

hyväksymispäivä

arvosana

arvostelija

Siirtyminen elinkaarimallista ketterään
ohjelmistoliiketoimintaan: hallinnoinnin haasteet

Mikael Wargh

Helsinki xx.x.2009
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET

Tiedekunta/Osasto Matemaattis-luonnontieteellinen		Laitos – Institution Tietojenkäsittelytieteen laitos	
Tekijä – Författare Mikael Wargh			
Työn nimi – Arbetets titel Siirtyminen elinkaarimallista ketterään ohjelmistoliiketoimintaan: hallinnon haasteet			
Oppiaine – Läroämne Tietojenkäsittelytiede			
Työn laji – Arbetets art Pro gradu –tutkielma		Aika – Datum xx.x.2009	Sivumäärä – Sidoantal
Tiivistelmä – Referat <p>Ketterät menetelmät tarjoavat nimensä mukaisesti ketteryyttä, muutosmyönteisyyttä ja joustavuutta perinteiseen, hieman kankeaan suunnitteluvetoiseen ohjelmistokehitysprosessiin nähden. Ketterät menetelmät luottavat pitkälti tiimityöskentelyyn ja asiantuntemukseen, mikä tuo haasteita hallinnonille. Varsinainen projektipäällikön rooli hämärtyy ja tilalle tulevat tiimivetoisuus, leadership- tyyppinen johtajuus ja entistä tärkeämpänä osaamispääoman hallinta. Tutkielman alussa tarkastellaan suunnitteluvetoista ohjelmistokehitystä perinteisen vesiputousmallin näkökulmasta. Toisessa osassa esitellään ketterää ohjelmistokehitystä kahden ehkä yleisimmän menetönsä, XP:n ja Scrumin avulla. Kolmannessa osassa syvennyttään johtamis-, henkilöstö- ja organisatorisiin haasteisiin, joita yritys voi muutoksessa kohdata. Lopuksi käydään käytännön esimerkin kautta kuvitteellisen yrityksen käymä yrityskulttuurimuutos sen siirtymässä perinteisistä elinkaarimallin mukaisista prosessimalleista ketterään liiketoimintamalliin.</p> <p>ACM Computing Classification System (CCS) C.0 [Computer Systems Organization]: General</p>			
Avainsanat – Nyckelord Elinkaarimallit, vesiputousmalli, ketterät ohjelmistometodit, XP, Scrum, projektinhallinta, johtajuus, tietämyksenhallinta, hyvinvointi, palkitsemisjärjestelmät, oppiva organisaatio, osaamispääoma, yrityskulttuuri			
Säilytyspaikka – Förvaringställe Kumpulan tiedekirjasto, sarjanumero C-			
Muita tietoja – Övriga uppgifter Muita tietoja...			

Sisällysluettelo

1	Johdanto.....	1
2	Alkuasetelma	2
3	Elinkaarimallin mukainen ohjelmistokehitys	4
3.1	Vesiputousmalli.....	4
3.2	Elinkaarimallin mukaisen projektin hallinnointi.....	8
4	Ketterät ohjelmistokehitysmetodit	10
4.1	Historiaa ja määrittelyä.....	10
4.2	Scrum	11
4.3	XP (eXtreme Programming).....	16
5	Siirtyminen elinkaarimallista ketteriin metodeihin.....	25
5.1	Johtajuus.....	27
5.2	Tietämyksenhallinta	31
5.3	Organisaatorakenne.....	37
6	Yrityskulttuurin näkökulma muutokseen.....	46
6.1	Yrityskulttuurin määritelmä	47
6.2	Transformatiivinen muutos	50
6.3	Yrityskulttuurimuutos keski-ikäisessä yrityksessä	54
6.4	Kulttuurimuutosesimerkki: siirtyminen elinkaarimallista ketteriin metodeihin ..	55
7	Yhteenveto.....	61
8	Lähteet	62

Liitteet

1. Ketteryyshänifesti
2. Microsoftin lähtötilanteen kartoitus

1 Johdanto

Perinteinen elinkaarimallin mukainen ohjelmistokehitys (life-cycle development) on saanut varteenotettavan kilpailijan ketteristä ohjelmistokehitysmetodeista (agile software development methods). Ketterät metodit tarjoavat nimensä mukaisesti ketteryyttä, muutosmyönteisyyttä ja joustavuutta perinteiseen, hieman kankeaan suunnitteluvetoiseen ohjelmistokehitysprosessiin nähden. Tunnusomaista ketterille ohjelmistoprojekteilte ovat tiimityöskentely, asiantuntijavetoisuus, matala arvohierarkia ja valmius nopeisiin suunnanmuutoksiin. Perinteinen projektipäällikön rooli hämärtyy ja tilalle tulevat tiimivetoisuus, hallinnoinnin (management) sijasta johtajuus (leadership) ja entistä tärkeämpänä haasteena osaamispääoman hallinta ja sen vaaliminen. Mikäli ohjelmistoyrityksen tarkoituksena on siirtyä suunnitteluvetoisista elinkaarimallin mukaisista toimintaprosesseista ketteriin kehitysmetodeihin, tulee huomioon ottaa myös mahdollinen kivulias yrityskulttuurimuutos ja muutoksen merkitys, niin henkilöstö-, prosessi-, rakenne- kuin hallinnointinäkökulmienkin kannalta. Ketterien metodien suuriin lupauksiin ei kannata langeta ilman perinpohjaista tutkimusta niiden soveltuvuudesta oman liiketoimintastrategian, organisaation, omien asiakassidosryhmien sekä kehitettävien tuotteiden kannalta. Vaikka ketterien metodien käytöstä on saatu hyviä tuloksia, niin läheskään joka projektiin ne eivät sovi. Tyypillisesti käyttöliittymäprojektit sekä isot tai erittäin turvallisuuskriittiset ohjelmistoprojektit soveltuvat huonosti ketterien metodien toimintakentiksi [Con01, NMM05, Boe02].

Tutkielman alussa esitellään kuvitteellinen perinteisen elinkaarimallin mukainen ohjelmistoyritys Macrosoft. Esimerkkiyritys toimii ikään kuin käytännön lähtökohtana tutkimusongelmalle eli ketterien metodien asettamille hallinnointihaasteille. Kolmannessa kappaleessa tarkastellaan elinkaarimallin mukaista ohjelmistokehitystä ja syvennyttään sen kouluesimerkkiin eli perinteiseen vesiputousmalliin. Tämän lisäksi käydään läpi yleisellä tasolla elinkaarimallin mukaisen projektin hallinnointia. Kolmannessa ja neljännessä osassa esitellään ketterää ohjelmistokehitystä kahden ehkä yleisimmän metodinsa, XP:n ja Scrumin avulla. Kummankin kappaleen lopussa käydään kevyesti läpi tarkasteltavan ohjelmistokehitysprosessin hallintaa. Viidennessä osassa pohditaan haasteita, joita syntyy yrityksen siirtyessä elinkaarimallin

mukaisesta ohjelmistokehityksestä ketteriin ohjelmistokehitysmetodeihin. Tutkimusalueen laaja-alaisuuden takia, pyritään tarkastelu rajoittamaan henkilöstönäkökulmaan, niin johtamis-, organisaatio-, kuin yrityskulttuuritasolla. Viidennen kappaleen avainsanoja ovat johtajuus, hyvinvointi, ja organisaatorakenne. Tutkielman lopussa tarkastellaan muutosta yrityskulttuurillisesti ja käydään läpi kuvitteellinen muutos toisessa kappaleessa esitellyn esimerkkiyrityksen Macrosoftin näkökulmasta.

2 Alkuasetelma

Tässä kappaleessa esitellään kuvitteellinen traditionaalista ohjelmistoliiketoimintaa harjoittava yritys. Esimerkistä on pyritty tekemään mahdollisimman realistinen kärjistämättä liiaksi mitään osa-aluetta. Esimerkissä käytetyt termit ja aihealueet käydään tutkielman edetessä läpi. Tutkielman lopussa tarkastellaan ketteriin toimintamalleihin siirtymistä yrityskulttuurin näkökulmasta, jossa lähtöasetelmana toimii tässä esitelty esimerkkiyritys, Macrosoft.

Oletetaan lähtöesimerkiksi perinteisen elinkaarimallin mukainen ohjelmistoyritys Macrosoft. Yritys on suhteellisen iäkäs ja jo sen perustamisesta lähtien luottanut kontrollin ja ohjauksen tuomaan tehokkuuteen. Vaikka pörssiyrityksi kasvaneen yrityksen perustajajäsenet ovat aikaa sitten korvattu yhtiön hallituksen suosittelemilla ylemmillä johtajilla, on kontrolloiva toimintatapa edelleen vallalla. Osittain tiukkaan kontrolliin ja ohjaukseen on syynä myös yrityksen toimintasektori eli internet-pankkipalveluiden kehitys. Ohjelmistokehitykseen käytetään hieman kevyempää vesiputousmallia, jossa voidaan tarpeen tullen palata vaiheissa myös taaksepäin. Käytännössä tämä vesiputousmallin mukainen toiminta näkyy projektien kokonaisvaltaisena toteutuksena eli projektit kulkevat aina saman vesiputousmallin mukaisen kaavan läpi, jonka ympärille yrityksen toiminnanohjausjärjestelmät on rakennettu. Pääpaino on siis kokonaisvaltaisella suunnittelulla ja jatkuvalla prosessin tehostamisella. Apuna käytetään laaja-alaisesti projektihallinnan työkaluja, kuten eri tason suunnitteludokumentteja, kaavioita, tarkkoja kustannusarviolaskelmia sekä aikataulusuunnitelmia. Organisaatiotasolla valvontaa ja tehostamista tukevat

muutoksenhallintajärjestelmät ja työtuntiseurannat. Valvontatyökalujen hallinta ja ylläpito työllistävät jo itsessään useita henkilöitä.

Työtiloina toimivat avokonttorit, joissa henkilöt istuvat suhteellisen tiiviisti hyvin neutraalin väri- ja kalustemaailman ympäröimänä. Kerran puolessa vuodessa pidetään kaksisuuntaiset kehityskeskustelut alaisten ja esimiesten välillä tiettyä kaaviota mukaillen. Organisaatiotasolla hyvinvointia mitataan verkossa vastattavilla mielipide- ja tyytyväisyyskartoituksilla. Palkitsemiskeinoina ovat lähes yksinomaan esimiesten peruspalkan päälle laskettavat tulospalkkiolisät. Muuten hyvinvointiin yritys pyrkii vaikuttamaan tukemalla työpaikkaruokailua lounasalennusten muodossa, tarjoamalla liikuntamahdollisuuksia omissa tiloissaan, helpottamalla sosialisointia yhteisillä tiloilla ja teknisillä apuvälineillä sekä pyrkimällä tasa-arvoiseen kohteluun kaikilla tasoilla.

Asiakasyhteistyö on suhteellisen tiivistä, mutta melko tehotonta käytettyyn aikaan nähden. Osittain tähän on syynä käytettyjen prosessimallien kankeus sekä suunnitteluun ja valvottavuuteen keskittynyt toimintamalli. Asiakassuhteiden tilaa taas mitataan asiakastyytyväisyystutkimuksilla, joita tehdään kohdeasiakayritysten eri toimintotasoilla. Johtamismalliltaan yritys pyrkii kovasti olemaan henkilökeskeinen, mutta organisaatorakenne useine tasoineen sekä runsaine valvonta- ja toimintatapoineen pakottaa keskittymään enemmän asioiden johtamiseen ja ylläpitoon, kuin johtajuustyyppiseen toimintaan. Arvohierarkia ei ole kovin korkea, mutta alaisten vaikutusmahdollisuudet ovat heikot johtuen osittain autoritäärisestä arvohierarkiasta ja esimiesten vähäisestä ajasta sekä kiinnostuksesta kuunnella muutosehdotuksia.

Tähän mennessä yritys on tuottanut suhteellisen hyvin, mutta viime vuosina on nousu- ja laskusuhdanteista huolimatta yrityksen tulos ollut yhä enenevässä määrin tappiollinen. Myös asiakastyytyväisyys on ollut hienoisessa laskussa. Kuitenkin henkilöstön tyytyväisyys on pysynyt suhteellisen samana, kuten myös tehdyt työtuntimäärät ja muutokset. Tappiollisen tuloskierteen syynä näyttäisi olevan uusi kilpaileva yritys Fastcorp, joka on lyhyessä ajassa kaapannut suuren osan markkinoista itselleen. Samaan aikaan yhä useampi asiakas suunnittelee siirtyvänsä käyttämään uuden yrityksen edullisempia

tuotteita vanhojen sijaan. Vaikka Macrosoft onkin pyrkinyt tehostamaan toimintaansa entisestään, ei se silti tahdo pystyä samaan toimitusvauhtiin ja kustannustehokkuuteen kuin Fastcorp. Samalla kasvaneiden työpaineiden takia moni arvokas avainasiantuntija on siirtynyt uuden yrityksen palvelukseen vieden samalla kaiken osaamisen omalta osa-alueeltaan mukanaan. Säilyttääkseen asemansa Macrosoft päättää aloittaa laajamittaisen muutoksen kohti kilpailevan yrityksen käyttämää ketterää toimintamallia.

