha 2000).

Horvatin ym. (2000) mukaan pienissä itsenäisissä ohjelmistoalan yrityksissä kohdataan seuraavanlaisia haasteita:

- Suuri riippuvuus yksilöistä
- Pieni määrä työntekijöitä, joille jakaa eri rooleja ja tehtäviä

- Inhimillisten tekijöiden ja psykologisten kysymysten suuri vaikutus
- Riippuvaisuus yksittäisten tärkeiden projektien onnistumisesta
- Asiakaskommunikaation tärkeys
- Hankaluudet ohjelmistotuotantoprosessin kehittämiseen panostamisessa ja sitä kehittävien työntekijöiden resursoinnissa

Pienet ohjelmistoalan yritykset ovat voimakkaasti riippuvaisia yksittäisistä työntekijöistä, koska pienissä yrityksissä työskentelee lukumääräisesti vähemmän työntekijöitä kuin suurissa yrityksissä. Silloin samojen työntekijöiden tehtäviksi annetaan yleensä samankaltaisten ongelmien ratkaiseminen ja näistä työntekijöistä tulee asiantuntijoita kyseisten ongelmien ratkaisemisessa. Koska ohjelmistotuotantoprosessia ei ole, tai se on hyvin summittainen, ovat projekteissa tuotetut dokumentit vajavaisia. Ilman riittävää dokumentaatiota projektien onnistuminen ja jopa koko yrityksen menestyminen on riippuvainen yksilöistä, joilla on jo valmiiksi tietämys aikaisempien projektien perusteella kuinka projekteissa vastaan tulevia samankaltaisia ongelmia on nopeinta ja helpointa ratkaista.

Pienissä ohjelmistoalan yrityksissä yrityksen eri roolit ja tehtävät jaetaan harvojen työntekijöiden kesken. Koska pienellä yrityksellä on rajallinen määrä työntekijöitä, tulee saman työntekijän tehtäväksi yleensä monien erilaisten ohjelmistotuotantoprosessin tehtävien suorittaminen. Erilaiset ohjelmistotuotannon tehtävät vaativat erilaisia taitoja ja voidaankin kyseenalaistaa yhden työntekijän soveltuvuus optimaalisesti kaikkiin pienessä yrityksessä suorittamiinsa tehtäviin. Kun työntekijät suorittavat tehtäviä, joihin eivät optimaalisesti sovellu, laskee myös ohjelmistotuotannon kokonaistehokkuus.

Inhimillisillä tekijöillä ja psykologisilla kysymyksillä on suuri vaikutus yrityksen toimintaan pienissä ohjelmistoalan yrityksissä. Pienet yritykset syntyvät yleensä muutaman yksilön innostuneesta aloitteesta ja niiden ensimmäinen projekti on myös yleensä pieni ja tehdään ennalta määrätulle asiakkaalle pienellä budjetilla, rajallisin resurssein ja rajoittuneella laitteistolla. Tällaisessa yrityksessä työntekijöiden väliset ihmissuhteet perustuvat yleensä ystävyyteen ja syvään omistautumiseen yritykselle ja sen tavoitteille. Ongelmia syntyy yleensä kun yritys kasvaa ja sen organisaation rakenne ei pysty tukemaan yrityksen johtamista. Horvatin ym. (2000) mukaan kymmenestä viiteentoista työntekijää on kriittinen koko yritykselle, jonka jälkeen se tarvitsee jonkinlaisen ennalta määrätyn organisaatorakenteen. Ohjelmistotuotantoprosessin kehittämisen onnistuminen pienessä ohjelmistoalan yrityksessä vaatiikin, että jokainen yrityksen työntekijä hyväksyy ohjelmistotuotantoprosessin tavoitteet ja tehtävät. Sen vuoksi



inhimilliset, sosiaaliset ja kulttuuriin liittyvät kysymykset tulisi ottaa huomioon pienen yrityksen ohjelmistotuotantoprosessia kehitettäessä.

Pienet ohjelmistoalan yritykset ovat usein riippuvaisia muutamista tärkeistä projekteista ja niiden onnistumisesta. Niinpä pienissä yrityksissä suoritetaan yleensä useita eri projekteja yhtäaikaisesti vain siitä syystä, että kriisin sattuessa muista projekteista tulevilla rahallisilla tuloilla ja muulla tuella pystytään auttamaan ongelmissa olevaa projektia. Uusia projekteja aloitetaan liian helposti, jotta pystytään tukemaan jo olemassa olevia tärkeiden asiakkaiden projekteja. Tämä johtaa yleensä ajan kuluessa vain uusiin kriiseihin.

Toimiva asiakaskommunikaatio on hyvin tärkeää pienille ohjelmistoalan yrityksille. Ne tuottavat yleensä ohjelmistoja ennalta tunnetuille asiakkaille ja kommunikointi asiakkaan kanssa on intensiivistä. Varsinkin vaatimusmäärittelyssä, analyysivaiheessa, testauksessa, sovelluksen toimittamisessa ja ylläpidossa kommunikointia asiakkaan kanssa on paljon. Tästä johtuen asiakkaiden kanssa kommunikoinnin kehittäminen on tärkeää. Asiakasviestinnän kehittämisessä myös parannukset ohjelmistotuotantoprosessissa nähdään nopeasti.

Pienissä ohjelmistoalan yrityksissä on yleensä hankala löytää vapaita resursseja ohjelmistotuotantoprosessin kehittämiseen. Niissä ohjelmistotuotantoprosessin kehittämiseen joudutaan yleensä resursoimaan työntekijöitä, jotka ovat jo valmiiksi ylikuormitettuja muiden projektien tehtävistä. Työntekijöiden resursoiminen ohjelmistotuotantoprosessin kehittämiseen vaikuttaa silloin muiden projektien etenemiseen ja aiheuttaa siten välillisesti lisää kustannuksia ohjelmistotuotantoprosessin kehittämiseen.

Ahonen, Junntila ja Taskinen (2003) löysivät ongelmia, jotka olivat yhteisiä kaikille heidän tutkimilleen pienille ohjelmistoalan yrityksille. Näitä olivat dokumenttienhallinta, versionhallinta ja koodin luettavuus ja ylläpidettävyys. He toteavat, että nämä ongelmat eivät ole harvinaisia eivätkä ne liity vain pieniin yrityksiin. Heidän mukaansa ongelmat kuitenkin korostuvat pienissä yrityksissä, koska niissä ei ole erillisiä resursseja laadun varmistukseen.

Horvatin ym. (2000) mukaan pienten ohjelmistoalan yritysten ohjelmistotuotantoprosessin tulisi olla yksinkertainen, helppo ymmärtää ja sen pitäisi tarjota kestävä ohjenuora prosessin määrittämiseen ja esittämiseen. Heidän mukaansa ohjelmistotuotantoprosessin pitäisi kuitenkin kattaa kaikki ne vaiheet, jotka on suoritettava ohjelmistotuotantoprojektin läpiviemiseksi. Ahosen ym. (2003) mukaan suurin osa pienten yritysten haasteista liittyy rajallisiin resursseihin ja tiukkoihin budjetteihin. Heidän mukaansa pienet yritykset tarvitsevatkin hyvin kevyitä ohjelmistotuotantoprosessin parantamis- ja kehitystekniikoita ja ketteriä menetelmiä tukevia prosesseja.

## 2.2 Yrityksen koon vaikutus ohjelmistotuotantoprosessiin ja sen kehittämiseen

Russin ja McGregorin (2000) mukaan pienillä ohjelmistotuotantoprojekteilla on usein suuri määrä ulkoisia riippuvuuksia yhtä projektiryhmän jäsentä kohden. Heidän mukaansa pienen projektiryhmän jäsenillä on esimerkiksi läheisempi suhde asiakkaisiin, joka vaatii enemmän ryhmän jäsenten välistä interaktiota kuin suuremmissa projekteissa. He toteavat tämän johtuvan siitä, että pienten projektien asiakkaat eivät välttämättä ole samanlaisia kuin suurempien projektien asiakkaat. Pienten projektien asiakkaat saattavat olla hyvin tarkalla tasolla mukana suoraan itse projektissa ja sen päätöksenteossa. Pienten projektien on usein saavutettava sama laadun taso, joka on ominaista isommille projekteille, mutta kuitenkin pienemmällä määrällä projektiryhmän jäseniä. Lisäksi pienissä projekteissa on usein mukana työntekijöitä, jotka osallistuvat projekteihin vain osittaisella työpanoksella. Tällaisia ovat esimerkiksi jonkin liiketoiminta-alueen asiantuntijat, arkkitehdit ja systeemitestaushenkilöstö. Eri henkilöiden erilainen työpanos projektissa vaatii yleensä enemmän koordinaatiota ja kanssakäymistä, jotta kaikkien työntekijöiden resurssit ja asiantuntemus saadaan parhaalla mahdollisella tavalla hyödynnettyä.

Russin ja McGregorin (2000) mukaan ohjelmistotuotantoprosessi on myös tärkeä pienille projekteille, koska sen avulla voidaan välttää niille tyypillisiä ongelmia. Tällaisia ovat esimerkiksi:

- Tehtävien priorisointi niiden mielekkyyden mukaan
- Nopeasti muuttuvassa ympäristössä ohjelmistoa tai projektia ei saada koskaan valmiiksi
- Resurssien tehoton hyödyntäminen

Russ ja McGregor (2000) toteavat, että jos prosessi ei ohjaa projektin etenemistä, aiheutuu helposti tilanteita, joissa työntekijät valitsevat itse ja korostavat niitä asioita, joita on mielekästä toteuttaa. Tällöin osa sovelluksesta on toteutettu mallikkaasti, mutta joitain osia saattaa jäädä jopa kokonaan toteuttamatta ja asiakkaan vaatimukset jäävät saavuttamatta. Heidän mukaansa prosessi ohjaa projektiryhmää, jotta mitään laadukkaan ohjelmiston tuottamisen osa-aluetta ei unohdeta.

Nopeasti muuttuvassa ympäristössä ohjelmiston vaatimukset saattavat muuttua ja vanhentua ennen kuin niitä on saatu kunnolla loppuun asti toteutettua. Iteratiivisuutta tukevan ohjelmistotuotantoprosessin avulla ohjelmiston kehityksessä saavutetaan merkkipaaluja (milestone), jolloin ohjelmistotuotannon eri vaiheille saadaan selkeät päätepisteet. Prosessin avulla voidaan välttää myös työntekijöiden tyhjäkäyntiä. Prosessia noudatettaessa

kenellekään ei tule ”mitä tekisin tänään” -tunnetta ja ohjelmiston tuottaminen tapahtuu oikeassa prosessin mukaisessa järjestyksessä. (Russ & McGregor 2000).