3 Elinkaarimallin mukainen ohjelmistokehitys

Elinkaarimallin (life-cycle model). mukaista ohjelmistokehitystä kutsutaan myös välillä ehkä kuvaavammin suunnitteluvetoiseksi ohjelmistokehitykseksi (plan-driven software development). Suunnitteluvetoinen ohjelmistokehitys edustaa niinsanottua perinteistä ohjelmistokehityssuuntausta. Järkeilevä, tiukkaa kontrollia hakeva ns. insinööripohjainen ohjelmistokehitys on ollut vallalla melkein sen synnystä lähtien [NMM05]. Ohjelmistokehityksessä ongelmana perinteiseen teollisuustuotantoon verrattuna on se, ettei ongelmia voi yleensä ratkaista formaalisti, ainakaan järkevässä ajassa, ja siten optimaalisen ratkaisun löytäminen ei aina ole mahdollista. Perinteinen ohjelmistokehitys perustuukin pitkälti prosessikeskeiseen olettamukseen, että jatkuvalla tehostamisella ja etukäteissuunnittelulla pystytään tuotantoprosessi hiomaan lähes täydelliseksi [HiC01]. Johtamismalliltaan tämä tarkoittaa yleensä perinteisen byrokraattista kontrolloitua kehitysmallia [Hig04, Hig03].

Perinteisiksi elinkaarimalleiksi luetaan muun muassa vesiputousmalli (waterfall model) ja spiraalimalli (spiral model) [NMM05]. Näistä lähemmin tarkastellaan klassista vesiputousmallia.

3.1 Vesiputousmalli

Vesiputousmalli on ehkäpä kaikkein tunnetuin suunnitteluvetoisista ohjelmistokehitysmalleista. Sen ensimmäisen virallisen määritelmän esitti Winston W. Royce vuonna 1970 artikkelissa "Managing the Development of Large Software Systems" [Roy70]. Ironiseksi artikkelin tekee se, että siinä Royce itse toteaa mallin kelvottomaksi. Sana vesiputous tulee mallin tavasta edetä vaiheesta

toiseen tiukan suunnitelmallisesti projektin alusta sen päätöksen ikään kuin vesiputous valuessaan vuorenrintettä alas. Edellinen vaihe tulee aina saada päätökseen ennen siirtymistä seuraavaan vaiheeseen, sillä paluuta eri vaiheiden välillä ei ole. Alkuperäinen vesiputousmalli koostui viidestä eri vaiheesta, jotka ovat vaatimusmäärittely (requirements specification), suunnittelu (design), toteutus (implementation), verifiointi ja validointi (verification) ja ylläpito (maintenance) [Roy70](Kuva 1). Seuraavaksi läpikäydään nämä vaiheet pintapuolisesti kuvaamalla niiden tyypillinen sisältö. Käytännön ohjelmistoprojekteissa harvoin pystytään kuitenkaan etenemään puhtasoppisesti vaiheesta toiseen, sillä vaiheet voivat sisältää projektista riippuen hieman erilaisia painopisteitä, sekä iterointia voidaan joutua tekemään niin vaiheiden sisällä kuin niiden ulkopuolella [Wei03, Roy70].



Kuva 1, Vesiputousmallin eri vaiheet

3.1.1 Vaatimusmäärittely

Vaatimusmäärittelyvaiheen aikana määritellään tuotteelta vaadittavat ominaisuudet muun muassa kartoittamalla tuotteen sidosryhmät ja heidän tarpeensa [Pre00]. Usein tärkeä osa vaatimusmäärittelyä on käyttötapausten (use

cases) määrittely, joilla sidosryhmät pyrkivät kuvaamaan mitä he tuotteen käytöltä odottavat. Vaatimukset myös luokitellaan edelleen käyttäjä- ja järjestelmävaatimuksiksi, jotka kummatkin sisältävät useita eri alaluokkia [Pre00]. Lopuksi kerätyt vaatimukset validoidaan, joka tarkoittaa käytännössä vaatimusten ja asiakkaan toiveiden sekä vaatimusten välisten ristiriitojen ratkomista. Vaikka vesiputousmalli eteneekin kronologisesti vaiheesta toiseen, voidaan vaatimusmäärittelyprosessia iteroida [Roy70]. Tällöin aloitetaan tärkeimpien vaatimusten etsimisellä ja edetään kohti yksityiskohtia.

Vaatimusmäärittelyn tuloksena syntyy vaatimusdokumentti (software requirements document), joka määrittelee suunnan tulevalle projektille. Vaatimusdokumentti toimii suunnittelu- ja järjestelmätestausvaiheen pohjana sekä sopimuksena asiakkaan ja tuottajan välillä. Vaatimusmäärittely onkin hyvin tärkeä vaihe vesiputousmallissa, sillä tämän jälkeen asiakkaan rooli vähenee merkittävästi toteuttajaosapuolen keskittyessä suunnitteluun, toteutukseen ja testaukseen. Käytännössä tämä tarkoittaa sitä, että validoinnista mahdollisesti läpi päässeet virheelliset vaatimukset voivat aiheuttaa suuria ristiriita- ja aikatauluongelmia myöhemmissä projektin vaiheissa. [Pre00]

3.1.2 Suunnittelu

Suunnitteluvaiheen lähtökohtana on vaatimusdokumentti, jonka pohjalta aletaan luoda yksityiskohtaisempaa suunnitteludokumenttia. Suunnitteludokumentti sisältää eri abstraktiotasoa, joista korkeinta tasoa edustaa arkkitehtuurisuunnitelma [Pre00]. Arkkitehtuurisuunnitelmassa järjestelmä ositetaan osajärjestelmiksi ja määritellään niiden väliset viestinvälitystavat. Arkkitehtuurimäärittelyn tuloksena saadaan järjestelmän arkkitehtuurikuvaus. Seuraavassa abstraktiotasossa eli rajapintasunnittelussa määritellään arkkitehtuurisuunnitelmassa löydettyjen osajärjestelmien rajapinnat toisiinsa nähden. Hyvien rajapintamäärittelyiden mukaan rajapintojen tulee toimia niin, että rajapintoja käyttävän osajärjestelmän ei tarvitse tietää kutsuttavan osajärjestelmän sisäisestä toteutuksesta mitään [Pre00]. Rajapintasunnittelu voidaan näin ajatellen todeta suunnitteluvaiheen ehkäpä tärkeimmäksi osaluueksi. Oliopohjaisissa järjestelmissä voidaan suunnitteluvaiheen lähtökohdaksi ottaa myös oliopohjainen lähtökohta, jossa järjestelmän suunnittelu tehdään

tärkeimpien olioiden näkökulmasta. Riippuen ohjelmistoprojektista suunnitteluabstraktiotasoissa voidaan näin edetä lopulta pseudokooditasolle asti.

Suunnitteluvaiheen lopputuloksena on suunnitteludokumentti, joka toimii projektin teknisenä punaisena lankana tulevissa työvaiheissa. Oikeaoppisen vesiputousmallin mukaan tämäkin dokumentti lukitaan siirryttäessä seuraavaan vaiheeseen.

3.1.3 Toteutus

Vesiputousmallissa varsinainen toteutus on, tai ainakin sen tulisi olla, massiivisen etukäteissuunnittelun ansiosta vain pieni osa koko projektin työmäärästä. Mikäli suunnitteludokumentissa on päästy pseudokooditasolle asti, niin toteutusvaiheessa lähinnä puhtaaksikirjoitetaan suunnitelmat käytetyllä ohjelmointikielellä.

3.1.4 Verifiointi ja validointi

Toteutuksen jälkeen valmis tuote verifioidaan ja validoidaan. Kuten vaatimusmäärittelyvaiheessa verifiointivaiheen validoinnilla pyritään selvittämään, vastaako tuote sille asetettuja vaatimuksia. Verifioinnilla taas selvitetään tuotteen mahdolliset puutteet ja viat [Pre00]. Käytännössä verifiointi ja validointi suoritetaan erillisillä tarkastuksilla ja ohjelmistotestauksella. Tarkastukset eivät vaadi testauksen tapaan toimivaa ohjelmakoodia, vaan niissä pyritään etsimään valitusta tuotteen osa-alueesta virheitä tai puutteita. Tarkastuksen kohteena voi olla ohjelmakoodi tai vaatimusmäärittelyn osa. Tarkastusabstraktiotasosta riippuen tilaisuudessa voivat olla osanottajina projektin toteuttajat tai sitten laajempi sidosryhmäjoukko asiakas mukaan luettuna [Pre00]. Ohjelmistotestausmetodeja on useita, esim. korkeamman tason mustalaatikkotestaus (black-box testing) tai ohjelman sisäisen toiminnan huomioonottava lasilaatikkotestaus (glass-box, white-box testing).

Vaikka vesiputousmallin mukaan testaus ja validointi suoritetaan vasta toteutuksen jälkeen, käytännössä nämä ovat jatkuvia prosesseja. Testausta

lukuunottamatta voidaan verifiointi ja validointi aloittaa jo ennen valmista ohjelmakoodia [Pre00]. Isommissa projekteissa validointia ja verifiointia suorittaa toteutuksen "sivussa" usein siihen määrätty oma ryhmänsä yhteistyössä ohjelmoijien kanssa.

3.1.5 Ylläpito

Kun tuote on hyväksyttävästi verifioitu, validoitu ja luovutettu asiakkaalle, ollaan vasta tuotteen elinkaaren alussa. Lehmanin teoreemojen [Leh80] mukaisesti ohjelmistolla on oma evoluutionsa ja ohjelmiston tulee muuttua tai se käy vähä vähältä hyödyttömämmäksi muuttuvassa toimintaympäristössään. Ohjelmiston ylläpitovaihe on siis aikavaativuudeltaan yleensä huomattavasti pidempi kuin sitä edeltävät vaiheet. Yleisiä ylläpitovaiheen tehtäviä ovat ohjelmiston virheiden korjaus, ohjelmiston sovittaminen uuteen ympäristöön (software porting) ja uusien ominaisuuksien lisäys. Ohjelmiston evoluutiossa näitä paremmin nykyisiä tarpeita vastaavia päivityksiä kutsutaan ohjelman uusiksi julkaisuiksi (software release).

3.2 Elinkaarimallin mukaisen projektin hallinnointi

Vesiputousmallin mukaisen projektin hallinta perustuu yleensä ns. perinteisiin projektiliiketoiminnan oppeihin [Hig04, NMM05]. Tämä tarkoittaa käytännössä vaativaa projektipäällikön roolia. Projektipäälliköltä vaaditaan liiketoiminnallista, teknistä, organisatorista ja sosiaalista sekä luonnollisesti myös hyvää projektinhallinnan osaamista [AMK06]. Toki projektin muiden jäsenten on hyvä hallita myös näitä osa-alueita, mutta yleensä puhtaassa vesiputousmallissa projektipäällikön koordinoiva rooli on merkittävästi suurempi kuin ketterien metodien mukaisissa projekteissa.

Vaatimusmäärittelyssä korostuu projektipäällikön liiketoiminnallinen osaaminen, sillä hänen tulee ymmärtää asiakkaan todelliset liiketoimintalähtöiset tarpeet ja pyrkiä esittämään näihin tarpeisiin mahdollisia ratkaisuja. Aina edes asiakas itse ei pysty kertomaan kaikkia tarpeitaan edes tarpeeksi formaalisti, jotta tekninen kehitystiimi sen ymmärtäisi.

Ehkäpä projektin alun vaativin työ projektin johtajalle on tarjouksen valmistelu. Talon sisäisissä projekteissa tämä voi tarkoittaa myös pelkän projektin budjetin määrittelyä, joka esitetään yhtiön johdolle hyväksyttäväksi. Tarjouksen suunnittelussa tulee hänen pystyä hahmottamaan koko tuleva projekti sen kaikilta mahdollisilta puolilta, kuten tekniseltä vaativuudelta, tarvittavilta osaamistarpeilta ja liiketoiminnallisilta vaatimuksilta [AMK06]. Projektin pääasiallinen kohde eli tuote tulee myös olla symmetriassa yhtiön vision kanssa, mikä karkeasti ottaen tarkoittaa sitä, ettei esimerkiksi peliyhtiö lähde kehittämään virustorjuntaohjelmaa.

Suunnitteluvaiheessa projektipäällikön varsinaiset projektihallintataidot ja tekniset taidot ovat erityisasemassa [AMK06]. Eri abstraktiotason suunnitelmia tehdessä on myös asiakkaan liiketoimintatarpeet pidettävä mielessä, jottei tekninen "tyylikkyys" ajaisi niiden ohi [Kr108]. Aikataulukutus ja henkilöresurssien tehokas hyväksikäyttö vaativat projektipäälliköltä useiden eri projektinhallintakeinojen osaamista. Näitä ovat muun muassa janakaaviot, virstanpylväskaaviot, tehtävien keston arviointilaskennat ja tehtäväverkot kriittisine polkuineen [AMK06].

Toteutus- ja verifiointivaiheissa projektipäällikön tekninen asiantuntemus on avainasemassa, jotta hän pystyy koordinoimaan ja seuraamaan tarkasti projektin etenemistä. Seuraamisen apuvälineenä projektipäällikkö voi käyttää esimerkiksi toteutuneen työn budjeoitua kustannuslaskentaa (Earned Value Management, EVM), jolla pyritään laskemaan saavutetun todellisen työn suhde kulutettuihin ja käytössä oleviin resursseihin. Myös ongelmanratkaisutaidot, muutosten, tiedon ja laadun hallinta ovat tärkeitä taitoja johdettaessa projektin toteutusta kohti päämääräänsä [AMK06].

Projektin päättämisvaiheessa korostuu seuranta, raportointi ja päättämistoimet. Hyvin edenneestä projektista tulee pystyä tunnistamaan onnistumisen avainkohdat tulevia projekteja ajatellen [AMK06]. Asiakkaan liiketoimintaymmärrys on myös tärkeä osa päättämis- ja ylläpitovaihetta, sillä tuotehan on vasta elinkaarensa alussa. Asiakkaan muutostarpeiden tunnistaminen ja uusien asiakasyhteistyömahdollisuuksien löytäminen vaativat kuitenkin hyvää asiakasyhteistyötä ja ymmärrystä [AMK06].

Koko projektin keston aikana aina ylläpitovaiheeseen asti projektipäällikön yksi tärkeimmistä tehtävistä on projektisuunnitelman kirjoitus ja sen ylläpito. Projektisuunnitelmassa kuvataan projektin sisältö, tavoitteet, työ, toimintatavat ja johtamisperiaatteet [AMK06]. Jos mahdollista, projektisuunnitelma laaditaan yhdessä asiakkaan ja tärkeimpien sidosryhmien kanssa. Projektisuunnitelman päämääränä on saada koko projektiryhmä ymmärtämään yhteinen tehtävänsä [AMK06].

4 Ketterät ohjelmistokehitysmetodit

Ketterät ohjelmistokehitysmetodit ovat tyypiltään iteratiivisia (iterative development) menetelmiä, joissa projekti pyritään pilkkomaan pieniin erikseen suoritettaviin osiin. Joka iteraation alussa selvitetään sen tavoitteet asiakkaan kanssa ja määritellään syklin puitteet. Iteraatioiden lopputuloksena saadaan aina toimiva tuote asiakkaan testattavaksi ja jatkojalostettavaksi. Yhteistä ketterille metodeille ovat asiakkaan vahva läsnäolo, lyhyet kehityssykli, pienet tiimit ja vahva kommunikaatio [Con01].

4.1 Historiaa ja määrittelyä

Ketterien ohjelmistometodien todellista syntyä on vaikea määrittää. Vuonna 1972 IBM esitteli tiimipohjaista ohjelmoijajohtajavetoista sovelluskehitysmallia (chief programmer team management of production programming), jossa pienet ketterät tiimit työskentelivät asiantuntevan johtajan alaisuudessa [Bak72]. Johtajan rooli oli erittäin tärkeä ja hänen tehtävänä oli hahmottaa ohjelmistoprojektin sisäinen arkkitehtuuri ja tiukalla asiantuntemuksella johtaa projekti päätökseen. Testaus oli tärkeässä roolissa ja yksi toiminnan kulmakivistä oli kehittää koko ajan toimivaa koodisientä pikkuhiljaa lopulta valmiiksi lopputuotteeksi.

Toinen ehkä yleisemmin hyväksytty ketterien metodien esikuva on Toyota tuotantojärjestelmällään (TPS, Toyota production system) ja joustavalla tuotekehityksellään (lean development) [Mor05]. Joustavan tuotekehityksen pääteemana on jätteen minimointi (waste minimization). Jätteet luokitellaan

seitsemään kategoriaan, joista esimerkkeinä mainittakoon ylituotanto, odottaminen ja liike. Tehostusta haettiin siis minimoimalla vaikkapa työntekijän työssään tarvitsema liike tai odotusaika. Ketterissä sovellusmetodeissa yksi pääteema onkin vain tarpeellisten osien toteuttaminen jättäen kaikki ylimääräinen pois. Voidaankin sanoa, että ketterät menetöt rohkaisevat joustavaan ajatteluun [NMM05]. Käytännön määrittelyiltään joustava tuotekehitys ja siitä aikanaan johdettu joustava ohjelmistokehitys ovat pääteemoiltaan pitkälti samanlaisia ketterien ohjelmistometodien kanssa [Mor05].

Mukautuva sovelluskehitys (adaptive software development) taas eroaa ketteristä sovelluskehitysmetodeista siten, ettei se varsinaisesti tarjoa mitään käytännön toimintamalleja, vaan on enemmänkin kehys ja toimintafilosofia [Con01]. Kuitenkin vuonna 2001 julkaistun ketteryysmanifestin (agile manifesto, kts. liite 1) [Bec01] suhteellisen väljät määritykset voivat kattaa myös mukautuvan ohjelmistokehityksen. Muita ketteryysmanifestin mukaisia sovelluskehitysmetodeja ovat mm. Scrum, Crystal Clear, XP (eXtreme Programming) ja ominaisuusvetoinen sovelluskehitys (Feature Driven Development). Tässä tutkielmassa tarkastellaan lähemmin kahta ehkä yleisimmin käytettyä ketterää metodologiaa: Scrumia ja XP:tä. Kummankin peruseräite selvitetään pääpiirteittäin painopisteen ollessa henkilöstönäkökulmassa.

4.2 Scrum

Scrum on yksi vanhimmista ketteristä sovelluskehitysmetodeista. Jo vuonna 1986 alkunsa saanut Scrum on hyvin skaalautuva ja yleisesti käytetty metodi. Scrumin ydinajatus on, että luonteeltaan kompleksisia ja monimutkaisia ohjelmistoprojekteja ei voida suunnitella etukäteen tai muutenkaan käyttää tuotantoteollisia oppeja sellaisenaan, mihin elinkaarimalleissa pyritään, vaan kehitysprosessin tulee olla empiirinen [ScB02]. Ehkä merkittävin Scrumia popularisoiva teos on Ken Schwaberin ja Mike Beedlen kirjoittama "Agile Software Development with Scrum", jota käytetään tämän kappaleen lähdeteoksena.

Yksi Scrumin ominaispiirre on, että se ottaa kantaa myös projektinhallintaan ja projektin sidosryhmiin [TaN86]. Kuten muutkin iteratiiviset kehitysmetodit, Scrumin toiminta perustuu peräkkäisiin iteraatioihin, joiden lopputuloksena on

aina toimiva lopputuote. Scrumissa toteutusvaiheen (sprint) kestoksi on määritelty noin neljä viikkoa. Projektin sidosryhmät jaetaan vastuultaan sika- ja kanarooleihin. Roolinimet tulevat vanhasta vitsistä, jossa sika ja kana perustavat ravintolan ja sika valittelee, että häneltä vaaditaan täydellistä sitoutumista, kun taas kanalta vain osittaista sitoutumista munimisen muodossa.

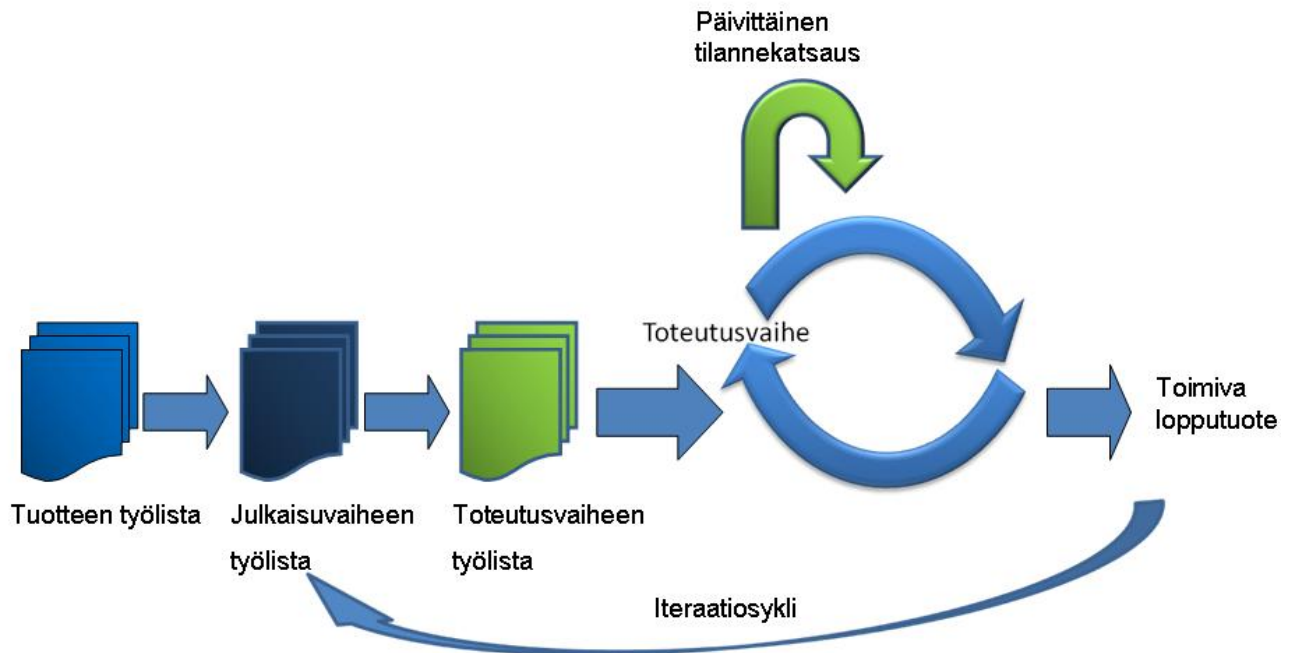
Scrum ei XP:n tapaan sisällä useita arvo- tai periaatemäärittelyitä. Käytännön toimintatapoihinkaan Scrum ei varsinaisesti ota kantaa kuten XP. Schwaber ja Kent määrittelevätkin Scrumin olevan usein vain "käärepaperi" perinteisille ohjelmistokehitysmetodeille. Toteutusvaihe kuvataankin yksinkertaisesti täysin tiimin hallinnoimaksi eli tiimi voi esimerkiksi pitää vaikka päivän mittaisen suunnittelupalaverin mikäli niin tahtoo.

4.2.1 Roolit

Sikaroolia ovat tuotteen omistaja eli asiakas, tiimi ja ns. scrum-mestari (ScrumMaster). Kuten muissakin ketterien ohjelmistometodien määrittelyissä, asiakas esittää projektin kannalta ehkäpä tärkeintä roolia. Scrumissa omistaja toimii asiakkaan toiveiden keulakuvana, kirjoittaa käyttäjätarinoita (user stories), priorisoi ne ja lisää ne tuotteen työlistaan (product backlog). Käyttäjätarinat ovat muuten samanlaisia kuin vesiputousmallin käyttötapaukset, mutta tarinat voivat koskea käyttöliittymäsidonnaisuudesta poiketen lähes kaikkea mahdollista. Tarinat ovat vapaamuotoisia, noin kolmen lauseen mittaisia määrityksiä ilman teknistä sanastoa [Wel06]. Scrum-mestarin tehtävänä on tarjota tiimille mahdollisimman soveliaat työskentelypuitteet ja dokumentoida projektin etenemistä. Tiimin tehtävänä on tuottaa varsinainen lopputuote. Huomattavaa on, että tiimin jäsenet koostuvat taitavista asiantuntijoista, joilla on kykyä toimia monella sovelluskehityksen osa-alueella. Samoin sikaroolia kantavat projektin jäsenet omaavat virallisista nimityksistä riippumatta projektissa suuren päätösvalan ja kantavat myös ison osan projektin vastuista.

Kanaroolia ovat käyttäjät ja muut sidosryhmät, jotka hyötyvät tuotteesta eli antavat projektin olemassaololle merkityksen sekä päättävissä asemissa olevat henkilöt, jotka määräävät ympäristön projektille. Kanat ovat siis nimensä

mukaisesti välillisesti mukana projektissa, mutta joita ilman projektia ei kuitenkaan olisi.



Kuva 2, Scrum-prosessi

4.2.2 Työlistat

Scrumin toiminta keskittyy sen kolmeen työlistään, jotka ovat tuotteen työlista, julkaisuvaiheen työlista (release backlog) ja toteutusvaiheen työlista (sprint backlog) (Kuva 2). Tuotteen työlista on projektin korkeimman tason työlista, joka sisältää kaikki toteutettavaksi aiotut vaatimukset ja toiveet. Tämän listan virallinen omistaja on yleensä asiakas, mutta kuka tahansa voi muokata sitä. Tuotteen työlistan vaatimuksille määritellään yleensä projektin omistajan antama liiketoiminnallinen tärkeys ja tiimi puolestaan määrittelee vaatimuksille arvioidun työpanoksen. Ketterän ajattelumallin mukaisesti kahdesta liiketoimintatärkeydeltään vastaavasta vaatimuksesta valitaan yleensä pienemmän työpanoksen vaativa versio, koska se tuo enemmän tuottoa kulutetuille resursseille. Joka iteraation alussa projektin omistaja ilmoittaa toteutustiimille, mitkä tuotteen työlistan vaatimuksista hän haluaisi

toteutettavan. Tämän tiedon pohjalta tiimi pyrkii arvioimaan, mitkä halutuista ominaisuuksista ehditään ja voidaan toteuttaa seuraavan toteutusvaiheen aikana.

Näin valitut vaatimukset päätyvät julkaisuvaiheen työlistaan, joka sisältää tarkat toteutus- ja kehityssuunnitelmat sekä mahdollisimman todenmukaiset resurssitarpeet valituista vaatimuksista. Ennen siirtymistä varsinaiseen toteutusvaiheeseen tiimi pitää toteutusvaiheen suunnittelutapaamisen (sprint planning meeting), jossa julkaisuvaiheen työlistan vaatimukset pilkotaan aikavaatimukseltaan noin 8-16 tunnin työsuorituksiin ja sijoitetaan toteutusvaiheen työlistaan.