Dybån (2003) mukaan ohjelmistotuotantoprosesseilla on merkittävä rooli isoissa organisaatioissa. Ne koordinoivat organisaation eri ryhmien käytäntöjä, niin että käytännöt pysyvät samankaltaisina ja yhteensopivina. Dybån (2003) mukaan isoilla ja pienillä organisaatioilla on merkittäviä toiminnallisia eroja. Pienet organisaatiot ovat enemmän kiinnostuneita käytännöstä, kun taas suuret organisaatiot keskittyvät formaaleihin prosesseihin. Dybån (2003) mukaan suuret organisaatiot olivat merkittävästi enemmän liiketoimintaorientoituneita, pitivät toimintojen ja prosessin mittausta tärkeänä ja niiden johto oli enemmän prosessin kehittämiseen sitoutunutta, kuin pienissä organisaatioissa. Hänen mukaansa pienet organisaatiot taas panostivat suuria enemmän uusien asioiden tutkimiseen ja niiden työntekijät olivat sitoutuneempia prosessin kehittämiseen, kuin suurissa organisaatioissa. Dybån (2003) mukaan olemassa olevan tietämyksen hyödyntäminen oli suunnilleen samanlaista sekä pienissä että suurissa organisaatioissa. Suuret yritykset lisäksi raportoivat huomattavasti suuremman onnistumisasteen ohjelmistotuotantoprosessin kehittämisessä ja parantamisessa, kuin pienet. Dybån (2003) mukaan sekä suuret että pienet organisaatiot, jotka olivat onnistuneet ohjelmistotuotantoprosessin parantamisessa, raportoivat samansuuruista onnistumista prosessin kehittämisessä. Tilastollisesti merkittäviä eroja suurten ja pienten prosessin parantamisessa menestyneiden organisaatioiden välillä hänen mukaansa oli menestyneiden pienten organisaatioiden työntekijöiden suurempi sitoutuminen ja pienissä yrityksissä uuden tietämyksen hyödyntäminen suuria organisaatioita enemmän.

Dybå (2003) toteaa, että suurten ja pienten ohjelmistoja tuottavien organisaatioiden suurin ero on niissä tavoissa, miten ne reagoivat epävakaisiin ja muuttuviin tilanteisiin. Hänen mukaansa pienissä organisaatioissa vastaus muutokseen on uuden tutkiminen, joka johtaa siihen, että epävakaisissa olosuhteissa toimivat pienet organisaatiot tarvitsevat oppimisstrategioita, jotka tukevat mahdollisimman hyvin uuden tutkimista, mutta samalla mahdollistavat myös olemassaolevan kokemuksen ja tietämyksen tehokkaan hyödyntämisen. Dybå (2003) jatkaa, että monet johtajat ovat samaa mieltä siitä, että muutokset heidän kilpailuympäristössään ovat nopeita ja yhä enemmän ennalta arvaamattomia. Hänen mukaansa suurten organisaatioiden johtajat luottivat kuitenkin enemmän aikaisempaan kokemukseensa, kuin uusien mahdollisuuksien ja ratkaisutapojen tutkimiseen. Hän toteaaakin, että suurten organisaatioiden muuttuminen vastauksena ympäristön muutokseen on epätodennäköisempää, kuin pienten organisaatioiden. Pienet organisaatiot pysyvät yleensä orgaanisina ja mukautuvat helposti muutokseen, kun taas suurissa organisaatioissa muodostuu rakenteita, jotka korostavat vakautta, järjestystä ja kontrollia. Tämä taas johtaa siihen, että suurilla organisaatioilla on

usein suuria vaikeuksia mukautua muuttuviin tilanteisiin, koska ne on rakennettu saavuttamaan ennalta määrättyjä tavoitteita, ei niinkään innovoimaan ja luomaan uutta.

Dybån (2003) mukaan suuret yritykset ovat kuitenkin yleensä menestyviä, koska organisaatio kasvaa isommaksi toistuvan menestyksen myötä. Hänen toteaa, että suuret organisaatiot suosivat usein niin kutsuttua parhaiden käytäntöjen (best practices) lähestymistapaa, jossa ohjelmistoja tuotetaan vanhoilla hyväksi havaituilla malleilla. Tämä toimiikin yleensä hyvin vakaisissa olosuhteissa, mutta olosuhteiden nopeasti muuttuessa se saattaa rajoittaa uusien ja usein parempien toimintatapojen tutkimista ja muuttua jopa esteeksi kehitykselle. Lisäksi vanhoja, aikaisemmissa olosuhteissa toimivaksi havaittuja, malleja käytettäessä epäonnistumista ei pidetä hyväksyttävänä. Dybå (2003) toteaa, että tämä onkin linjassa suurten yritysten tavoitteessa tukea vakautta ja lyhyen ajan tehokkuutta. Dybån (2003) mukaan onnistunut ohjelmistotuotantoprosessin parantaminen vaatii kuitenkin myös epäonnistumisen mahdollisuuden hyväksymistä. Hän toteaaakin, että epäonnistuminen on tärkeä alkuvaatimus oppimiselle ja vallitsevan asiantilan (niin kutsuttu status quo) kyseenalaistamiselle.

Dybå (2003) päätyy johtopäätökseen, että organisaation koko ei rajoita sen mahdollisuuksia onnistua ohjelmistotuotantoprosessin kehittämisessä. Hänen mukaansa pienet organisaatiot kehittävät prosessejaan yhtä tehokkaasti kuin suuretkin. Suuret menestyvät organisaatiot korostavat omien parhaiden toimintatapojensa (best practices) käyttämistä ja kehittämistä formaalien mallien, tarkkojen ohjeiden ja tarkistuslistojen avulla. Pienet menestyvät organisaatiot kuitenkin korostavat enemmän uusien mahdollisuuksien tutkimista ja hyödyntävät tehokkaasti projekteissa mukana olevien työntekijöiden eri vahvuusalueita ja luovuutta. Dybån (2003) mukaan pienten organisaatioiden tulisikin hyödyntää vahvuuksiaan työntekijöiden sitoutumisessa ja uuden tietämyksen tutkimisessa, jotta ne pystyvät kehittämään ohjelmistotuotantoprosessiaan yhtä tehokkaasti kuin suuretkin organisaatiot. Hänen mukaansa pienissä organisaatioissa formaaleja prosesseja pitää täydentää ja ohjata epäformaalilla käytännönläheisellä näkökulmalla. Dybån (2003) mukaan kehittyminen vaatii virheiden korjaamisen lisäksi myös sen, että organisaatio on valmis kyseenalaistamaan omia peruskäytäntöjään ja oletuksiaan ja tarvittaessa muuttamaan niitä kehittymisen mahdollistamiseksi.

### **3 SUURILLE YRITYKSILLE KEHITETTYJEN OHJELMISTOTUOTANTOPROSESSIEN JA MENETELMIEN SOVELTUVUUS PIENILLE OHJELMISTOALAN YRITYKSILLE**

Guerreron ja Eterovicin (2004) mukaan menestyäkseen markkinoilla ja vahvistaakseen kilpailuasemiaan, pienten ohjelmistoalan yritysten on pystyttävä näyttämään toteen, että ne käyttävät kansainvälisesti hyväksytyjä ohjelmistotuotantoprosesseja ja käytäntöjä. Brodmanin ja Johnsonin (1994) mukaan ohjelmistotuotantoprosessin kehittäminen on kuitenkin pienille yrityksille kallista ja aiheuttaa niille paljon ylimääräistä kuormitusta muiden työtehtävien ohessa. Toisin sanoen pienten yritysten on vaikea löytää resursseja ohjelmistotuotantoprosessinsa arviointiin ja prosessin kehittämiseen arvioinnin perusteella. Brodmanin ja Johnsonin (1994) mukaan ohjelmistotuotantoprosessin kehittämisen aloittamiselle on olemassa minimikustannus, joka on suunnilleen samansuuruinen yrityksen tai organisaation koosta riippumatta. Heidän mukaansa pienille yrityksillä tämä kustannus aiheuttaa kuitenkin enemmän kuormitusta kuin suurille yrityksille, joiden liikevaihto ja yrityksen toimintoihin käytettävissä oleva rahamäärä ja resurssit ovat pieniä yrityksiä suuremmat. Brodmanin ja Johnsonin (1994) mukaan prosessin kehittämisen aloituskustannukset voivat vähentää pienten yritysten kilpailukykyä markkinoilla. He toteavat, että sellaiset pienet yritykset, jotka eivät kehitä ohjelmistotuotantoprosessiaan tai suuret ohjelmistotuotantoprosessiaan kehittävät yritykset ovat kilpailukykyisempiä kuin pienet yritykset, jotka ovat juuri aloittaneet ohjelmistotuotantoprosessinsa kehittämisen. Pienet yritykset, jotka eivät kehitä prosessiaan eivät kärsi prosessin kehittämisen vaatimista aloituskustannuksista tai sen viemistä resursseista. Suurissa yrityksissä ohjelmistotuotantoprosessin kehittämisen aloittaminen ei kuormita yritystä yhtä paljon kuin vastaavassa tilanteessa olevaa pientä yritystä. Lyhyesti sanottuna ohjelmistotuotantoprosessin arvioinnin ja kehittämisen aloittaminen on pienille yrityksille suhteessa kalliimpaa kuin suurille yrityksille.

Brodmanin ja Johnsonin (1994) mukaan CMM aiheuttaa pienille yrityksille ongelmia, koska monet sen käytännöistä eivät sovellu pieniin projekteihin, jotka ovat tärkeässä roolissa pienissä ohjelmistoalan yrityksissä. Heidän mukaansa pienet yritykset myös pelkäävät, että CMM:n mukaiseen prosessin kehittämiseen käytetyt varat menevät osittain hukkaan, koska yrityksen prosessia arvioivat tarkastajat eivät välttämättä ymmärrä pienten yritysten käyttämiä vaihtoehtoisia lähestymistapoja CMM:n tavoitteiden saavuttamiseksi. Yritys ei tässä tapauksessa saa virallista sertifiointia ohjelmistotuotantoprosessilleen jollekin tietylle tarvitsemalleen CMM:n tasolle, vaikka pystyykin hoitamaan CMM:n vaatimukset omalla toimintatavallaan, joka luultavasti soveltuu paremmin pienelle yritykselle. Brodmanin ja

Johnsonin (1994) mukaan myös suuren yrityksen sisällä oleva ohjelmistotuotanto-organisaatio, kun se toimii omissa kustannuspaikassaan, kohtaa samanlaisia ongelmia ja haasteita arviointimenetelmien käytössä ja ohjelmistotuotantoprosessin kehittämisesä kuin pieni itsenäinen yritys. Heidän mukaansa erillinen ohjelmistotuotanto-organisaatio suuressa yrityksessä voidaankin nähdä pienenä itsenäisenä yrityksenä.