Toteutusvaiheen työlista sisältää siis tarkan kuvauksen tulevan toteutusvaiheen tavoitteista pilkottuna alle 16 tunnin työsuorituksiin. Kaikki toteutusvaiheen työlistan tavoitteet on tarkoitus saada kokonaan valmiiksi toteutusvaiheen aikana dokumentointia, testausta ja koodausta myöten. Toteutusvaiheen aikana toteutusvaiheen työlista on lukittu, eikä siihen saa tehdä muutoksia. Toteutuksen edetessä ylläpidetään päivittäin tilannekaaviota (burn down chart), joka kertoo jäljellä olevien tehtävien määrän toteutusvaiheen työlistassa suhteessa käytettävissä olevaan aikaan.

4.2.3 Tapaamiset

Scrum sisältää omia iteraatioiden eri vaiheisiin liittyviä tapaamisia. Toteutusvaiheen aikana pidetään päivittäinen 15 minuutin tilannekatsaus (scrum, daily stand-up) (Kuva 2). "Daily stand-up"-sanonta tulee siitä, että yleensä on kaikkien tapana seistä tapaamisen aikana, jottei tilaisuus veny turhan pitkäksi [Ber06]. Tilannekatsaus järjestetään aina täsmälleen samaan aikaan, samassa paikassa ja samalla tavalla. Katsauksessa vain "siat" saavat puhua ja jokainen jäsen vastaa kolmeen kysymykseen, jotka ovat:

- Mitä olet tehnyt eilisen tapaamisen jälkeen?
- Mitä aiot tehdä tänään?
- Onko vastassa jotain ongelmia, jotka estävät suorittamasta työtehtävää?

Verrattuna perinteiseen autoritäärisen johtamismalliin, tilanne on sinänsä erikoinen, että Scrum-sääntöjen mukaan esimerkiksi korkean esimiesaseman omaavalla kanaosanottajalla ei ole valtaa ohjata toteutusvaiheessa sikaroolin

omaavia henkilöitä, vaan hän toimi lähinnä sivustaseuraajana. Näin Scrum ohjaa arvohierarkian madaltamiseen sekä vallan ja vastuun hajauttamiseen.

Päivittäisen tapaamisen lisäksi muita tapaamisia on edellä mainittu toteutusvaiheen suunnittelupalaveri ja toteutusvaiheen päätöspalaveri (sprint review meeting), jossa käydään läpi suoritettut työt, suorittamattomat työt, sekä esitellään lopputulos sidosryhmille (demotilaisuus). Toteutusvaiheen päätöspalaverin lisäksi pidetään yleensä myös toteutusvaiheen katselmus, jossa tiimin jäsenet tarkastelevat suoritettua toteutusvaihetta ja pyrkivät kehittämään toteutusprosessiaan.

4.2.4 Scrum-projektin hallinta

Scrum-projektin hallinta ja toiminta poikkeavat melko paljon perinteisestä projektinhallinnasta. Toteutusvaiheen työlistan töitä ei määrätä tietyille henkilöille, vaan tiimin jäsenet poimivat ne tuntemuksiensa mukaan. Schwaber ja Beedle toteavat, että projektin alussa tiimi sisäistää nopeasti itseorganisoinnin (self-organization) ja tuottavuuden periaatteen ja alkaa toimia yhtenäisenä yksikkönä projektin päämäärän saavuttamiseksi.

Joka iteraatiossa, ja myös siis sen sisällä, tiimin jäsenillä on mahdollisuus toimia monessa eri roolissa. Tämä poikkeaa perinteisestä vesiputousmallista, koska yleensä pitkän vaatimusmäärittelyn ja suunnittelun jälkeen eri osa-alueille nimetään vastuulliset, jotka kantavat roolinsa projektin loppuun asti. Työntekijän näkökulmasta Scrum siis avartaa kokonaisuuden hahmottamista ja antaa mahdollisuuden kehittyä usean osa-alueen moniosaajaksi. Tämä onkin epävirallisesti eräs ketterien metodien vaatimus: tavallista asiantuntevammatt kehittäjät [Con01, Kr108].

Toinen hyvin merkittävä ero tavalliseen elinkaarimallin mukaiseen projektikehitykseen on asiakkaan suuri rooli projektin eri vaiheissa. Itse asiassa asiakkaasta tulee osa kehitystiimiä ja hänen roolinsa projektissa on hyvin haastava [Ber06, Kr108]. Asiakkaan vastuulla on mm. projektin määränpään ja tilan hahmottaminen niin liiketoiminta-, kuin toteutusmielessä. Hänen tulee pystyä kommunikoimaan kaikkien sidosryhmien kanssa ja kyetä pitämään

kokonaisuus mielessä vastatessaan tiimin jäsenten mahdollisesti hyvinkin haastaviin teknisiin kysymyksiin. Vaarana on myös koko projektin viivästyminen, ellei asiakas pysty keskustelemaan kehittäjien kanssa tarpeeksi konkreettisella tasolla. Onkin erittäin tärkeää, että asiakasedustaja on tarpeeksi innostunut ja asiantunteva pystyäkseen ohjaamaan projektia tehokkaasti [Ber06, Kr108]. Asiakkaan roolin voidaan katsoa kasvaneen ketterissä metodeissa lähemmäksi projektipäällikön asemaa verrattuna elinkaarimallin mukaiseen projektiliiketoimintaan.

Vaikka Schwaber ja Beedle korostavatkin tiimin yhteistoimintaa ja roolijaon sumeutta, on scrum-mestari ehkä selvemmin esillä kuin projektipäällikkö XP:n roolijaossa. Scrum-mestarin rooli on kuitenkin perinteisestä projektipäällikön roolista poiketen enemmänkin valmentamista ja ystävystymistä tiimin kanssa, kuin käskyttämistä, yksityiskohtaista asiajohtamista tai auktoriteetin ylläpitoa [Hig04]. Schwaber ja Beedle jatkavatkin, että ohjaamisen normaalit apuvälineet, kaaviot, työajanseurannat, pitkät seurantapalaverit, sekä dokumentointi voidaan jättää kokonaan pois tai ainakin niiden käyttöä vähentää radikaalisti. Menestyvä johtaminen vaatii tietämyksen ja muutoksen ohjaamista kevyellä otteella kohti haluttua päämäärää. Toisin sanoen johtamisen henkilöstönäkökulma korostuu perinteisen kontrolloinnin tieltä.

4.3 XP (eXtreme Programming)

XP:n kehitti Chryslerillä aikoinaan Kent Beck. Vuonna 1999 hän julkaisi kirjan "Extreme Programming explained", joka aloitti tarinan "äärimmäisestä ohjelmoinnista" eli eXtreme Programming:sta. Tämän kappaleen lähdeteoksena käytetään edellä mainitun kirjan vuonna 2004 ilmestynyttä toista painosta [BeA04].

Vaikka Beckin projekti Chryslerillä keskeytettiin vuonna 2000, on XP:n käyttö laajentunut voimakkaasti sen jälkeen. Scrumista XP eroaa pääasiassa kuvaamalla lähemmin toimintalogiikka ja matalan tason toimintamalleja, kuin projektinhallintaa. Scrumiin verrattuna, XP on myös hieman enemmän aatevoittoisempi, sillä XP:ssä on eri käytäntöjen lisäksi arvo- ja periaatemäärittelyitä.

4.3.1 XP:n toimintavaiheet

XP-prosessi jaetaan neljään perusvaiheeseen, jotka ovat suunnittelu (planning), ohjelmointi (coding), testaus ja mallinnus (design). Toteutustasolla XP muistuttaaakin hieman vesiputousmallia, mutta vain prosessitasolla. Käytännön toimintatavat ovat selkeästi määriteltyjä ja edustavat XP:n mukaisia arvoja ja periaatteita.

Ilman hyvää suunnittelua järjestelmästä tulee ennen pitkää helposti tehoton ja monimutkainen vaikeasti hallittava kokonaisuus. Kaikkien tiimiläisten tulisi ottaa osaa suunnitteluun, sillä mikäli tiimiläiset eivät tunne kaikkia liiketoiminnallisia tarpeita, he eivät voi toimia optimaalisesti työtehtävissään eri abstraktiotasoilla. Vesiputousmallista poiketen XP:ssä suunnittelua tehdään jopa päivittäin ja mahdollisimman myöhään. Myöhäisellä suunnittelulla pyritään varmistamaan kaiken tarpeellisen informaation saatavuus suunnitelmia tehtäessä.

Testausta tehdään usein, automatisoidusti ja mahdollisimman aikaisessa vaiheessa. Näillä testausperiaatteilla pyritään vähentämään ohjelmavirheiden (defects) määrää mahdollisimman paljon. XP:n käytännöistä eritoten testivetoinen kehitys (test driven development) ja pari-ohjelmointi (pair programming) tukevat virheiden minimointia. Testaus voidaan jakaa kahteen alakategoriaan, joista matalan tason testit vastaavat vesiputousmallin ohjelmistotestausta ja korkeamman tason testit vesiputousmallin validointia [BeA04, Pre00]. Jatkuvilla testauksella ja sen äärimmäisellä automatisoinnilla pyritään myös helpottamaan toteuttajien työtaakkaa.

Ohjelmointivaihe ja testausvaihe vuorottelevat jatkuvasti, sillä XP:n periaatteiden mukaisesti vain testattu koodi on hyväksyttyä koodia. Käytännössä testaus- ja ohjelmointivaihe ovat pitkälti samoja, koska lukuisten testien kirjoittaminen voidaan myös laskea välillisesti testivetoisen kehityksen mukaan ohjelmakoodin tuottamiseksi.

Järjestelmän mallinnus vastaa jotakuinkin vesiputousmallin suunnitteluvaiheen eri abstraktiotasojen mallinnusta. XP:n näkökulma asiaan on yksinkertaisesti järjestelmän vaativuuden ja halutun laatutason suhde. Jos toteutettava

järjestelmä ei ole vaativa, ja vaadittu laatutaso on matala, voidaan hyvin toimia mallinnuksessa pelkän vaiston varassa, mutta vaativuuden ja laatuvaatimusten kasvaessa ainoa ratkaisu ongelmaan on Beckin mielestä kokeneen asiantuntijan läsnäolo. Mallinnusvaiheessa XP:n yksinkertaistamisarvo on tärkeässä asemassa ja Toyotan joustavan ajattelumallin mukaisesti jätettä eli turhan ohjelmakoodin, rakenteiden ja logiikoiden toistoa pyritään minimoimaan.

4.3.2 XP:n arvot, periaatteet ja käytännöt

XP sisältää useita suositeltavia ajattelumalleja, aatteita ja periaatteita. Nämä voidaan jaotella abstraktiotasoinn korkeammasta matalampaan arvoiksi (values), periaatteiksi (principles) sekä ensisijaisiin (primary) ja seuraajakäytäntöihin (corollary practices). Periaatteet yhdistävät korkean tason arvomäärittelyt matalan tason käytäntöihin.

4.3.2.1 Arvot

Ylimpänä XP:n mukaisessa toiminnassa ovat arvot. Arvot ohjaavat aistien tuottaman informaation tulkintaa, sekä ohjaavat toimintaa. Ilman käytäntöjä tukevia arvoja, toiminta muuttuu nopeasti pelkäksi päämäärättömäksi itseään ruokkivaksi rutiinityöskentelyksi. Beck antaa esimerkin ohjelmoijasta, joka yrittää kiistää ohjelmassa olevan virheen. Ongelmana on tässä tapauksessa enemmänkin ohjelmoijan oppimis- ja itsekehittymisarvojen matala tärkeysaste kuin virheellinen ohjelmointitekniikka. Työtä tukevien arvojen puute on haitaksi niin työn kohteelle, organisaatiolle kuin työntekijälle itselleen. XP kuvaa viisi tärkeää sen toimintaa tukevaa arvoa: kommunikointi (communication), yksinkertaistaminen (simplicity), palaute (feedback), rohkeus (courage) ja kunnioitus (respect). Myös muita tiimisuntautuneita arvoja voidaan sisällyttää XP:n mukaiseen kehitysmalliin kuten esimerkiksi yrityksen oman tiimityöskentelyprosessin mukaisia arvoja.

Tärkein kaikista arvoista on kommunikointi. Ongelmatilanteissa tarvittava tietotaito yleensä on olemassa, mutta kommunikointiongelmien takia vaikeasti

saatavissa. Mikäli ongelma on täysin uusi, kommunikoinnin merkitys korostuu yhteisessä ratkaisunhaussa sekä todetun ratkaisun edelleeninformoinnissa.

Beckin toteamuksen mukaan yksinkertaistaminen on kaikkein järkeenkäyvin XP:n arvoista. Ongelma pyritään ratkaisemaan yksinkertaisimmalla mahdollisella tavalla asettamatta järjestelmän toimintaa vaakalaudalle. Sidosryhmien todelliset tarpeet ja tietotaitotaso määräävät myös yksinkertaistamistason. Asiakkaalle on esimerkiksi turha tehdä monimutkaista järjestelmän hallinnointityökalua, jos tarkoitus on vain käyttää järjestelmän peruspalveluita. Yksinkertaistamisen peruseriaate on Toyotan jätteen minimoimis-periaatteen mukainen.

Kaikki suunnitelmat vanhenevat ennen pitkää. Oli sitten kyseessä järjestelmän arkkitehtuuri tai sen vaatimukset. Koska muutos on väistämätöntä, on sitä pyrittävä myös seuraamaan ja toimittava uusien tilanteiden mukaisesti. Palautteen kerääminen ja sen esiintuominen ovat kiinteästi sidoksissa ensimmäiseen arvomäärittelyyn eli kommunikointiin.

Rohkeutta vaaditaan vaarallisten asioiden kohtaamiseen. Sovelluskehityksessä pelottavia asioita voivat olla esimerkiksi tosiasioiden puhuminen, kärsivällisyys ongelmanratkaisussa, tehottoman koodin hylkääminen ja estottomasti uusien ratkaisujen etsiminen [BeA04]. Muut arvot vaativat myös välillisesti rohkeutta kuten esimerkiksi tosiasioiden puhuminen tukee kommunikointia ja tehottoman koodin hylkääminen tukee yksinkertaistamista.

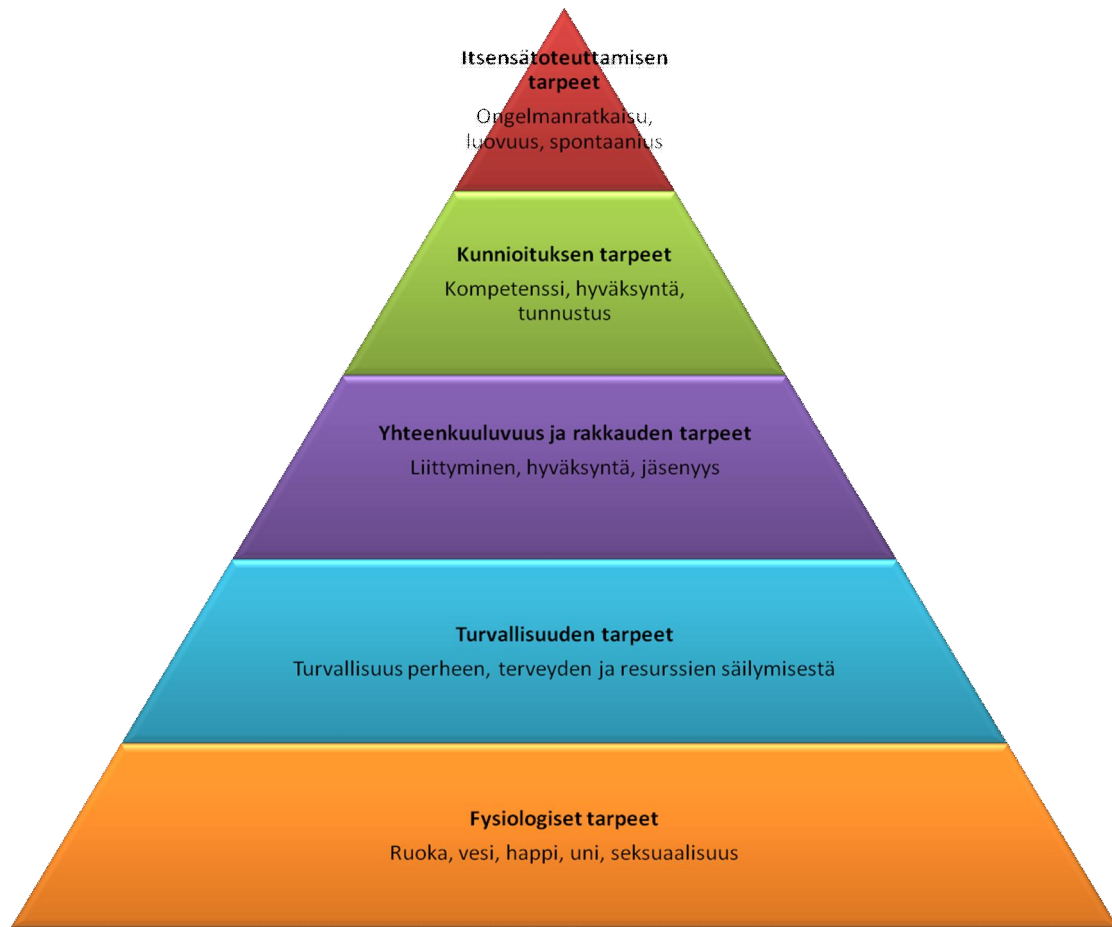
Kaikkien edellä mainittujen arvojen tukiranka on kunnioitus. Mikäli tiimin jäsenet eivät kunnioita toisiaan tai kannu vastuuta projektin lopputuloksesta, XP ei voi toimia.

4.3.2.2 Periaatteet

Arvot ovat liian abstrakteja ohjatakseen suoraan XP-tiimin jokapäiväistä toimintaa. Esimerkiksi joissakin tilanteissa dokumentointi voi olla tehokkaampi ratkaisu kuin tiivis keskusteluyhteistyö. Kuitenkin sekä dokumentointi että keskustelu ovat kummatkin kommunikointi-arvomäärittelyä alle laskettavia toimintatapoja. Beck korostaa tässä tapauksessa kontekstin merkitystä. Käytettävä kommunikointitapa tulisi olla luontaisen toimintatavan mukaista, joka

useimmissa tapauksessa on ihmisille luonteva keskustelu. XP:n periaatteiden tavoitteena on selventää näitä arvojen ja käytäntöjen välisiä tarkoituksia.

Sovelluskehityksessä tarvitaan inhimillisyyttä (humanity). Epävarma tulevaisuus ja yhteisöön kuulumattomuuden tunne eivät ole hyviä kannustimia työntekijälle. Beckin määritelmä inhimillisyydelle noudattelee löyhästi Maslowin tarvehierarkiaa (Kuva 3), joka pyrkii kuvaamaan ihmisen tarvetta toteuttaa itseään fyysisistä tarpeista henkisiin [Mas43]. Motivoitunut ja inhimillisesti kohdeltu työntekijä on taloudellisesti tärkeä voimavara projektille, mutta Maslowin hierarkian mukaisesti tämän motivaation saavuttamiseen vaatii ensin matalampien tasojen tyydyttämistä. Maslowin hierarkiaa ja sen suhdetta henkilön hyvinvointiin käydään lähemmin hyvinvointi-kappaleessa. Toinen keskeinen XP:n periaate on taloudellisuus (economics). Tiimin toiminnan tulisi aina hyödyttää liiketaloudellisesti jotain osapuolta eikä keskittyä tekniseen näyttävyyteen. Esimerkiksi liiketoimintamielessä tärkeimpien vaatimusten toteuttaminen ensin maksimoi projektin tuoton (roi, return of investment). Ohjelmistoprojektin taloudellisuutta kuvataan sen valmistumisnopeudella ja jatkokehiteltävyydellä. Mitä nopeammin tuote on valmis, sen nopeammin se alkaa tuottaa. Mikäli samaa tuotetta tai sen runkoa voidaan käyttää useampaan tarkoitukseen, se on arvokkaampi kuin vain alkuperäisen tarpeen täyttävä ohjelmisto. Kaikki XP:n käytännöt on suunniteltu näitä kahta taloudellista näkökulmaa silmällä pitäen.



Kuva 3, Maslowin tarvehierarkia

Muita periaatteita ovat samanlaisten ratkaisujen uudelleenkäyttö (self-similarity), jatkuva kehitys (improvement), tasainen kehitysvauhti (flow), ongelmien näkeminen muutosmahdollisuuksina (opportunity), resurssien päällekkäisyyksien minimointi (redundancy), ongelmista oppiminen (failure), mahdollisimman korkean laadun ylläpito (quality) ja isojen muutosten toteutus pienissä osissa (baby steps).

Erityisesti tiiminäkökulmasta tärkeitä ovat lisäksi yhteinen hyöty (mutual benefit), erilaisuuden hyväksyntä (diversity) ja vastuun hyväksyntä (accepted responsibility). Yhteinen hyöty tarkoittaa muiden osapuolten huomioonottamista, kuten esimerkiksi kattavien testien ja riittävän dokumentaation kirjoittamista tuotteen ylläpitoa ajatellen. Jokainen tiimi tarvitsee taidoiltaan, asenteiltaan ja näkökulmiltaan erilaisia osaajia. Tämä luo suotuisan pohjan tiimin sisäisille konflikteille. Konfliktit tulee kuitenkin pystyä näkemään erilaisina

mahdollisuuksina ratkaista kohteena oleva ongelma. Vastuun hyväksynnällä tarkoitetaan kykyä nähdä saatu vastuu myös antajan vastuuna. Jos esimerkiksi projektin vetäjä pyytää ohjelmoijaa vastaamaan lukualgoritmin suorituksesta, ohjelmoija voi luottaa siihen, että projektin vetäjä on myös vastuussa ohjelmoijan työstä.

4.3.2.3 Käytännöt

Beck jakaa käytännöt ensisijaisiin ja seuraajakäytäntöihin. Käytännöt tukevat ja toteuttavat periaatteiden välityksellä XP:lle määritetyt arvot.

Työntekijän hyvinvointia ja humaaniusarvoa tukevia ensisijaisia käytäntöjä ovat erityisesti yhdessä istuminen (sit together), koko tiimi (whole team), informatiivinen työympäristö (informative workspace), innostava työ (energized work) ja löyhän ajan järjestäminen (slack). Järjestämällä tiimiläisille yhteiset informaationvälitystä tukevat tilat ja mahdollisuus tehokkaaseen kanssakäymiseen saadaan aikaan tiimihenkeä tukeva ilmapiiri, jossa tiimiläiset tuntevat kuuluvansa yhteen ja pyrkivät auttamaan toisiaan, niin työssä, kasvamisessa, kuin oppimisessa. Työtaakan kohtuullisena pitämistä voidaan auttaa löyhällä ajalla, jota voidaan saada esimerkiksi määrittelemällä joka iteraatiolle muutama vähempiarvoinen tehtävä, joka voidaan tarpeen tullen pudottaa pois.

Muita ensisijaisia käytäntöjä ovat pariohjelmointi (pair programming), tarinat (stories), viikottaiset syklit (weekly cycle), neljännesvuosittaiset syklit (quarterly cycle), kymmenen minuutin järjestelmätestaus (ten-minute build), jatkuva integraatio (continuous integration), testivetoinen kehitys (test-first programming, test-driven development) ja inkrementaalinen suunnittelu (incremental design).

Seuraajakäytännöt ovat käytäntöjä, joihin siirtyminen on mielekästä vasta kun ensisijaiset käytännöt on pääpiirteittäin sisäistetty. Beck jakaa seuraajakäytännöt:

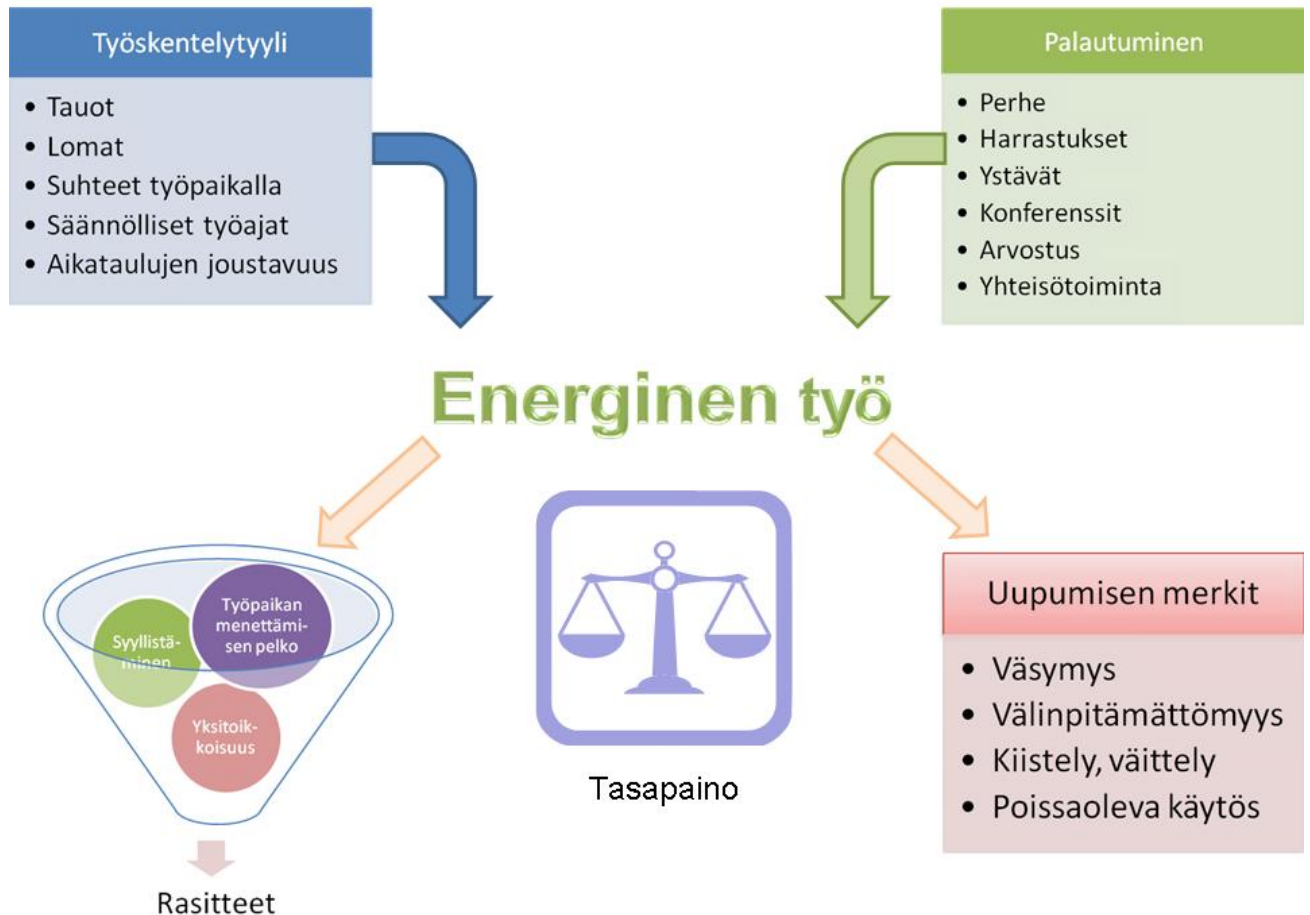
- Asiakkaan mukaanottamiseen (real customer involvement), joka vastaa pitkälti Scrumin mukaista asiakkaan integroimista tiiviisti tiimiin.