### 3.1 Nykyisten arviointimenetelmien soveltuvuus pienille yrityksille

Mc Cafferyn, Taylorin ja Colemanin (2007) mukaan ohjelmistotuotantoprosessin arviointi arviointimenetelmää käyttäen on edelleen organisaatiolle paras tapa aloittaa oman ohjelmistotuotantoprosessinsa kehittäminen. Heidän mukaansa prosessin arviointi voi osoittaa nykyisen prosessin vahvuudet ja heikkoudet ja siitä johtuen keskittää prosessin kehittämisen ja parantamisen heti sen ydinkohtiin. He toteavat kuitenkin, että resurssien puutteen vuoksi pienet yritykset tarvitsevat kevyen arviointimenetelmän, ja että pienissä yrityksissä yleisesti hyödynnetyt ketterät menetelmät pitää ottaa huomioon käytetyssä prosessin arviointimenetelmässä. Wardin ym. (2000) mukaan suurille yrityksille tehdyt arviointimenetelmät ja mallit skaalautuvat huonosti alaspäin pienten yritysten tarpeisiin sopiviksi. He toteavat, että kypsyystasojen mittaaminen voi olla hyödyllistä myös pienissä yrityksissä, mutta sitä ei kannata välttämättä perustaa suoraan suurille yrityksille suunniteltujen mallien pohjalle.

Gresse von Wangenheimin, Anacleton ja Salvianon (2006) mukaan pienet alle viidenkymmenen työntekijän yritykset huomaavat oman ohjelmistotuotantoprosessinsa arvioimisen olevan hankalaa. Heidän mukaansa tämä johtuu osittain siitä, että monet pienet yritykset eivät ole tietoisia olemassaolevista prosessien arviointimenetelmistä ja standardeista. He toteavat myös, että pienet yritykset olettavat, että arviointimenetelmät ovat kalliita ja aikaa vieviä ja sen vuoksi hankalia käyttää. Gresse von Wangenheimin ym. (2006) mukaan pienet yritykset myös ovat sitä mieltä, että arviointimenetelmät on pääosin suunnattu suurille yrityksille, eivätkä sovellu pienten yritysten käyttöön. Heidän mukaansa yleensä pienten yritysten suurimpana huolenaiheena on saada projektinsa valmiiksi ja tuotteensa mahdollisimman nopeasti markkinoille, jotta yritykset yleensäkin pysyvät hengissä.

Myös Guerreron ja Eterovicin (2004) mukaan pienet yritykset uskovat, että CMM-mallia ei ole mahdollista käyttää pienissä organisaatioissa. Syyksi he mainitsevat alkuperäisen CMM:n kuvaustekstin maininnan siitä, että CMM on kehitetty Yhdysvaltain puolustusministeriön metodiksi sen omien ohjelmistotoimittajien kypsyuden mittaamiseen. Heidän mukaansa yleensä myös tieteelliset artikkelit (Johnson 1994, Keeni 2000, Pitterman 2000 ja Wohlwend & Rosenbaum 1994), jotka tukevat CMM:n hyödyllisyyttä organisaatiolle, kuvaavat tyypillisiksi CMM:ää käyttäviksi yrityksiksi sellaisia

yrityksiä, joilla on satoja työntekijöitä useassa eri toimipisteessä ja maassa. Lisäksi Guerreron ja Eterovicin (2004) mukaan tällaisten yritysten tyypillisiä projekteja ovat monivuotiset budjetiltaan miljoonien dollarien projektit, joissa on mukana kymmeniä yrityksen työntekijöitä. Edelleen heidän mukaansa CMM:ää käyttävien yritysten tyypillisiä asiakkaita ovat maailman suurimmat ohjelmistojen hankkijat ja CMM:ää käyttävillä yrityksillä on kehitystyössä käytössään uusimmat laitteet ja ohjelmistot. Nämä yritykset tuottavat lähinnä kriittisiin olosuhteisiin tarkoitettuja reaaliaikaisia sulautettuja järjestelmiä, kuten esimerkiksi puolustusteollisuuden ja lentokoneiteollisuuden ohjelmistoja ja järjestelmiä. Silloin yritysten asiakkaat luonnollisesti automaattisesti myös vaativat, että yrityksillä on voimassa olevia kansainvälisiä laatusertifikaatteja ja että niiden ohjelmistotuotantoprosessi on tietyllä asiakkaan hyväksymän ohjelmistotuotantoprosessin arviointimenetelmän (esimerkiksi CMM) mukaisella kypsyytasolla.

Brodmanin ja Johnsonin (1994) pienille yrityksille tekemässä kyselyssä ongelmia CMM:n mukaisessa ohjelmistotuotantoprosessin kehittämisessä koettiin olevan:

- CMM:n soveltuvuus yleensäkin ohjelmistotuotantoprosessin parannustyökaluksi
- Ohjelmistotuotantoprosessin parantamisen ja kehittämisen kustannukset
- Ohjelmistotuotantoprosessin kehittäminen muiden liiketoimintojen kustannuksella
- Yrityskulttuuri, joka ei tue ohjelmistotuotantoprosessin parantamista
- Ohjelmistotuotantoprosessin kehittämiseen yrityksen ulkopuolelta saatavissa olevan tuen puute
- CMM:n mukainen organisaatio ei vastaa yrityksen käyttämää organisaatiomallia
- CMM:n mukaiseen ohjelmistotuotantoprosessin kehittämiseen tarvittavien resurssien puute

Brodmanin ja Johnsonin (1994) mukaan pienissä yrityksissä ensisijainen ongelma on riittävän rahoituksen, työvoiman ja ajan resursointi ohjelmistotuotantoprosessin kehittämiseen.

Lähes puolet yrityksistä, joita Brodman ja Johnson (1994) tutkivat tähtäsivät ohjelmistotuotantoprosessin kehityksessään CMM:n kolmannelle tasolle. Heidän mukaansa hyvin harvat yritykset uskoivat pystyvänsä saavuttamaan tasoa kolme korkeamman CMM-tason. Kahdeksan prosenttia tutkituista

yrittäjistä ilmoitti keskittyvänsä prosessin parantamiseen yleensä tai kehittävänsä sellaisia prosessin osa-alueita, joista kokevat saavansa eniten hyötyä. Tällaiset yritykset eivät siis prosessia kehittäessään tähänneet millekään tietylle CMM-tasolle. Brodmanin ja Johnsonin mukaan (1994) yritykset myös kaipaavat enemmän käytännön esimerkkejä ja mallipohjia CMM-ohjeistuksessa. Heidän mukaansa CMM:n käyttäminen voisi olla halvempaa, jos osa CMM-malleista tulisi valmiina, eikä niitä tarvitsisi luoda alusta asti omaa organisaatiota varten. Brodmanin ja Johnsonin (1994) tutkimat yritykset kyseenalaistivat myös sopiiko CMM yhteen muiden yrityksessä käytössä olevien laadunvarmistustapojen kanssa ja onko esimerkiksi jokin tietty CMM-taso pienessä yrityksessä täysin vastaava kuin sama taso suuressa yrityksessä.

Brodmanin ja Johnsonin (1994) mukaan yli kolmasosa pienistä yrityksistä pitää suurimpina CMM:n ongelmina pienessä yrityksessä tarvittavan koulutuksen kalleutta, CMM:n määrittelemien erillisten organisaatioiden, jotka on oltava yrityksessä olemassa, jotta tietty CMM-taso voidaan saavuttaa, kustannuksia ja yrityksen avainprosessialueiden CMM:n mukaista räätälöintiä. Suurin osa yrityksistä uskoo, että CMM:n vaatimaa sisäistä koulutusohjelmaa ei ole mahdollista toteuttaa yrityksessä saatavilla olevilla resursseilla. Yritykset ovat myös sitä mieltä, että niiden omat räätälöidyt koulutusmallit voisivat olla riittäviä CMM:n tavoitteiden saavuttamiseen. Pienessä yrityksessä ei myöskään ole mahdollista perustaa CMM:n vaatimia erillisiä organisaatioita konfiguraation hallintaan tai laadun varmistukseen resurssien puutteen vuoksi. Pienissä yrityksissä konfiguraation hallinta ja laadunvarmistus hoidetaan siten, että työntekijät hoitavat erilaisia rooleja eri projekteissa. Lisäksi CMM vaatii yritykseen sellaisen hallinnollisen rakenteen, joka ei sovellu Brodmanin ja Johnsonin (1994) mukaan pienille yrityksille. CMM:n vaatimia hallinnollisia tasoja ei ole olemassa pienissä yrityksissä, joissa on matala organisaatio ja usein yrityksen johdolla on vastuullaan yrityksen hallinnon lisäksi myös projektien teknisiä osa-alueita aina ohjelmistojen toteutuksesta alkaen.

Brodmanin ja Johnsonin (1994) mukaan osa CMM:n ongelmista koskee myös suurempia yrityksiä. Pienillä yrityksillä on kuitenkin resurssien puutteen vuoksi enemmän vaikeuksia näiden ongelmien ratkaisussa. Pienet organisaatiot eivät myöskään aina löydä tarvitsemaansa apua ongelmien ratkaisuun, vaikka pienillä organisaatioilla isojen yritysten sisällä olisikin suuren yrityksen tuki takanaan. Brodmanin ja Johnsonin (1994) mukaan suuren yrityksen sisäiselle ohjelmisto-organisaatiolle antama tuki on kuitenkin usein hyvin rajoittunutta, varsinkin jos organisaatio toimii suuren yrityksen sisällä omassa kustannuspaikassaan. Vaikka pienillä yrityksillä onkin ongelmia ohjelmistotuotantoprosessin kehittämisessä, ne kuitenkin Brodmanin ja Johnsonin (1994) mukaan ymmärtävät, että yrityksen pienuus voi olla myös hyödyksi. Kun yrityksessä on päätetty alkaa kehittää ohjelmistotuotantoprosessia, pienen yrityksen kulttuuria ja toimintatapoja on



paljon helpompi muokata prosessin kehittämistä tukeviksi kuin suurissa yrityksissä. Myös pienille yrityksille tyypilliset pienet ja lyhytkestoiset projektit tukevat hyvin uusien toimintatapojen ja prosessien käyttöönottoa. Uusia käytäntöjä on helpompi ottaa käyttöön heti projektin alussa, kuin yrittää sovittaa niitä kesken kaiken jo pitkään käynnissä olleeseen suureen projektiin. Myös asiakasta saattaa olla vaikeampaa vakuuttaa uusien toimintatapojen käyttöönoton hyödyllisyydestä kesken projektin, koska asiakas on jo tottunut ja usein myös tyytyväinen olemassa oleviin käytäntöihin, kunhan projekti vain etenee sovitussa aikataulussa. Myös viestintä pienissä yrityksissä on helpompaa. Pienissä organisaatioissa yrityksen johto työskentelee kehittäjien rinnalla ja kehityksen ulkopuoliset toiminnot, kuten järjestelmien hallinta ja testaus, on integroitu samaan organisaatioon ja kommunikointi organisaation eri osa-alueiden välillä on päivittäistä.