- Inkrementaaliseen järjestäytymiseen uuden ajattelumallin mukaisesti (incremental deployment) eli askeleittain siirrytään vanhoista perinnejärjestelmistä pois.
- Tiimien yhdessäpitämiseen (team continuity)
- Tiimien järjestämiseen (shrinking teams)
- Ydinongelmien analysointiin (root-cause analysis)
- Kaikille yhteiseen koodiin (shared code)
- Koodin ja testin lähtökohtana pitämiseen kaikille muille dokumenteille (code and tests)
- Yksittäisen koodijuuren hallintaan (single code base) eli ei ylläpidetä useita päällekkäisiä koodipolkuja samassa projektissa.
- Päivittäiseen julkaisuun (daily deployment) eli joka yö luodaan uusi ohjelmistopäivitys, koska ohjelmoijan koneella olevan koodin ja julkaisun välinen ero on aina riski.
- Lyhyihin sopimuskausiin (negotiated scope contract) eli riskien minimoimiseksi sovitaan mielummin monta pientä kiinteää sopimusta, kuin yksi pitkä.
- Käytönaikaiseen laskutukseen (pay-per-use) eli kerätään maksu järjestelmän käytön määrästä

4.3.3 XP-projektin hallinta

XP ei itsessään sisällä Scrumin tapaan selkeitä roolijakoja projektin sisällä. Toki XP jaottelee koko tiimin eri rooleihin, joita on kokonaiset yhdeksän aina johtajista ohjelmoijaan, mutta Beck peräänkuuluttaa joustavien työroolien merkitystä [BeA04]. Joku voi toimia tietyssä roolissa tai sitten tilanteen mukaan useammassa roolissa samanaikaisesti. XP:n arvojen mukaisesti toiminta perustuu tiimin saumattomaan yhteistyöhön ja sosiaalisten esteiden purkamiseen. Kaiken takana ovat tietysti liiketaloudelliset näkökulmat, mutta ne saavutetaan vain panostamalla sidosryhmien yhteistoimintaan ja pyrkimällä kaikille edulliseen toimintamalliin. Toisin sanoen Beck perustaa paljon työntekijöiden hyvinvoinnille tuloksen tekijänä. Suuri osa XP:n arvo-, periaate- ja käytäntömäärittelyistä tähtääkin nimenomaan tähän päämäärään. Muun muassa kommunikointi- ja kunnioitusarvoihin, sekä näistä johdettuihin inhimillisyyttä-, yhteisen hyödyn ja erilaisuuden hyväksymisperiaatteisiin pohjautuvan innostavan työn käytännön

määrittely keskittyy projektin jäsenten tehokkuuteen heidän hyvinvoinnin näkökulmasta. Beck kuvaa innostavan työn rakennusaineet työtyyliin (work style), rasitteiden (drains) ja palauttavien (recharge) tekijöiden tasapainona (Kuva 4).



Kuva 4, Innostavan työn tasapainokartta [BeA04 mukaillen]

Tehokas ja tuottava tiimi koostuu jäsenistä, jotka pystyvät tasapainottelemaan työn ja vapaa-ajan välillä. Huomattavaa on, että muut tiimiläiset ja sidosryhmät voivat vaikuttaa tasapainoon välittömästi vain rasitteita vähentämällä, esimerkiksi välttämällä syyllistävästä ilmapiiristä. Työuupumista kuvaavat merkit, kuten levottomuuden, huomaavat useimmiten ensimmäisenä työntekijän läheiset, eikä hän itse, minkä takia läheisten tulisi informoida ja pyrkiä avustamaan henkilöä mahdollisimman aikaisessa vaiheessa. Beckin kuva antaa alustavan hahmotelman ketterien metodien sosiologisista ja hyvinvoinnillisista haasteista.

Itse asiassa XP:stä on yleistetty humaani, ihmisläheinen johtamismalli XPM (eXtreme Project Management), jonka tarkoituksena on toimia erittäin monimutkaisten ja epävarmuutta sisältävien projektien johtamismallina.

5 Siirtyminen elinkaarimallista ketteriin metodeihin

Perinteisistä ohjelmistoprosessimalleista poiketen ketterät menetöt tukeutuvat ihmisiin varsinaisen prosessin sijaan [CoH01]. Kontrollon sijaan tehostusta haetaan vastuuta jakamalla ja henkilöstöä sitouttamalla. Tämä kontrollin heikentyminen tuo mukanaan epävarmuutta, koska järjestelmää ei edes pyritä kuvaamaan täysin etukäteen. Ketterien metodien tyypillisiä piirteitä ovat lyhyet iteratiiviset syklit, tuotteen kehittäminen ominaisuuksia lisäämällä, jatkuva kausittainen prosessin kehittäminen, yhteisöllinen päätöksenteko, nopea reagoiminen muutoksiin ja palautteeseen, sekä jatkuva järjestelmän ohjelmakoodin muuttaminen ja päivitys [HiC01, Hig04]. Projekti pilkotaan pienempiin alaprojekteihin eli iteraatiosykleihin, jotka sisältävät yleensä vesiputousmallista tuttuja suunnittelu-, kehitys-, testaus- ja luovutusvaiheita. Jokaisen iteraation lopputuloksena on toimiva, asiakkaan käytettävissä oleva tuote. Työ tapahtuu pienehköissä tiimeissä asiakkaiden eli loppukäyttäjien ollessa tiiviisti kehityksessä mukana muun muassa päättämässä tiimin kanssa, mitkä ominaisuudet ovat seuraavan iteraation kohteena. Eri taustaisten ja eri päämääriä hakevien sidosryhmien mukaan ottaminen yhteiseen päätöksentekoon onkin yksi ketterien metodien tunnuspiirre. Tämän tyyppisissä projekteissa perinteinen elinkaarimallin mukainen kontrolloiva johtajamalli ei ole enää mahdollinen, vaan johtajan tulee pystyä yhteistyötä kannustavaan, motivoivaan johtajuuteen (leadership) [Hig04].

Ehkä merkittävin haaste ketterään ohjelmistoliiketoimintaan siirryttäessä on tietämyksen hallinta (knowledge management). Ketterät menetöt kannustavat toimimaan suunnittelun ja dokumentoinnin sijaan [Bec01]. Tämä tarkoittaa käytännössä tietämyksen muuttumista yhä enenevässä määrin näkymättömäksi, sisäiseksi tietämykseksi (tacit knowledge) ulkoisten artefaktien sijaan [NMM05]. Toiminta muuttuu samalla perinteistä ohjelmistoprosessia vapaamuotoisemmaksi

ja ihmiskeskeisemmäksi, jolloin työntekijöiden hyvinvoinnin ja jaksamisen merkitys korostuu.

Perinteisten ja ketterien ohjelmistoprosessien avainerot on lueteltu allaolevassa taulukossa (taulukko 1).

	Traditionaalinen	Ketterä
Keskeiset oletukset	Järjestelmät ovat kokonaan määriteltävissä ja ne voidaan rakentaa huolellisella sekä mittavalla suunnittelulla	Korkealaatuiset ja mukautuvat ohjelmistot voidaan kehittää pienissä tiimeissä käyttämällä toimintaperiaatteina jatkuvaa prosessin kehitystä ja nopean palautteen ja muutoksen päälle rakentuvaa testausta.
Kontrolli	Prosessikeskeinen	Henkilöstökeskeinen
Johtamistyyli	Käskytyks ja kontrollointi	Johtajuus ja yhteistyö
Tietämyksenhallinta	Ulkoinen	Sisäinen
Roolijaottelu	Erikoistumista suosiva yksilökeskeisyys	Roolinvaihtoa suosivat itsejärjestäytyvät tiimit
Kommunikaatio	Formaali	Informatiivinen
Asiakkaan rooli	Tärkeä	Kriittinen
Projektisykli	Tehtävien tai aktiviteettien ohjaama	Tuotteen ominaisuuksien ohjaama
Kehitysmalli	Elinkaarimallin mukainen (vesiputous, spiraali tai joku variaatio)	Ketterien metodien mukainen iteratiivinen evoluutio-toimitusmalli
Suosittelava organisaatorakenne	Mekaaninen, byrokraattisen formaali	Orgaaninen, joustavuutta, sitoutumista ja sosiaalista yhteistyötä suosiva rakenne
Teknologia	Ei rajoituksia	Suosii oliopohjaista teknologiaa

Taulukko 1, Perinteisten ja ketterien ohjelmistoprosessien avainerot [NMM05 mukaillen]

Taulukosta näkee, että perinteiset ja ketterät menetelmät ovat pitkälti toistensa vastakohtia. Tästä johtuen tiukasti perinteiseen elinkaarimalliin sitoutuneet yritykset kohtaavat todennäköisesti isoja haasteita ottaessaan ensiaskeleita kohti ketteriä metodeja [NMM05]. Tutkimukset ovatkin osoittaneet, että sovelluskehitysprosessien muutokset ovat monimutkaisia ja vaikeita organisaatiomuutoksia. Pelkkien työkalujen ja teknologian vaihtaminen ei riitä, vaan huomioon on otettava organisaation rakenne, kulttuuri ja johtamistekniikat

[SNM01]. Tämän takia siirryttäessä ketteriin metodeihin on ensimmäisten – ja kaikkein tärkeimpien – askeleiden tavoitteena oppia ymmärtämään organisaation asettamat puitteet muutokselle [NMM05].

Seuraavaksi tarkastellaan lähemmin johtajuustyylin (management, leadership), tietämyksenhallinnan (knowledge management) ja organisaatorakenteen muutoksesta syntyviä haasteita. Näiden tarkastelukohteiden yhteydessä sivutaan myös välillisesti kontrolli-, roolinjako- (role assignment), kommunikointi- ja asiakkaan roolin muutoksen asettamia haasteita. Eri osa-alueet luonnollisesti ihmisistä ja organisaatiosta puhuttaessa menevät hieman ristiin. Esimerkiksi organisaatorakenteen muutoksiin liittyy tiiviisti johtamistyylin muutos. Yrityskulttuuri-kappaleessa käydään varsinaista muutosprosessia läpi kuvitteellisen Microsoft-esimerkkiyrityksen avulla, jolloin tässä kappaleessa esitetyt haasteet yhdistyvät yhdeksi kokonaisuudeksi.

5.1 Johtajuus

Perinteisen johtamisen (management) ja johtajuuden (leadership) ero on pieni mutta merkittävä. Siinä missä johtaminen keskittyy monimutkaisuuden hallintaan ja järjestämiseen, johtajuus keskittyy muutoksen hallintaan [Hig04]. Kaikki projektit tarvitsevat jonkun tason hallintaa tai lopputuloksena on kaos [Hig04, BuH06]. Perinteiselle johtamiselle tyypilliset apuvälineet, kustannuslaskelmat, projektisuunnitelmat, raportit, kaaviot ja budjetit, eivät kuitenkaan riitä ohjelmistokehityksessä tyypilliseen huomattavan epävarmuuden hallintaan [Hig04]. Kontrollin sijasta vastuuta jaetaan ja kokonaisuutta ohjataan hienovaraisesti kohti haluttua päämäärää. Highsmith [Hig04] esittelee käsitteen yhteistyöjohtajasta (leadership-and-collaboration), joka on hieman epämääräinen yleistys johtajuusteorioista. Seuraavassa kappaleessa pyritään tarkentamaan tätä johtajuusmääritelmää, minkä jälkeen tarkastellaan sen soveltuvuutta ketterien ohjelmistoprojektien johtamiseen perinteisen johtamismallin tukena.

5.1.1 Transformatiivinen johtajuus

Johtajuustutkimuksen alkutaipaleella 1900-luvun alkupuoliskolla pyrittiin perinteiset johtajat ja johtajuuteen kykenevät henkilöt tunnistamaan

piirreteorioilla (trait-spotting) [BuH06]. Lukuisia tutkimuksia tehtiin, jotta pystyttäisiin esittämään hyvältä yhteistyöjohtajalta vaadittavat piirteet. Koska eri johtamistilanteet vaativat yleensä monimuotoisia ohjaustapoja tiukasta kontrollista kevyeen ohjaukseen, syntyi uusi sisältö- ja tilanneteorioihin keskittyvä tutkimushaara. Tämän tutkimushaaran perusteella hyviä johtajia olivat ne, jotka pystyivät tilanteesta riippuen saumattomasti toimimaan visonäarisestä toimintatavasta kontrolloivaan [BuH06]. Muita johtajuuden tutkimushaaroja ovat johtajien toimintatyylejä tutkivat tyyliteoriat, johtajan ja alaisten välisen vuorovaikutussuhteen vaihdantana kuvaavat vaihdantateoriat (transactional leader) ja psykodynaamiset johtajuuden selitysmallit, jotka korostavat johtajan persoonallisuuden, tunteiden ja reagoititapojen merkitystä vuorovaikutussuhteissa [Leh04, BuH06].