Monet pienet yritykset toimivat suurten yritysten alihankkijoina. Brodmanin ja Johnsonin (1994) mukaan se aiheuttaa ongelmia myös yrityksen oman ohjelmistotuotantoprosessin kehittämisessä. Alihankkijana pienten yritysten on pakko noudattaa tärkeimpien asiakkaidensa käytäntöjä ja prosesseja, eivätkä ne voi vapaasti kehittää omia itselleen parhaiten sopivia toimintamalleja. Myös vertailukelpoisen historiatiedon kerääminen projekteista on vaikeaa, koska jokaisella pienen yrityksen tärkeällä asiakkaalla on yleensä erilaiset vaatimukset käytettävälle ohjelmistotuotantoprosessille ja erilainen tapa tuottaa ohjelmistoja. Eri asiakkaiden tarpeet ja vaatimukset ohjelmistotuotantoprosessille ovat erilaisia, joten on hankala muodostaa kaikkia projekteja koskevia yhtenäisiä käytäntöjä. Suurilla yrityksillä voi olla myös vastaavanlaisia haasteita, mutta niillä on yleensä useita projekteja samoilta asiakkailta, jolloin yhtenäisten projektikäytäntöjen luominen on helpompaa.

Brodmanin ja Johnsonin (1994) mukaan yrityksen koon mukaiset tekijät, jotka vaikuttavat ohjelmistotuotantoprosessin kehittämiseen ja CMM:n käyttämiseen ovat yhdistelmä projektien koosta, organisaation koosta ja koko yrityksen koosta. Projektin koon ollessa pieni, Brodmanin ja Johnsonin (1994) mukaan alle kaksikymmentä ihmistä, CMM:n projektikohtaiset käytännöt (esimerkiksi erilliset konfiguraationhallinta- ja laadunvarmistusorganisaatiot), eivät näytä olevan sovellettavissa. Ohjelmistoja tuottavan organisaation ollessa pieni (myös pienet suurten yritysten sisällä olevat organisaatiot), CMM:n organisaatiota koskevat käytännöt (esimerkiksi erilliset itsenäiset ryhmät ja yrityksen sisäinen koulutusohjelma), aiheuttavat ongelmia. Itse yrityksen ollessa pieni (osittain myös silloin kun kyseessä on erillinen omana kustannuspaikkanaan toimiva organisaatio suuren yrityksen sisällä), CMM:n aiheuttamat kustannukset ovat yleensäkin liian suuret. Brodmanin ja Johnsonin (1994) mukaan pienessä yrityksessä täysin CMM:n mukaisesta prosessin kehittämisestä saadut hyödyt eivät ole tarpeeksi suuret, jotta CMM:n aiheuttamat kustannukset voidaan yrityksen resursseilla maksaa tai perustella CMM:n käyttäminen yrityksessä.

Vaikka CMM:n käytössä pienissä yrityksissä onkin haasteita, niin Brodmanin ja Johnsonin (1994) mukaan useat pienet yritykset ja organisaatiot ovat onnistuneet suhteellisen hyvin ohjelmistotuotantoprosessinsa kehittämisessä CMM:n avulla. Usein yritysten käytännöt eivät ole täysin CMM:n mukaisia, mutta kuitenkin innovatiivisia ja toimivia. Brodmanin ja Johnsonin (1994) mukaan tekijöitä, jotka vaikuttavat yritysten ohjelmistotuotantoprosessin kehittämisen onnistumiseen ovat:

- Kokonaisvaltainen laatuun keskittyminen ja siihen panostaminen koko yrityksessä
- Asiakkaiden tuki yrityksen ohjelmistotuotantoprosessin kehittämisponnisteluille
- Yrityksen johdon sitoutuminen ohjelmistotuotantoprosessin ja sitä kautta yrityksen tuottamien ohjelmistojen laadun kehittämiseen
- Yhteistyö suuren yrityksen kanssa, jonka ohjelmistotuotantoprosessi on jo valmiiksi kypsemällä tasolla
- Yrityksen ulkopuolisen, prosessin kehittämisessä tukevan, organisaation jäsenyys

Vaikka useat yritykset eivät Brodmanin ja Johnsonin (1994) mukaan usko koskaan pääsevänsä CMM-tasoa kolme korkeammalle kypsyystasolle, niin monilla parhaiten menestyneillä yrityksillä on käytössään myös korkeampien kypsyystasojen käytäntöjä. Yleisesti laadukkaiden ohjelmistojen tuottamiseen pyrkiminen on luonut menestyville yrityksille monia kypsiä käytäntöjä, vaikka ne eivät välttämättä ole edes toisella CMM:n mukaisesti arvioidulla kypsyystasolla. Brodmanin ja Johnsonin (1994) mukaan yleensä suurimmat CMM:n mukaiset puutteet yrityksissä nähdään dokumentoinnissa ja ohjelmistotuotantoprosessin tarkassa määrittämisessä.

Brodmanin ja Johnsonin (1994) mukaan useat pienet yritykset haluavat parantaa ohjelmistotuotantoprosessiaan, mutta se ei ole mahdollista CMM:n puitteissa, koska pienet yritykset eivät järkevästi pysty täyttämään kaikkia CMM:n asettamia vaatimuksia. Prosessin kehittämisen vaatimukset ovat selkeitä myös pienille yrityksille, mutta ne tarvitsevat ohjausta vaatimusten täyttämiseen ja omien räätälöityjen käytäntöjensä hyväksymisen myös CMM:n kypsyystasojen mukaisiksi käytännöiksi. Lisäksi pienen yrityksen prosessin kehittäminen ei saa aiheuttaa niin suuria kustannuksia, että pieni yritys ei prosessin kehitysaikeistaan huolimatta ole valmis tai pysty niitä maksamaan.

### 3.2 Arviointimenetelmien räätälöinti pienten yritysten tarpeisiin

Johnson ja Brodman (1997) räätälöivät CMM:ää paremmin pienille yrityksille sopivaksi ja 82 prosenttia sen käytännöistä muokattiin pienten yritysten tarpeet huomioiden. Heidän mukaansa räätälöinti ei kuitenkaan radikaalisti muuta CMM:ää, mikä käy ilmi heidän listaamistaan räätälöinnissä tehtyjen muutosten erityispiirteistä. Tällaisia piirteitä ovat olemassaolevien käytäntöjen selkiyttäminen, tärkeiden asioiden korostaminen ja tarkempi selittäminen myös esimerkkien avulla, vaihtoehtoisten käytäntöjen esittely ja käytäntöjen sovittaminen pienten yritysten rakenteeseen ja resursseihin sopiviksi. Johnsonin ja Brodmanin (1997) mukaan pienille yrityksille räätälöintiä CMM:n osaluista vaativat erityisesti dokumentaatiovaatimukset, CMM:n määrittämät organisaation rakenne- ja resurssivaatimukset, katselmointivaatimukset, koulutusvaatimukset ja pienille yrityksille tarpeettomien käytäntöjen muokkaaminen tai poistaminen.

Coleman Danglen ym. (2005) mukaan CMM ei ole vain lista säännöksiä, vaan se määrittää viitekehyksen, jonka tarkoituksena on parantaa ohjelmistotuotantoprosessia riippumatta yrityksen koosta. Heidän mukaansa esimerkiksi CMM:n kolmannen tason saavuttaminen voi mahdollistaa pienelle yritykselle uusien asiakkaiden ja sopimusten hankkimisen. Myös monet asiakkaat vaativat tietyn CMM:n tason saavuttamista, jotta suostuvat ostamaan yritysten palveluita. Coleman Danglen ym. (2005) mukaan organisaatiot yleensä pitävät CMM:n kolmatta tasoa tärkeänä merkkipaaluina ohjelmistotuotantoprosessia kehittäessään ja monet organisaatiot jäävät prosessin kehittämisessä tälle tasolle. Heidän mukaansa osa yrityksistä kuitenkin haluaa saavuttaa myös CMM:n ylimmän tason. He toteavatkin, että CMM:n yksi heikkous on sen keskittyminen tasoihin, joille kullekin on määritetty tietyt avainprosessialueet, jotka on toteutettava saavuttaakseen kyseessä olevan CMM:n tason. Koska CMM:n mukainen prosessin kehittäminen on aikaa vievää, monet organisaatiot jättävät huomioimatta korkeampien CMM:n tasojen vaatimia avainprosessialueita, kun pyrkivät saavuttamaan vain jonkun alemman CMM:n tason. Coleman Danglen ym. (2005) mukaan tämä hidastaa huomattavasti kokonaisvaltaista edistymistä, kun organisaatiot toteuttavat CMM:ää vain taso kerrallaan. CMM esimerkiksi näkee prosessin mittaamisen tärkeänä vasta tasolla neljä ja tämän vuoksi yritykset eivät ala keräämään dataa prosessistaan riittävän alhaisella CMM:n tasolla ja korkeammalle CMM:n tasolle on vaikeampaa siirtyä, koska sen vaatimaa dataa ei ole kerätty. CMM:n uudistetussa CMMI versiossa tätä ongelmaa on huomioitu lisäämällä mittaamisen ja analysoinnin vaatimuksia tasolle kaksi. Coleman Dangle ym. (2005).

Coleman Danglen ym. (2005) mukaan jokaisen CMM:n avainprosessialueen saavuttaminen edellyttää prosessien määrittelyt, käytettyjen prosessien tulosten

tallentamisen ja todisteet avainprosessialueen saavuttamisesta. Myös avainprosessialueen saavuttamisen vaatimien uusien prosessien ja koulutuksen kustannukset on huomioitava ja prosesseja on valvottava, jotta voidaan todentaa, että organisaatio noudattaa avainprosessialueen vaatimuksia. Pienten yritysten ongelmana on, että niillä ei välttämättä ole tarvittavia voimavaroja muodostaa ryhmiä, jotka vastaavat organisaation ohjelmistotuotantoprosessin kehittämisestä. Coleman Dangelen ym. (2005) mukaan pienten yritysten on myös huomattava, että CMM:n avainprosessialueet ovat informatiivisia, joten niitä sovellettaessa on otettava huomioon organisaation koko ja sen vaatimukset. Heidän mukaansa voidaan myös esittää kysymys, että jos pieni organisaatio menestyy ja tuottaa voittoa eikä seuraa mitään CMM:n avainprosessialuetta, niin onko mitään syytä aloittaa organisaatiossa CMM:n mukaista ohjelmistotuotantoprosessin kehitysohjelmaa?