Viime vuosikymmenen jälkipuoliskolla vallalle on kuitenkin noussut ajattelutapa transformatiivisesta johtamisesta (transformational leading) [Bur78], joka on myös Highsmithin yhteistyöjohtajakäsitteen takana. Transformatiivinen johtaja pyrkii motivoimaan alaisiaan ja aikaansaamaan heissä ja samalla itsessään muutoksen kehittää itseään kohti haluttua parempaa päämäärää sekä yhä korkeampaa osaamistasoa. Transformatiivisen johtamisen tunnuspiirteet voidaan jakaa organisaatiojohtamisen, persoonallisuuspiirteiden sekä yleisen johtamisen ja työyhteisön kehittämisen alakategoriaihin [MeM02, MeM03] (Kuva 5).

Yleinen johtaminen ja työyhteisön kehittäminen

- On aidosti kiinnostunut kanssaihmiesten hyvinvoinnista
- Delegoi, tukee ja rohkaisee muutokseen
- On helposti lähestyttävissä

Organisaatiojohtaminen

- Pyrkii kommunikoidaan ja verkostoitumaan
- Tarkentaa ja selkeyttää yksilön ja tiimin tavoitteita, prioriteetteja ja tarkoitusperiä
- Ohjaa kohti yhteistä visiota
- Luo kehitystä tukevan kulttuurin
- Hienovaraisesti ohjaa muutosta kohti haluttua päämäärää

Persoonallisuuspiirteet

- Käyttäytyy tasapuolisesti, oikeudenmukaisesti ja yhdenmukaisesti sekä on avoin uusille ideoille
- Inspiroi muita karismaattisuudellaan ja tietämyksellään
- Pystyy analyyttisellä ja luovalla ajattelulla ratkaisemaan monimutkaisia ongelmia
- On pätevä liiketaloudellisessa ja yrittäjämäisessä ajattelussa

Kuva 5, Transformatiivisen johtajan tunnuspiirteitä [MeM02, MeM03 mukailen]

5.1.2 Transformatiivinen johtajuus ja ketterät ohjelmistoprosessit

Transformatiivisen johtamisen syvin olemus piilee muutoksessa. Muutokseen tulee pyrkiä ja siihen kannustaa. Organisaatiotason ratkaisulla rakennetaan myös muutosta kannustava ilmapiiri. Kuten edellä on todettu, muutoksen hyväksyminen on erityisen tärkeää ketterissä ohjelmistometodeissa. Sekä Scrumissa, että XP:ssä henkilöiden roolit saattavat vaihtua moneen kertaan projektin aikana ja usein tähän pyritäänkin, jotta tiimiläiset saisivat arvokasta kokemusta eri työtehtävistä. XP:ssä kannustetaan tiimiläisiä kokeilemaan uusia toimintatapoja ja hylkäämään vanhoja. Scrumissa taas määritellään täysin perinteisestä vesiputousmallista poikkeava projektitoimintatapa erityisine tapaamisineen ja toimintavaiheineen. Ketterien metodien lyhyet iteraatiot tarkoittavat samalla nopeaa kehitystahtia ja siten jatkuvasti muuttuvaa toimintaympäristöä. Myös projektipäällikön rooli hämärtyy, koska projektin vetäjän näkyvimät artefaktit eli suunnitelmat, esitykset, dokumentit ja raportit

jäävät ketteryysmanifestin mukaisesti toimittaessa taka-alalle. Projektin toiminta keskittyy sen toteuttajiin, joiden asiantuntemuksen varassa koko projekti etenee.

Tiimikeskeisestä näkökulmasta johtuen projektipäällikön tärkeimmäksi rooliksi jää tiimin toimintaympäristön ja ilmapiirin kehittäminen tukemaan yhteistä päämäärää. Erityisesti XP:ssä tämä hienovarainen transformatiivisuus korostuu, koska prosessi rakentuu arvojen ja periaatteiden päälle, joiden omaksumiseen projektipäällikkö voi vain välillisesti vaikuttaa. Scrumissa lähtökohta on hieman rajatumpi selkeine roolijakoineen, mutta myös siinä tiimidynamiikan ja yhteistyön merkitys projektin onnistumiselle on selkeästi suuremmassa asemassa, kuin perinteisessä elinkaarimallin mukaisessa projektijohtamisessa. Lähes kaikki transformatiivisen johtajuusmäärittelyn kohdat tukevat nimenomaan tätä ilmapiirin ja toimintaympäristön kehittämistä tukemaan haluttua tavoitetta (kuva). Tiimiläisten tulee pystyä toimimaan projektipäällikön kanssa kehittävästi ilman ylimääräisiä jännitteitä. Tämän yhteistyön edellytyksenä projektipäälliköltä vaaditaan tasa-arvoista, rehellistä, yhtenäistä ja kiinnostunutta käytöstä (kuva). Myös johtajan vaikutusmetodien tulee tukea tätä kehityssuuntausta. Käskytyksen sijaan transformatiivinen johtaja visioi muutoksen ja pyrkii hienovaraisesti saamaan aikaan ilmapiirin, joka tukee tätä muutosta ja saa ihmiset motivoitumaan. Tärkeää on myös vastuun delegeoiminen ja päätöksenteon jakaminen muille tiimiläisille. Johtajasta tulee tässä prosessissa osa tiimiä, jonka arvostuksen ja samalla hyväksytyyn auktoriteettiaseman hän ansaitsee toisin kuin perinteisessä byrokratiassa, jossa auktoriteetti ja valta-asema tulevat automaattisesti ammattinimikkeen myötä [Hig04, Ota05].

On huomattava, etteivät johtaja ja johtajuus missään nimessä ole toisiaan poissulkevia käsitteitä. Käytännössä näiden kahden johtamistavan rajat ovat hyvin hämäriä. Yleisesti ottaen johtajalla tulee olla myös johtajuustaitoja ja itse johtajuus voidaan tulkita vain yhdeksi osaksi monihaaraista johtamisroolia [Min77]. Esimerkiksi elinkaarimallin mukaisen sovellusprojektin johtaja tarvitsee myös johtajuustaitoja normaalin hallinnointityönsä lisäksi, mutta ketterässä ohjelmistoliiketoiminnassa tämä tarve korostuu. Pelkällä asioiden tiukemmalla johtamisella ja järjestämisellä saadaan toimintaa ehkä lyhyellä tähtäimellä tehostettua, mutta samalla henkilöstön hyvinvointi ja motivaatio kärsii, mikä taas pitkällä tähtäimellä ei ole eduksi erityisesti tiimipainoiteisessa asiantuntijaliiketoiminnassa [Buh06, Ota05]. Jonkinlainen kontrolli on kuitenkin

pidettävä, koska jo pelkästään yrityksen yhteinen visio sitä vaatii. Ketterien ohjelmistoprojektien vetäjien tulisi pystyä kumpaankin johtamismalliin johtajuustyypin korostuessa. Tämä on usein hankalaa, koska perinteinen johtamismalli pyrkii samaan järjestystä kaaokseen, kun taas yhteistyöjohtajat pyrkivät muutokseen [Hig04]. Muutos tarkoittaa perinteisen johtamismallin näkökulmasta usein lisää epäjärjestystä ja kontrollointihaasteita, minkä takia näiden kahden ajattelumallin yhdistäminen voi olla johtajalle liikaa. Taiteilu näiden kahden välillä onkin ohjelmistoprojektin johtamisen varsinainen haaste: tulee dokumentoida organisoida ja suunnitella, mutta vain sen verran kuin on välttämätöntä [Hig04, Coc06].

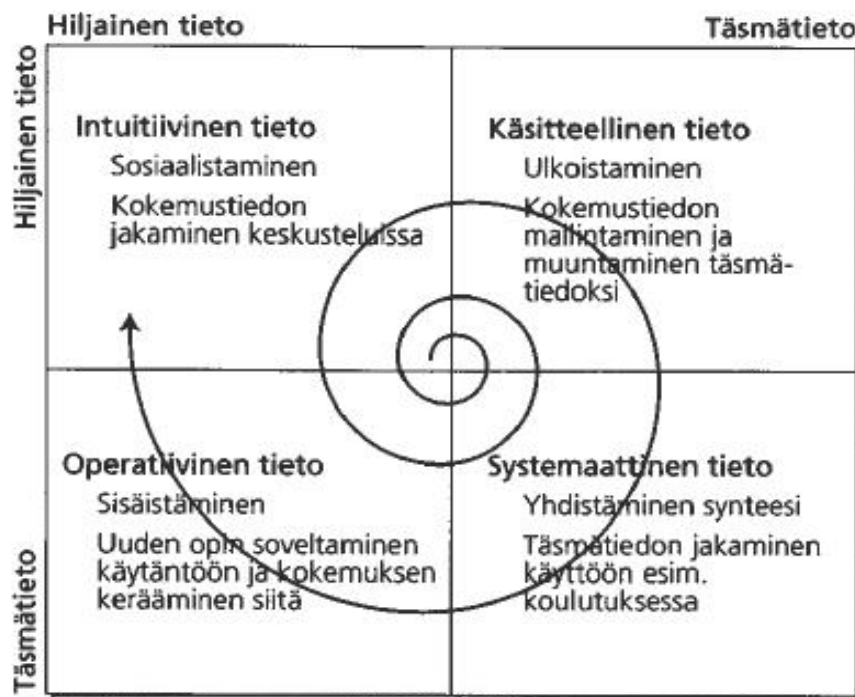
5.2 Tietämyksenhallinta

Lyhyesti määriteltynä tietämyksenhallinnan tarkoituksena on saada yksittäisten organisaation jäsenten sisäinen tietous ulkoiseen muotoon niin, että se voidaan jakaa organisaation muille jäsenille [BuH06]. Kuten jo edellä on todettu perinteisestä elinkaarimallin mukaisesta projektimallista ketteriin metodeihin siirryttäessä sisäisen tiedon merkitys kasvaa. Tämä ilmenee jo ketteryysmanifestissa, joka painottaa toteutusta dokumentoinnin kustannuksella. Ennen pitkää siis yhä suurempi osa työyhteisön tietämyksestä muuttuu kokemuksen kautta sisäiseksi, näkymättömäksi osaamispääomaksi. Elinkaarimallin mukaisen prosessikeskeisen ajattelun sijaan tulisi näin ollen siirtyä ihmiskeskeiseen ajattelumalliin, jossa työntekijöitä kohdellaan yhtiön tärkeimpänä voimavarana, eikä pelkkänä kustannusrasitteena [NoT95]. Highsmith [Hig04] toteaaakin, että tärkeintä projektin onnistumiselle ovat sen jäsenet. Tässä transformaatioissa on tärkeitä ymmärtää tietämyksenhallinnan varsinainen päämäärä eli sisäisen tietouden muuntaminen ulkoiseksi tietämykseksi.

5.2.1 Sisäinen ja ulkoinen tietämys

Tietämyksenhallinnan kahtiajaottelun keksijinä pidetään Nonakaa ja Takeuchia, jotka kirjoittivat vuonna 1995 kuuluisan kirjansa "The Knowledge Management Company: How Japanese Companies Create the Dynamics of Innovation" [NoT95]. Sisäinen tietämys käsittää yksilön intuition, päätöksenteon ja muut

näkymättömät mentaaliset mallit ja taidot. Tunnusomaista sisäiselle tietämykselle on sen ilmaisun vaikeus. Kuuluisa vertauskuva on pyörällä ajamisen taito ja sen siirtäminen pienelle lapselle. Ulkoista tietämystä (myös täsmätieto, explicit knowledge) on taas kaikki artikuloitu, koodattu ja kirjoitettu tieto, joka on kaikkien saatavilla [NoT95]. Tärkein Nonakan ja Takeuchin väite on sisäisen ja ulkoisen tietämyksen kiinteä riippuvuus toisistaan. Sisäinen tietämys opitaan ja sisäistetään ulkoisesta tietämyksestä, josta tulee näin sisäistä tietämystä ja ennen pitkää taas ulkoista ihmisen omien ajattelumallien suodattamina. Tätä sykliä kutsutaan myös uuden tiedon syntysykliksi (Kuva 6).



Kuva 6, Uuden tiedon luominen sisäisen ja ulkoisen tiedon muuttamissyklillä [Ojalan mukaella Nonakan ja Takeuchin mallista, Ota05, NoT95]

Kommunikoimalla yksilön sisäinen hiljainen tietämys saadaan ulkoistettua muille organisaation jäsenille. Jatkuvalle kanssakäymisellä organisaation jäsenet pyrkivät kiteyttämään yhteisen sisäisen hiljaisen tiedon helpommin levitettävään ulkoiseen muotoon. Tämä kiteytetty ulkoinen tieto voidaan sitten jakaa esimerkiksi koulutustilaisuuksien, tiimipalaverien tai erilaisten dokumenttien avulla. Prosessi jatkuu organisaatiojäsenten sisäistäessä tämän levitetyn tiedon,

soveltamalla sitä käytäntöön ja oppimalla samalla uusia asioita. Yksilön tuodessa nämä juuri opitun tiedon avulla hankkimansa uudet tiedot esiin sosiaalisella kanssakäymisellä on spiraali taas saavuttanut alkuasetelmansa. Jotta uuden tiedon syntyisykli toimisi mahdollisimman tehokkaasti, on työntekijöiden oltava motivoituneita ja vastaanottavaisia muutokselle sekä uuden oppimiselle. Tätä motivoituneisuutta yritys voi kehittää parantamalla työyhteisönsä hyvinvointia, joka on seuraavan kappaleen teemana. Työympäristö-kappaleessa sivutaan myös uuden tiedon syntyisykliä käytännön tasolla.