Coleman Dangle ym. (2005) löysivät prosessin kehittämisessä avainkohtia, joihin he etsivät ratkaisuja. Näitä ovat prosessin avainnäkökulmien löytäminen, resurssien osoittaminen, prosessihistoriatiedon tallennus, säännölliset katselmoinnit ja terminologian ja prosessimääritysten standardointi. He löysivät ohjelmistotuotantoprosessiin liittyviä ja prosessin ulkopuolisia asioita, jotka vaikuttavat yrityksen kehittämiseen ja menestymiseen. Heidän mukaansa prosessiin liittyviä asioita ovat sen ymmärtäminen, että prosessia ei kehitetä vain prosessin vuoksi, vaan sen kehittämisessä täytyy pitää aina mielessä myös yrityksen liiketoimintatavoitteet. He jatkavat, että prosessia pitää kehittää tarpeiden mukaan, eikä tuijottaa sokeasti CMM:n asettamia avainprosessialueita. Coleman Dangelen ym. (2005) mukaan prosessin kehittäminen ei myöskään onnistu ilman sille suoraan osoitettuja resursseja. He toteavat myös, että uudet käytännöt kannattaa ensin testata pilottiprojekteissa ja laajentaa ne sen jälkeen koskemaan koko organisaatiota ja kaikkia projekteja. Uusien toimintatapojen käyttöönotto pitää myös suunnitella etukäteen ja ottaa huomioon yrityksen nykyhetken tilanne. Coleman Dangle ym. (2005) huomasivat myös, että kun yritys kehittää ohjelmistotuotantoprosessiaan, sen organisaation on muututtava samalla tukemaan uutta toimintamallia ja se johtaa selkeämpää ja hallitumpaan organisaatioon. Coleman Dangle ym. (2005) mukaan yrityksen ohjelmistotuotantoprosessia kehitettäessä on syytä ottaa huomioon seuraavat asiat:

- Prosessin kehitystä voidaan nopeuttaa, jos siihen on määrätty henkilöitä, joilla on suora vastuu prosessin kehittämisen etenemisestä
- Prosessin kehittämiseen on varattava riittävästi resursseja
- Muodolliset katselmoinnit prosessin kehittämisen etenemisestä kannattaa aloittaa heti, jolloin yrityksen johto saadaan myös heti sitoutettua prosessin kehittämiseen ja seuraamiseen

- Prosessia kehittävää henkilöstöä on koulutettava ja heidän osaamistasoaan seurattava, jotta prosessin kehitys etenee, eikä pysähdy henkilöstön vajavaiseen prosessiosaamiseen

Richardsonin ja Gresse von Wangenheimin (2007) ja Gresse von Wangenheimin, Anacleton ja Salvianon (2006) mukaan pienille yrityksille räätälöityjä uusia arviointimenetelmiä on kehitetty viime vuosina aktiivisesti. Heidän mukaansa tällaisia ovat esimerkiksi Rapid Assessment for Process Improvement for Software Development (RAPID), Software Process Improvement Initiation (SPINI), Método de Avaliação de Processo de Software (MARES), Fraunhofer Assessment Method (FAME), Toward Organized Processes in SMEs (TOPS) ja Software Process Improvement in Regions of Europe (SPIRE). He toteavat, että nämä menetelmät keskittyvät yleensä prosessin parantamisen arviointiin ja CMM:n kolmannen tason saavuttamiseen.

## 4 PIENTEN OHJELMISTOALAN YRITYSTEN OHJELMISTOTUOTANTOPROSESSIN KEHITTÄMINEN

Dybån (2000) mukaan ohjelmisto-organisaatiot voivat noudattaa kahta löyhästi määritettyä parannusstrategiaa. Ne voivat hyödyntää riistostrategiaa, jossa olemassaolevaa tietämystä ja kokemusta pyritään hyödyntämään maksimaalisesti tai tutkimusstrategiaa, jossa pyritään etsimään uutta tietämystä joko kopioimalla tai kehittämällä uutta. Hänen mukaansa perustason ongelmana on, kuinka hyödyntää riittävästi riistostrategiaa saavuttaakseen lyhyen tähtäimen tulokset, mutta toisaalta taas kuinka samaan aikaan tutkia uutta, jotta myös pitkän tähtäimen menestyminen on mahdollista. Dybån (2000) mukaan organisaatio, joka keskittyy vain riistostrategiaan jämähtää käyttämään jatkuvasti vanhemmaksi käyvää teknologiaa kun taas pelkästään uuden etsimiseen keskittyvä organisaatio ei koskaan ymmärrä omien löytöjensä koko potentiaalia. Hänen mukaansa pienet ohjelmistoalan organisaatiot käyttävät toimintaolosuhteista huolimatta suunnilleen saman verran riistostrategiaa, mutta epävakaisissa olosuhteissa pienet organisaatiot käyttävät huomattavasti enemmän resursseja uuden tutkimiseen kuin vakaisissa oloissa. Dybån (2000) mukaan ohjelmistoja tuottaessa odottamattomat ongelmat ja epäonnistumiset on pystyttävä kääntämään oppimismahdollisuuksiksi.

Gresse von Wangenheim, Anacleton ja Salvianon (2006) mukaan ensimmäinen askel ohjelmistotuotantoprosessin kehittämiseen on organisaation prosessin nykytilan ja sen vahvuuksien ja heikkouksien selvittäminen. Heidän mukaansa vasta tämän jälkeen voidaan määrittää tehokkaita prosessin parannustoimia. Gresse von Wangenheim ym. (2006) toteavat, että arviointimenetelmät voivat auttaa organisaatiota vertaamaan sen ohjelmistotuotantoprosessia viiteprosessimalliin ja selvittämään nykyisen prosessin toimivuuden ja kypsyyden. Tätä selvitystä voidaan käyttää hyväksi, kun tutkitaan kuinka prosessin avulla voidaan saavuttaa halutut laatu, kustannus ja aikatauluvaatimukset.

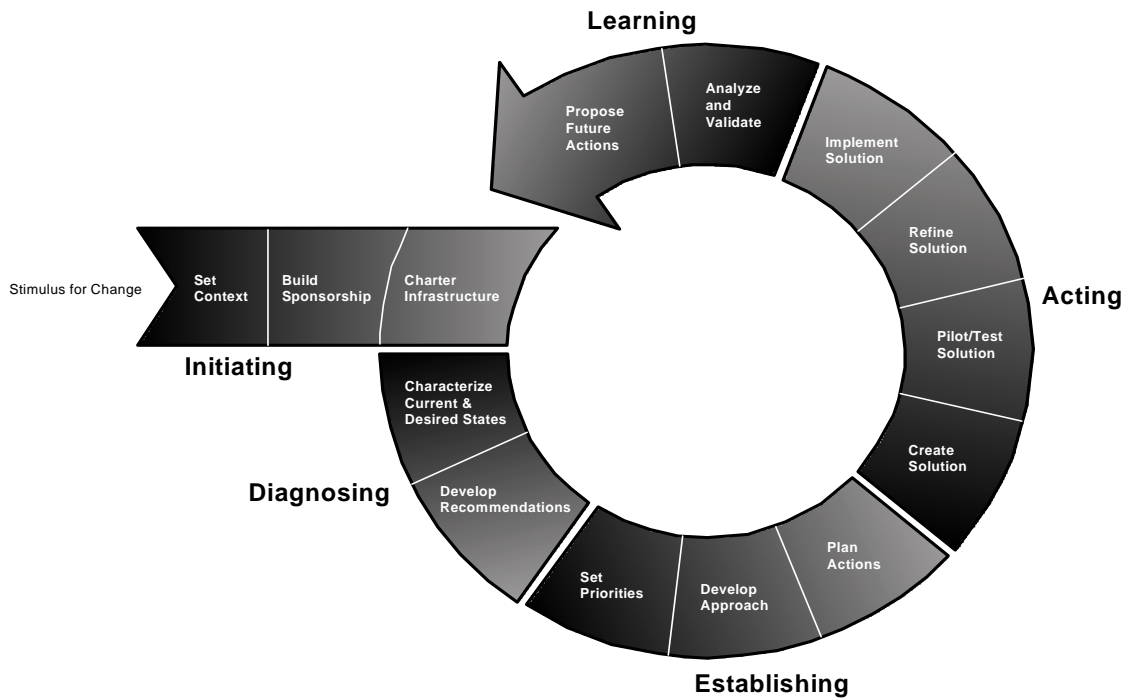
Guerreron ja Eterovicin (2004) mukaan haasteena pienen yrityksen ohjelmistotuotantoprosessin kehittämisessä on ymmärtää organisaation toimintaympäristö ja sen suhde prosessin kehittämiseen ja kuinka omassa ympäristössä prosessia voidaan kehittää. He jatkavat edelleen, että ymmärryksen puute voi estää tai hidastaa prosessin kehitystä, jos organisaatiossa ollaan sitä mieltä, että se on liian pieni käyttääkseen formaaleja malleja tai standardeja. Heidän mukaansa organisaation pieni koko saattaa kuitenkin helpottaa joitakin ohjelmistotuotantoprosessin kehittämisen vaiheita.

Dybå (2003) tutki 120 suurta ja pientä yritystä Norjassa ja hänen mukaansa pienet yritykset kehittävät prosessejaan formaalein keinoin yhtä tehokkaasti kuin suuretkin yritykset. Software Engineering Instituten (SEI) (2000, 2004,

2006) raporttien mukaan ohjelmistotuotantoprosessiaan kehittävien pienten yritysten määrä kasvoi yli kolminkertaiseksi vuosien 1999 ja 2003 välillä. SEI:n raporteissa pieniksi ohjelmistoalan yrityksiksi katsottiin yritykset, joiden ohjelmistotuotantoyksikössä oli 25 tai vähemmän työntekijöitä.

Dybån (2003) mukaan ohjelmistotuotantoprosessin kehittymistä mitattaessa on kuitenkin otettava huomioon, että sekä menestyminen ja menestymättömyys saattavat olla itseään tehostavia. Eli menestyvällä organisaatiolla on paremmat mahdollisuudet menestyä jatkossakin kuin huonommin menestyvällä organisaatiolla. Hänen mukaansa myös asetetuista tavoitteista jääminen tehostaa jatkossa organisaation pyrkimystä tavoitteisiin pääsemiseen ja siten myös organisaation mahdollisuuksia menestyä. Vastaavasti jatkuva menestyminen ja tavoitteiden saavuttaminen voi vähentää kehittymishalukkuutta ja heikentää toiminnan tehokkuutta, joka taas saattaa johtaa siihen, että menestystä ei enää tulekaan.

Pienen ohjelmistoalan yrityksen ohjelmistotuotantoprosessin kehittämisen käynnistäminen voi viedä monta vuotta. Guerreron ja Eterovicin (2004) tutkimuksessa yrityksen johto pohti prosessin kehittämisen aloittamista viisi vuotta ennen kuin prosessia alettiin kehittää. Päivittäisestä liiketoiminnasta aiheutuva kuorma esti resurssien osoittamisen prosessin kehittämiseen. Lopulta Guerreron ja Eterovicin (2004) tutkimassa yrityksessä valittiin IDEAL malli (KUVIO 3) ohjaamaan prosessin kehityksen läpivientiä (Myös Mc Caffery, Taylor ja Coleman (2007) kehittivät oman Adept arviointimenetelmänsä IDEAL-mallin Diagnosing osuutta varten).



KUVIO 3. IDEAL-malli (Software Engineering Institute 2007)

Guerrero ja Eterovic (2004) listaavat kymmenen oleellista tekijää, jotka vaikuttavat yrityksen prosessin kehittämiseen ja jakavat ne kahteen eri luokkaan: ympäristöstä riippuvaisiin ja riippumattomiin tekijöihin. Ympäristöstä riippuvaiset tekijät ovat sellaisia prosessin kehittämiseen vaikuttavia tekijöitä, jotka riippuvat yrityksen henkilöstöstä ja prosessiin panostetuista resursseista. Vastaavasti ympäristöstä riippumattomat tekijät ovat sellaisia, joihin vaikuttaa vain asiasta tehty yrityksen johdon päätös ja asian tiedostaminen.

Guerreron ja Eterovicin (2004) mukaan ympäristöstä riippuvaisia tekijöitä ohjelmistotuotantoprosessia kehitettäessä ovat:

- Prosessia koskeva koulutus
- Yrityksen työntekijöiden sitoutuminen prosessin kehittämiseen
- Prosessin kehittämisen jatkuvuuden ylläpito
- Ryhmäkeskeinen prosessin kehittäminen
- Prosessin kehittämisen avainhenkilöt ("mestarit")
- Prosessin kehittämisen mittaamisen tiheys
- Prosessin kehittämisen näkyvyys



Heidän mukaansa ympäristöstä riippumattomia tekijöitä taas ovat:

- Johdon sitoutuminen
- Organisaation oman kulttuurin tiedostaminen ja huomioon ottaminen
- Prosessi ja lopputuotenäkökulmien erottaminen

Ohjelmistotuotantoprosessia koskeava koulutus on tärkeää ja pienessä yrityksessä on mahdollista kouluttaa yrityksen kaikki työntekijät prosessin kehittämiseen liittyvissä asioissa. Jotta ohjelmistotuotantoprosessia voidaan onnistuneesti kehittää, on yrityksen työntekijät sitoutettava prosessin kehittämiseen. Pienen yrityksen työntekijät voidaan sitouttaa prosessin kehittämiseen antamalla heille heitä itseään kiinnostavia tehtäviä prosessin kehityksessä. Jotta ohjelmistotuotantoprosessin kehitykseen saadaan jatkuvuutta, kerätään sitä käyttäviltä työntekijöiltä jatkuvasti palautetta prosessista ja prosessia kehitetään saadun palautteen mukaan. Ryhmäkeskeisellä prosessin kehittämällä saadaan sitoutettua monia työntekijöitä prosessin parantamiseen ja prosessin kehittämisen onnistuminen ei ole vain yhden työntekijän vastuulla. Guerreron ja Eterovicin (2004) mukaan onkin järkevää, että prosessin eri osa-alueiden kehittäminen annetaan pienten ryhmien tehtäväksi sen sijaan, että se olisi määrätty ainoastaan jollekin yksittäiselle työntekijälle. Yrityksen projektiryhmistä pyritään tunnistamaan niissä mahdollisesti olevat prosessin kehittämisen avainhenkilöt, jotka pystyvät omalla esimerkillään vakuuttamaan muut ryhmäläiset prosessin kehittämisen hyödyistä. Näin prosessi saadaan yrityksessä mahdollisimman tehokkaasti käyttöön ja sen hyödyt nopeasti näkyviin. Guerreron ja Eterovicin (2004) mukaan pienissä yrityksissä toteutettavat suhteellisen lyhytkestoiset projektit mahdollistavat prosessin kehityksen mittaamisen muutaman kuukauden välein, jolloin prosessin kehittymisestä saadaan jatkuvaa tietoa. Prosessin kehittymisen mittaaminen on tärkeää senkin vuoksi, että konkreettisesti nähdään heti kehittämisestä saadut hyödyt ja löydetään prosessista uusia kehittämiskohteita. On myös tärkeää, että ohjelmistotuotantoprosessin kehittämiseen määritetään useita välitavoitteita ja prosessin kehittymistä ja saavutettuja parannuksia raportoidaan säännöllisesti jokaisen välitavoitteen yhteydessä. Tällä tavalla prosessin kehittämiseen saadaan näkyvyyttä.

Guerreron ja Eterovicin (2004) mukaan yrityksen johdon on oltava sitoutunut prosessin kehittämiseen. Ilman johdon sitoutumista prosessin kehittäminen ei onnistu eikä sille saada tarvittavia resursseja ja varoja. Johdon voimakas sitoutuminen helpottaa myös työntekijöiden sitouttamista prosessin kehittämiseen. Ohjelmistotuotantoprosessia kehitettäessä on otettava huomioon myös organisaation kulttuuri. Organisaatioon ei voida ajaa väkisin läpi muutoksia ja toimintatapoja, jotka ovat täysin olemassa olevan organisaatiokulttuurin vastaisia. Ohjelmistotuotantoprosessin kehittäminen on

myös erotettava varsinaisesta ohjelmistotuotteiden kehityksestä asiakkaille, jotta taataan prosessin kehittämisen onnistuminen. Yleensä ohjelmistoalan yrityksille aina tärkeintä on valmiiden ja toimivien ohjelmistojen ja ratkaisujen toimittaminen asiakkaille. Tämän ei kuitenkaan saisi antaa viedä aikaa ja resursseja prosessin kehittämisestä, koska paremmalla prosessilla myös asiakkaille pystytään toimittamaan parempaa laatua.

Guerreron ja Eterovicin (2004) mukaan prosessin kehittämiseen liittyviin toimintoihin on osoitettava selkeä ja näkyvä vastuu jollekin henkilölle tai henkilöille. Heidän mukaansa ohjelmistotuotantoprosessia ei pystytä kehittämään, jos kukaan ei ole vastuussa sen kehittämisestä, eikä prosessi myöskään kehity itsestään päivittäisten työtehtävien yhteydessä. Pienillä yrityksillä ei yleensä ole tarpeeksi työntekijöitä jokaisen työtehtävän eriyttämiseen, vaan yhdellä työntekijällä on useita eri työtehtäviä ja rooleja organisaatiossa. Tämän vuoksi pienessä organisaatiossa eri näkökohtien erottaminen prosessia kehitettäessä on vaikeaa. Guerrero ja Eterovic (2004) toteavat vielä, että heidän listaamiaan ympäristöstä riippumattomia tekijöitä ei voi olla huomioimatta ohjelmistotuotantoprosessia kehitettäessä, vaan jokaisen organisaation, joka kehittää prosessiaan, on tiedostettava yllä kuvatut ympäristöstä riippumattomat tekijät.

#### **4.1 Ohjelmistotuotantoprosessin mallintaminen**

Ahosen, Junttilan ja Taskisen (2003) mukaan jokaisella ohjelmistolla on elinkaari. Heidän mukaansa ohjelmistotuotantoprosessi- ja vaihejakomallit määrittävät tavan, jolla ohjelmiston elinkaari jaetaan vaiheisiin. Boehmin (1988) mukaan ohjelmistotuotantoprosessimallin tarkoituksena on määrittää ohjelmistotuotannon ja ohjelmiston elinkaaren vaiheiden järjestys ja selvittää minkä kriteerien perusteella vaiheesta toiseen prosessissa edetään. Hänen mukaansa vaiheesta toiseen siirryttäessä on tarkasteltava parhaillaan suoritettavana olevan vaiheen päättymiskriteerejä ja seuraavan vaiheen valinta- ja aloituskriteerejä. Boehm (1988) toteaa, että ohjelmistotuotantoprosessi eroaa ohjelmistotuotantomenetelmästä siinä, että menetelmässä keskitytään siihen kuinka yksittäinen ohjelmistotuotantoprosessin vaihe suoritetaan, kun taas prosessissa ydinkysymyksinä ovat: Mikä vaihe tehdään seuraavaksi, eli millä kriteereillä seuraava vaihe valitaan ja minkä kriteerien täytyttyä se voidaan aloittaa ja kuinka kauan seuraavaa vaihetta tehdään, eli mitkä ovat sen päättymiskriteerit. Boehmin (1988) mukaan ohjelmistotuotantoprosessit ovat tärkeitä, koska ne ohjaavat ohjelmistotuotantoprojekteja ja määrittävät missä järjestyksessä projektien tärkeimmät vaiheet tulee viedä läpi. Curtis, Kellner ja Over (1992) kuvaavat ohjelmistoprosessin mallintamisen kohteita ja tavoitteita (TAULUKKO 1).

TAULUKKO 1. Ohjelmistotuotantoprosessin mallintamisen kohteet ja tavoitteet (Curtis ym. 1992).

<p><b>Tavoitteena henkilöstön ymmärryksen ja kommunikoinnin parantaminen</b></p> <ul style="list-style-type: none"> <li>• Esitä prosessi yleisesti ymmärrettävässä muodossa</li> <li>• Lisää kommunikaatiota ja yhteisymmärrystä ohjelmistotuotantoprosessissa</li> <li>• Formalisoi prosessi tehokkuuden lisäämiseksi</li> <li>• Tarjoa riittävästi tietoa yksilöille tai tiimeille prosessin suorittamiseksi</li> <li>• Muodosta perusta prosessikoulutukselle</li> </ul>
<p><b>Tavoitteena prosessin kehittämisen tukeminen</b></p> <ul style="list-style-type: none"> <li>• Tunnista kaikki ohjelmistotuotannon tai ylläpidon prosessien välttämättömät komponentit</li> <li>• Uudelleenkäytä hyvin määriteltyjä ja tehokkaita prosesseja tulevissa projekteissa</li> <li>• Vertaile vaihtoehtoisia prosesseja</li> <li>• Arvioi mahdollisten muutosten vaikutusta ennen niiden toteuttamista käytännössä</li> <li>• Tue prosessia uudella teknologialla</li> <li>• Mahdollista organisaatiotason oppiminen tehokkaista prosesseista</li> <li>• Tue prosessin hallittua arviontia ja kehittämistä</li> </ul>
<p><b>Tavoitteena prosessin hallinnan tukeminen</b></p> <ul style="list-style-type: none"> <li>• Kehitä projektikohtainen ohjelmistotuotantoprosessi ominaisuuksien sisällyttämiseksi</li> <li>• Tarkastele ohjelmiston luomisen tai kehittämisen syitä</li> <li>• Tue projektikohtaisten suunnitelmien tekemistä</li> <li>• Valvo, hallinnoi ja koordinoi prosessia</li> <li>• Muodosta perusta prosessin mittaamiselle</li> </ul>
<p><b>Tavoitteena automatisoitu ohjeistus prosessin suorittamiselle</b></p> <ul style="list-style-type: none"> <li>• Määrittele tehokas ohjelmistotuotantoympäristö</li> <li>• Tarjoa ohjeistusta, ehdotuksia ja viitemateriaalia ohjelmistotyön helpottamiseksi</li> <li>• Säilytä uudelleenkäytettävät prosessiesitykset</li> </ul>
<p><b>Tavoitteena automasoitu suorittamisen tuki</b></p> <ul style="list-style-type: none"> <li>• Automatisoi prosessin osa-alueita</li> <li>• Tue yksilöiden ja tiimien välistä yhteistyötä automatisoimalla prosessia</li> <li>• Kerää mittaustietoa automaattisesti ja vertaa tuloksia prosessin käytännön havainnointiin</li> <li>• Valvo sääntöjä prosessin eheyden varmistamiseksi</li> </ul>

Kun monta henkilöä työskentelee yhdessä saman tavoitteen saavuttamiseksi, tarvitaan selkeät määrittelyt tehtävien suorittamisesta ja niiden järjestyksestä. Prosessimallit määrittävät prosessin vaiheet, suhteet ja riippuvuudet ja organisaation olemassaolevia prosessimalleja voidaan selvittää mallintamalla. Ahosen, Junttilan ja Taskisen (2003) mukaan prosessimallinnus sisältää viisi päätehtävää:

- Kommunikaation mahdollistaminen ihmisten välillä ja yhteenkuuluvaisuuden tunteen luominen
- Prosessin kehityksen ja parantamisen tukeminen
- Prosessin hallinnan tukeminen
- Prosessin kontrolloinnin automatisoiminen
- Prosessin toteutuksen tukemisen automatisoiminen

Curtisin, Kellnerin ja Overin (1992) mukaan prosessimallien päätavoite on varmistaa että prosessit pystyisivät mahdollisuuksien mukaan mukautumaan todelliseen työnkulkuun eivätkä välttämättä niinkään toista korkeamman tason vaatimuksia, jotka tosielämässä saattavat osoittautua turhiksi. Cook ja Wolf (1999) esittävät, että suurimmat hyödyt saavutetaan, kun prosessimalleja voidaan käyttää todellisen työnkulun ennustamiseen. Kun käytetään jotain formaalia menetelmää, niin lisähyötyjä saavutetaan vertailemalla toteutuneita projekteja, suorituskykyä ja saavutuksia. Tämä onkin usein välttämätöntä, jos halutaan parantaa ohjelmistotuotantoprosessia; prosessi täytyy mallintaa ja kuvata, jos sitä halutaan mitata, analysoida ja parantaa. Eri malleilla on erilaisia painotuksia ja jokaisella on omat hyvät ja huonot puolensa. (Cook & Wolf, 1999). Oikeaa prosessimallia valittaessa on myös muistettava että yrityksen koko ja käytettävissä olevat resurssit sekä työntekijäiden taidot ja käytössä olevat välineet vaikuttavat asiaan oleellisesti (Humprey 1989).

#### **4.2 Pienten yritysten ohjelmistotuotantoprosessin kehittämisen arviointia**

Laitisen, Fayadin ja Wardin (2000) mukaan ohjelmistotuotannossa toteutettavat kohteet kasvavat suuremmiksi ja suuremmiksi, toisin kuin perinteisessä insinööritaidossa. Viisi vuosisataa sitten rakennettiin siltoja, jotka olivat suunnilleen samankokoisia kuin nykyäänkin rakennetut sillat. Vain työkalut, materiaalit ja rakennustekniikka ovat kehittyneet. Laitisen ym. (2000) mukaan ohjelmistotuotannossa hyvin harvojen projektien suuruusluokka pysyy vakiona, joten ajan kuluessa on hankalaa vertailla eri ohjelmistotuotantoprojekteissa käytettyjä menetelmiä ja työkaluja.

Laitisen ym. (2000) mukaan ohjelmistotuotannon kirjallisuus yleensä vain kehottaa skaalaamaan olemassaolevat menetelmät alaspäin pieniin projekteihin

ja rajoitetuille resursseille sopiviksi, jos se yleensäkin antaa mitään ohjeita erikseen pienille yrityksille. Heidän mukaansa tämä on johtanut muutamiin pienille yrityksille huonosti sopiviin suosituksiin. He toteavat, että pelkkä alaspäin skaalaaminen ei yleensä ole pienille yrityksille sopiva tapa, koska suurille yrityksille tehdyt menetelmät ja ohjelmistotuotantoprosessit eivät tue tai dokumentoi sitä kunnolla. Jos suurille yrityksille tehtyjä menetelmiä ja prosesseja skaalataan alaspäin pienemmälle ryhmälle, pitää ryhmän jäsenten hoitaa useampia rooleja. Tämä ei kuitenkaan ole järkevää, koska osa rooleista ja tehtävistä on tarpeettomia pienemmän kokoluokan projekteissa. Laitisen ym. (2000) mukaan alaspäin skaalaaminen ei voi tarkoittaa sitä, että kaikki suuria projekteja varten tarvittavat tehtävät säilytetään pienissäkin projekteissa ja tehdään tehtävistä vain jollain tavalla pienempiä, tai että pienempi määrä ihmisiä hoitaa menetelmän käyttämisestä aiheutuvan suuremman työtaakan vain pelkän menetelmän vaatimusten vuoksi.

Laitinen ym. (2000) listaavat ongelmia, joita ohjelmistotuotannossa tulee alaspäin skaalaamisesta:

- Ohjelmistotuotannon menetelmät eivät huomioi pieniä yrityksiä riittävän hyvin
- Joitakin menetelmiä ei yksinkertaisesti voi skaalata pienemmiksi
- Raskaiden menetelmien käyttämisen kustannusten osuus projektista kasvaa liian suureksi, jos itse projekti on riittävän pieni
- Tarvittavien työkalujen hankkiminen ja käyttämisen oppiminen voi olla liian iso ponnistus pienille projekteille

Laitisen ym. (2000) mukaan projektiryhmien kokoa ei kuitenkaan tulisi kasvattaa sen varjolla, että suuri osa menetelmistä on suunnattu suuremmille projekteille. Heidän mukaansa on osoitettu, että pienet ryhmät ovat suuria ryhmiä tehokkaampia ja pienillä ryhmillä on yleensä suurempi tuotto yhtä henkilöä kohden kuin isoilla ryhmillä. He toteavat myös, että vaikka monet ohjelmistotuotannon menetelmät on todettu yleisesti toimiviksi, niiden kokonaistoimivuutta pienten ryhmien osalta ei ole osoitettu optimaaliseksi.

Wardin, Fayadin ja Laitisen (2001) mukaan pienet ohjelmistoalan yritykset, jotka haluavat parantaa ohjelmistotuotantoprosessiaan, kohtaavat kaksi haastetta: Ensimmäinen haaste on saada yritys yleensäkin ottamaan käyttöön jonkinlaista ohjelmistotuotantoprosessia ja alkaa kehittämään sitä. Toinen haaste on pienissäkin yrityksissä jatkuvasti käynnissä oleva taistelu olemassaolevan prosessin kehittämisestä vastaamaan muuttuvia olosuhteita.

Ward ym. (2001) toteavat, että tehokkain koodin katselmointikäytäntö pienelle kolmen tai neljän hengen toteutustiimille projektin alkuvaiheessa voi olla

spontaani keskustelu piirustustaulun äärellä. Heidän mukaansa tällaiseen "taulukokoontumiseen" täysin riittävä prosessi voi olla niinkin yksinkertainen kuin näiden kokoontumisten optimaalisen aikataulun sopiminen (esimerkiksi kerran viikossa), kaikkien ohjelmoijien koodin läpikäynnin varmistaminen ja kokoontumisen koolle kutsuvan vastuuhenkilön määrääminen. He toteavatkin, että tavoitteiden ja riskien ymmärtäminen on tärkeää oikeiden ohjelmistotuotantoprosessien valinnassa. Heidän mukaansa prosessit kuitenkin lopulta ovat vain sopimuksia siitä, kuinka vaikea työ saadaan tehtyä menestyksekkäästi ja onnistuneesti.

Prosessin kehitys suuressa yrityksessä tähtää tehokkuuden ja luotettavuuden kasvattamiseen ja samanaikaiseen kustannusten vähenemiseen. Jotkin suuret yritykset ovat raportoineet jopa kustannusten pienemistä seitsemänteen osaan alkuperäisestä ohjelmistotuotantoprosessin parantamisen avulla. Tällaiset numerot ovat kuitenkin yleensä merkityksettömiä pienille yrityksille. Pienessä kasvavassa yrityksessä ei luultavasti ole niin suurta toimintojen päällekkäisyyttä, että suurissa yrityksissä saavutettujen kaltaiset säästöt olisivat mahdollisia. Pienet yritykset ovatkin paljon kiinnostuneempia kasvusta aiheutuvan kaaoksen hallitsemisesta parempien prosessien avulla. (Ward ym. 2001).

Wardin ym. (2001) mukaan pienillä yrityksillä on nykyään mahdollisuus valita oma ohjelmistotuotantoprosessinsa monista hyvistä jo valmiista prosesseista tai kehittää oma versionsa yleisesti hyväksytyistä prosesseista. Heidän mukaansa formaalien prosessien vertaamista asiayhteyden ulkopuolella tietyn ryhmän tiettyyn tilanteeseen on kuitenkin vältettävä. He uskovatkin, että abstraktit rinnakkaisvertailut ohjelmistotuotantoprosessimallien välillä ovat merkityksettömiä. Menestyksekkäiden prosessien ongelma on, että ne aiheuttavat yrityksessä muutoksen. Yrityksen menestyminen prosessia hyödyntämällä voi aiheuttaa yrityksen työntekijöiden määrän kasvamisen, sen tuotteiden monipuolistumisen, uusia asiakkaita esimerkiksi ulkomailla, tiukempia julkaisuaikatauluja tai melkein minkä tahansa menestyksen mukanaan tuoman muunlaisen muutoksen. Muutoksesta johtuen menestymisen mahdollistaneet prosessit kuitenkin todennäköisesti vaativat kehittämistä, jotta ne pystyvät edelleen ohjaamaan toimintaa tehokkaasti muuttuneessa tilanteessa. Muutoksenhallinta onkin olennainen osa ohjelmistotuotantoprosessin kehitystä. (Ward ym. 2001).

Wardin ym. (2001) mukaan ohjelmistotuotantoprosessin kehittämisessä keskeisiä asioita ovat:

- Pelkkä prosessi ei tee organisaatiosta tehokasta. Prosessi on työkalu, ei päämäärä
- Prosessin tulee olla yksinkertainen

- Prosessin tulee olla helposti käytettävä ja auttaa minimoimaan väärinkäyttämisen mahdollisuudet

Ward ym. (2001) toteavat, että pienissä yrityksissä hyödynnetyt prosessit eivät luultavasti ole ihanteellisia katsottiinpa niitä kriitikkojen tai niiden käyttäjien kannalta. Rajoitetut resurssit ja työntekijöiden pieni määrä eivät tue raskaiden formaalien ohjelmistotuotantoprosessimallien käyttämistä, vaan pienten yritysten tulisi suosia keveitä ja helposti muokattavia prosessimalleja. Sen sijaan, että pienissä yrityksissä yritettäisiin pilkuntarkasti noudattaa jotain abstraktilla tasolla ihanteellista formaalia mallia, pitää prosessien tukea organisaatiota ja sen kasvua mahdollisimman hyvin ja kehittyä organisaation mukana sen kasvaessa.

## 5 YHTEENVETO

Pienet ohjelmistoalan yritykset ovat nykyään tärkeässä roolissa monen maan ohjelmistotuotannossa ja niissä tuotetaan laadukkaita ja yhä monimutkaisempia ohjelmistoja. Jotta pienet yritykset voivat yhä edelleen parantaa ja tehostaa toimintaansa, täytyy myös niiden ohjelmistotuotantoprosessia ja niille sopivia malleja tutkia tarkemmin. On osoitettu, että monet suurille yrityksille suunnitellut mallit ja prosessit eivät suoraan sovellu pienten yritysten käyttöön. Suurten yritysten malleja täytyy räätälöidä ja kehittää, jotta myös pienet yritykset saavat niistä mahdollisimman suuren hyödyn. Toinen vaihtoehto on kehittää erityisesti pienille yrityksille suunniteltuja malleja ja prosesseja. Niissä pienten yritysten erityispiirteet ja vahvuudet pyritään ottamaan huomioon, mutta samalla tehostamaan ohjelmistotuotantoa ja parantamaan laatua.

Pienillä yrityksillä on yleensä sellaisia erityispiirteitä, jotka tekevät niistä yksilöllisiä. Nämä piirteet pitää ottaa kunkin pienen yrityksen kohdalla huomioon erikseen, jotta yritysten prosessia saadaan parhaiten kehitettyä. Yksi tärkeimpiä kohtia pienten yritysten prosessin kehittämässä on yrityksen johdon ja työntekijöiden sitouttaminen kehitykseen. Monesti menestyvälle ja voittoa tuottavalle organisaatiolle voi olla hankalaa perustella miksi senkin tulisi alkaa tutkimaan ja kehittämään ohjelmistotuotantoprosessiaan. On kuitenkin osoitettu, että prosessia kehittämällä ja parantamalla, saadaan varmistettua menestymisen jatkuvuutta ja tehostettua organisaation toimintaa.

Pienten yritysten ohjelmistotuotantoprosessit olivat pitkään kirjallisuudessa sivulauseen asemassa. Nykyään niitä on kuitenkin tutkittu enemmän ja myös erityisesti pienille yrityksille suunnattuja malleja on kehitetty viime vuosina. Aihetta on kuitenkin syytä tutkia lisää ja löytää malleja, jotka parhaiten tukevat erilaisten ja erikokoisten organisaatioiden toimintaa.

Pienille yrityksille suunnattujen ohjelmistotuotantoprosessien tulisi täyttää ainakin seuraavat kriteerit:

- Prosessin pitää olla kevyt
- Prosessin pitää olla helposti muokattavissa
- Prosessin pitää olla helppo omaksua
- Prosessi pitää voida ottaa käyttöön osa-alue kerrallaan organisaation tarpeen mukaan
- Prosessin tulee pystyä mukautumaan toimintaympäristön nopeisiin muutoksiin



- Prosessin käyttäminen ja käyttöönottoaminen ei saa aiheuttaa suurta kustannusta organisaatiolle

Ohjelmistotuotantoprosessit ovat tärkeitä myös pienissä yrityksissä. Pieniä yrityksiä, niiden prosesseja ja niissä käytettyjä malleja tulisi kuitenkin tutkia lisää, jotta myös tämän alati kasvavan ohjelmistojen tuottavan organisaatioiden ryhmän toiminta saataisiin mahdollisimman tehokkaaksi ja laadukkaaksi.

## LÄHDELUETTELO

- Ahonen J.J., Junttila T. & Taskinen S-K. 2003. An Industry Proven Software Engineering Process Model for Small Companies. Proceedings of 21st IASTED International Conference on Applied Informatics, IASTED AI'2003.
- Boehm B. W. 1988. A Spiral Model of Software Development and Enhancement. IEEE Computer 21(5), 61-72.
- Brodman J. G., Johnson D. L. 1994. What Small Businesses and Small Organizations Say About the CMM, Experience Report. Proceedings of the 16<sup>th</sup> International Conference on Software Engineering, Sorrento, Italy, May 16-21, 331-340.
- Brooks F. P. Jr. 1995. The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition. Reading, Massachusetts: Addison-Wesley Longman, Inc.
- Cattaneo F., Fuggetta A., Lavazza L. 1995. An Experience in Process Assessment. Proceedings of the 17<sup>th</sup> International Conference on Software Engineering, Seattle, MD USA, April 24-28, 115-121.
- Coleman Dangle K., Larsen P., Shaw M., Zelkowitz M.V. 2005. Software Process Improvement in Small Organizations: A Case Study. IEEE Software 22(6), 68-75.
- 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, 8(2), 147-176.
- Curtis B., Kellner M. I., Over J. 1992. Process Modeling. Communications of the ACM 35(9), 75-90.
- Chrissis M. B., Konrad M., Shrum S. 2003. CMMI Guidelines for Process Integration and Product Improvement. Boston, Massachusetts: Pearson Education, Inc.

- Dybå T. 2000. Improvisation in Small Software Organizations. *IEEE Software* 17(5), 82-87.
- Dybå T. 2003. Factors of Software Process Improvement Success in Small and Large Organizations: An Empirical Study in the Scandinavian Context. Proceedings of the 9<sup>th</sup> European software engineering conference held jointly with 10<sup>th</sup> ACM SIGSOFT international symposium on Foundations of software engineering, Helsinki, Finland. New York, USA: ACM Press, 148-157.
- Fayad M. E., Laitinen M., Ward R. P. 2000. Software Engineering in the Small. *Communications of the ACM* 43(3), 115-118.
- Gresse von Wangenheim C., Anacleto A. & Salviano C. F. 2006. Helping Small Companies Assess Software Processes. *IEEE Software* 23(1), 91-98.
- Guerrero F., Eterovic Y. 2004. Adopting the SW-CMM in a Small IT Organization. *IEEE Software* 21(4), 29-35.
- Horvat R. V., Györkös J. & Rozman I. 2000. Managing the Complexity of SPI in Small Companies. *Software Process: Improvement Practice* 5, 45-54.
- Humphrey W.S., 1988. Characterising the Software process: A Maturity Framework. *IEEE Software*, 5(2), 73-79.
- Johnson A. 1994. Software Process Improvement in the DP/MIS Function Experience Report. Proceedings of the 16<sup>th</sup> International conference on Software engineering, Sorrento, Italy. Los Alamitos, CA, USA: IEEE Computer Society Press, 323-329.
- Johnson D. L. & Brodman J.G. 1997. Tailoring the CMM for small businesses, small organizations, and small projects. *Software Process Newsletter* 8, 1-6.
- Keeni G. 2000. The Evolution of Quality Processes at Tata Consultancy Services. *IEEE Software* 17(4), 79 - 88.
- Laitinen M., Fayad M. E., Ward R. P. 2000. The Problem with Scalability. *Communications of the ACM* 43(9), 105-107.
- Mc Caffery F., Taylor P. S., Coleman G. 2007. *Adept: A Unified Assessment*

- Method for Small Software Companies. *IEEE Software* 24(1), 24-31.
- Nunes N. & Cunha J. 2000. Wisdom: A Software Engineering Method for Small Software Development Companies. *IEEE Software* 17(5), 113-119.
- Paulk M., Curtis B., Chrissis M. & Weber C. 1993a. The Capability Maturity Model for Software, Version 1.1. Technical Report CMU/SEI-93-TR-25, Software Engineering Institute.
- Paulk M., Weber C., Curtis B. & Chrissis M. 1995. The Capability Maturity Model: Guidelines for Improving the Software Process. Reading, Massachusetts: Addison-Wesley.
- Pitterman B. 2000. Telecordia Technologies: The Journey to High Maturity. *IEEE Software* 17(4), 89-96.
- Richardson I. & Gresse von Wangenheim C. 2007. Why Are Small Software Organizations Different?. *IEEE Software* 24(1), 18-22.
- Russ M. L., McGregor J. D. 2000. A Software Development Process for Small Projects. *IEEE Software* 17(5), 96-101.
- Software Engineering Institute. 2000 [online]. Process Maturity Profile of the Software Community 1999 Year End Update. Saatavilla [www-muodossa <http://www.sei.cmu.edu/appraisal-program/profile/pdf/SW-CMM/1999aug.pdf>](http://www.sei.cmu.edu/appraisal-program/profile/pdf/SW-CMM/1999aug.pdf) [viitattu 28.2.2007].
- Software Engineering Institute. 2004 [online]. Process Maturity Profile Software CMM 2003 Year End Update. Saatavilla [www-muodossa <http://www.sei.cmu.edu/appraisal-program/profile/pdf/SW-CMM/2003sepSwCMM.pdf>](http://www.sei.cmu.edu/appraisal-program/profile/pdf/SW-CMM/2003sepSwCMM.pdf) [viitattu 28.2.2007].
- Software Engineering Institute. 2006 [online]. Process Maturity Profile Software CMMI v1.1 2006 Mid-Year Update. Saatavilla [www-muodossa <http://www.sei.cmu.edu/appraisal-program/profile/pdf/CMMI/2006sepCMMI.pdf>](http://www.sei.cmu.edu/appraisal-program/profile/pdf/CMMI/2006sepCMMI.pdf) [viitattu 28.2.2007].
- Software Engineering Institute. 2007 [online]. The IDEAL Model. Saatavilla [www-muodossa <http://www.sei.cmu.edu/ideal/>](http://www.sei.cmu.edu/ideal/) [viitattu

28.2.2007].

SPICE. 2004 [online]. Saatavilla [www-muodossa](http://www.muodossa.com) <<http://www.sqi.gu.edu.au/spice/>> [viitattu 30.3.2005].

Sutton S. M. Jr., 2000. The Role of Process in a Software Start-up. IEEE Software 17(4), 33-39.

Tilastokeskus. 2007 [online]. Tietojenkäsittelypalveluyritysten koot. Saatavilla [www-muodossa](http://statfin.stat.fi/statweb) <<http://statfin.stat.fi/statweb>> [viitattu 26.2.2007].

Ward R. P., Laitinen M., Fayad M. E. 2000. Management in the Small. Communications of the ACM 43(11), 113-116.

Ward R. P., Fayad M. E., Laitinen M. 2001. Software Process Improvement in the Small. Communications of the ACM 44(4), 105-107.

Wohlwend H. & Rosenbaum S. 1994. Schlumberger's Software Improvement Program. IEEE Transactions on Software Engineering 20(11), 833 - 839.