5.2.2 Hyvinvoinnin vaaliminen

Ohjelmistokehitys on asiantuntijatyötä, jossa henkinen hyvinvointi on esimerkiksi raskaaseen rakennustyöhön verrattuna tärkeämmässä asemassa kuin fyysinen kykenevyys. Ohjelmistoyrityksen tehokkuus ja kyky vastata kiristyneeseen kilpailuun ei piile sen omistamissa välineissä, vaan asiantuntevassa henkilöstössä, joka tuottaa välineiden avulla ohjelmistoja. Eritoten tämä korostuu asiantuntevuuteen ja tiimivetoisuuteen nojaavissa ketterissä ohjelmistokehitysmetodeissa, joissa muutos on jatkuvaa. Nykytutkimuksen valossa ketterillä ohjelmistoprojekteilla onkin heikot mahdollisuudet onnistua ilman keskiarvoa parempia taitajia, joilla on kompetenssia vastata jatkuvasti muuttuvaan työympäristöön [Con01, Boe02, Con01]. Highsmith kärjistää: "Vaikeisiin projekteihin parhaimmat taitajat ja vaativiin projekteihin keskimääräistä taitavampaa henkilöstöä" [Hig04]. Tämä tarkoittaa korkeaa riippuvuusastetta taitavista osaajista heidän muuttuessa lähes elintärkeiksi projektin onnistumisen kannalta. Koko projekti voi viivästyä tai kaatua sen yhden jäsenen uupuessa tai menettäessä motivaationsa. Kuten Beck jo XP-määrittelyissään totesi, hyvinvoinnin ja taloudellisen kannattavuuden välillä on side, jota tulee vaalia [BeA04].

5.2.2.1 Työuupuminen

Erityisen tärkeä näkökulma hyvinvointiin on työuupuminen (burn-out). Työuupumisen voidaan nähdä olevan puuttuvan hyvinvoinnin äärimmäinen päätetila. Ketterät metodit korostavat ohjelmistoprojektien monimutkaisuutta ja henkilön aktiivista roolia, jotka ovat tunnusomaisia henkisesti aktiiviselle työlle

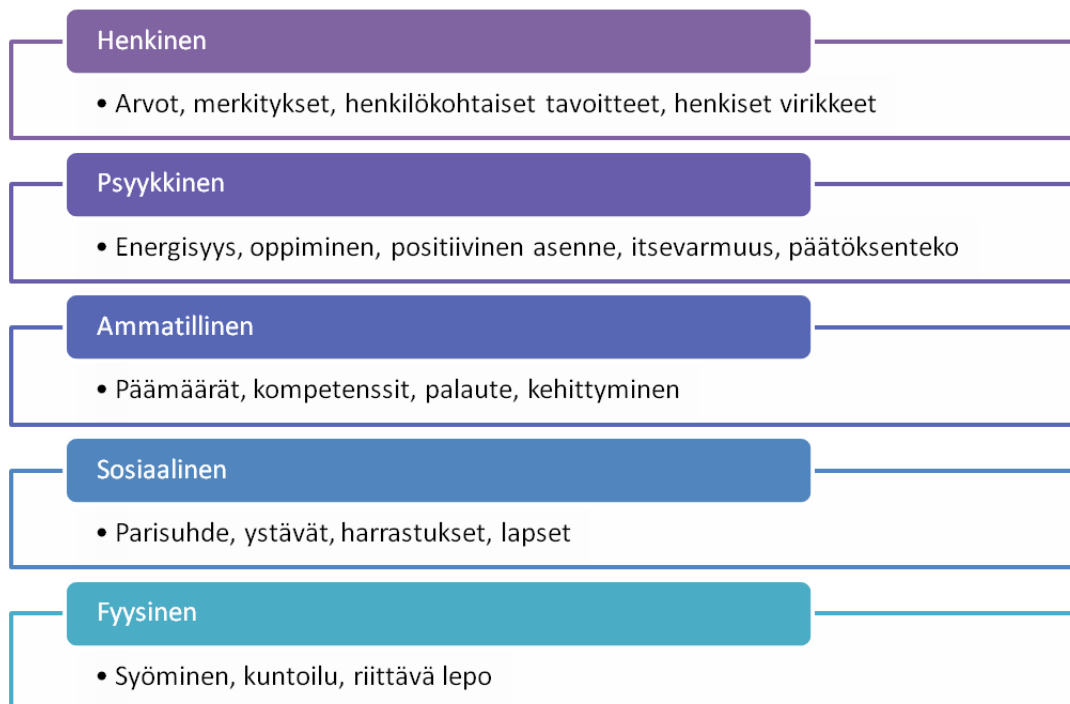
[ThK96]. Mikäli taas työntekijöille ei anneta mahdollisuutta vaikuttaa prosessiin, voi työ muuttua pian rasittavaksi. Samoin käy jos työntekijälle annetaan lisää töitä ilman, että hänelle annetaan tilaisuutta hankkia lisää osaamista [Ota05, ScB04]. On vaarana, että työ muuttuu liian vaikeaksi ja käy ahdistavaksi. Tällöin työntekijä kokee helposti itsensä riittämättömäksi työpaineen alla, minkä takia elimistö reagoi tähän stressitilalla [ScB04].

Stressin avulla keho pyrkii selviytymään kasvaneesta työpaineesta, mutta mikäli epätasapaino oppimismahdollisuuksien ja vaatimusten välillä säilyy samana, stressireaktiota seuraa uusi stressireaktio, eikä keho pääse palautumaan normaalitilaan. Tämän stressikierteen aikana henkilö käy yhä väsyneemmäksi, kyynisemmäksi, muutosvastarintaisemmaksi ja haluttomammaksi sitoutua työhönsä ja lopulta palaa loppuun [Ota05, ScB04].

Tunnusomaista tähän stressikierteeseen alttiille työntekijöille on heidän korkea tavoitetasonsa ja sitoutumisensa työhön, mikä on hyvinkin yleistä, ellei jopa vaadittua asiantuntijatyössä menestymiselle. Myös työpaikan ilmapiirillä kuten arvostuksen puutteella katsotaan olevan ratkaiseva vaikutus työuupumuksen syntyyn [ScB04]. Työuupumisesta toipuminen on hidas prosessi ja yrityksen kannalta hyvin kallis sairauspoissaolojen ja heikentyneen työtehon näkökulmasta [Ota05]. Loppuunpalaminen on usein merkki yritykselle siitä, että jotain on vialla ja siihen tulisi reagoida välittömästi. Pahimmassa tapauksessa ilmapiiri myrkyttyy entisestään henkilöstön levittäessä pahaa oloaan, työtehokkuus laskee ja sitä myötä yrityksen osaamispääoma sekä innovointikyky heikkenevät [Ota05]. Ketteriä metodeja toteuttavassa asiantuntijayrityksessä onkin ensiarvoisen tärkeää ymmärtää henkilöstön hyvinvoinnin merkitys yrityksen kilpailutekijänä ja minimoida mahdolliset työuupumiseen vaikuttavat osatekijät.

5.2.2.2 Hyvinvoinnin määritelmä ja sen tulkintaa

Hyvinvoinnin voidaan ajatella koostuvan viidestä osatekijästä: fyysisestä, ammatillisesta, sosiaalisesta, psyykkisestä ja henkisestä (Kuva 7) [Syd03].



Kuva 7, Hyvinvoinnin osatekijät [Syd03 mukailen]

Sydänmaalakka kutsuu tätä jaottelua yhdeksi näkökulmaksi henkilökohtaiseen johtajuuteen (self-leadership). Kun hyvinvointi on korkea kaikilla näillä osa-alueilla, henkilön suorituskyky on myös huipussaan. Ammatillinen hyvinvointi (professional condition) tarkoittaa työssä tarvittavan pätevyyden hallintaa, selkeitä työtehtäviä, palautteen saamista toiminnasta ja jatkuvaa kehittymistä. Fyysisellä hyvinvoinnilla tarkoitetaan tarpeellisen liikunnan, ruoan ja unen saamista. Psykkinen hyvinvointi tarkoittaa, että henkilö on energinen, utelias, nopeapäätöksinen ja opettelee uusia asioita. Sosiaalinen hyvinvointi pitää sisällään henkilön suhdetoimintakyvyt eli kuinka työyhteisön, puolison, lasten, sukulaisten, ystävien ja harrastusten parissa selvitään. Henkinen hyvinvointi sitoo nämä yhteen kuvaamalla henkilön oman elämän tasapainon eli tällöin henkilön päämäärät ovat tasapainossa ja henkinen virike ajaa eteenpäin elämässä. Teoria pohjautuu osittain jo XP-kappaleessa esitettyyn Maslowin tarvehierarkiaan, jota löyhästi noudattaen nämä viisi osa-aluetta voidaan jakaa pyramidiksi fyysisen hyvinvoinnin ollessa alimpana, sosiaalisen ja ammatillisen hyvinvoinnin sijoittuessa seuraavaksi, minkä jälkeen ylimmäiseksi jäävät psykkinen ja henkinen hyvinvointi. Maslowin hierarkian perusajatuksen mukaisesti ihminen voi saavuttaa hyvinvoinnin seuraavan tason vain edellisen

tason tarpeiden täytyttyä. Esimerkiksi ammatillista hyvinvointia ei tämän ajattelumallin mukaan voida saavuttaa mikäli fyysiset tarpeet, kuten unentarve ja ruoansaanti ovat tyydyttämättä. Tärkeänä yksityiskohtana Ojala lisää Maslowin tarvehierarkiajotteluun vielä turvallisuuden tunteen, kuten oman työpaikan säilymisen. Tämä turvallisuudentarve sijoittuu Otalan versiossa heti fyysisen hyvinvoinnin jälkeen seuraavalle Maslowin tarvehierarkian tasolle [Ota05]. On myös huomattava, ettei kaikissa tapauksissa Maslowimainen pyramidijattelu ole välttämättä totuudenmukaista. Kritiikkiä on esitetty koko hierarkisen pyramidirakenteen olemassaolosta ja sen selkeästä jaottelusta [WaB76]. Ehkä tämän takia Sydänmaalakka jakaa hyvinvointitasot "Maslowimaisesti", mutta jättää hierarkian pois [Syd03]. Tutkielmassa kuitenkin pitäydytään Otalan [Ota05] esittämässä pyramidimaisessa rakenteessa.

Tämän hyvinvointijaottelun perusteella voidaan päätellä, että melko suuri osa henkilön hyvinvoinnista riippuu hänen omasta toiminnastaan, mihin henkilökohtaisen johtamisen käsite viittaakin [Syd03]. Mikäli henkilö ei huolehdi liikunnasta, syö terveellisesti tai hänellä on ongelmia kotona, on näitä vaikea, ellei mahdotonta korjata ylemmillä tarvehierarkiapyramidin tasoilla. Organisaatio pystyy kuitenkin välillisesti ja joskus myös välittömästi vaikuttamaan matalan tason tekijöihin tarjoamalla esimerkiksi liikunta-aktiviteettejä, terveellistä ruokaa, mahdollisuudet työajan ulkopuoliseen sosiaaliseen kanssakäymiseen ja vähentämällä epävarmuutta työsuhteen jatkumisesta. Samoin tiimin vetäjä on ketterissä metodeissa lähempänä ystävää kuin perinteistä auktoriteettia, jolloin hän pystyy olemaan työntekijän tukena jo sosiaalisella tasolla hyvinvointipyramidissa. Tämä on myös yksi hyvä argumentti transformatiivisen johtajuuden puolesta ketterissä ohjelmistoprojekteissa.

Tarvehierarkiaa tarkemmin tutkimalla voidaan nähdä, että vesiputousmalli täyttää huonommin yksilön hyvinvointitarpeet kuin ketterät menetelmät, sillä verrattuna ketteriin metodeihin työntekijä harvoin pääsee jatkuvasti kehittämään itseään yhtä paljon, omaa osaamista ja ammattitaitoa ei arvosteta samalla tavalla, työyhteisö ja etenkin tiimi ei ole niin läheinen, sekä mahdollisuus vaikuttaa asioihin on pienempi. Toisin sanoen ketterät menetelmät tarjoavat jo lähtökohtaisesti paremman hyvinvointialustan kuin tiukan elinkaariajattelun mukainen vesiputousmalli. Toisaalta taas, kuten edellä on todettu, ketterissä metodeissa tiimin jäsenten hyvinvointi on ensiarvoisen tärkeää projektin onnistumiselle.

Voidaankin sanoa, että ketteriin metodeihin siirryttäessä yrityksen tulee pystyä huomioimaan nämä hyvinvoinnin osa-alueet ja ketterien metodien toimintamallien suhde niihin. Yritys voi näin parantaa yhteisönsä hyvinvointia kohdistamalla voimavaroja oikean toimintaympäristön luontiin, niin hallinnointi-, organisaatio- kuin teknisessä mielessä.

5.3 Organisaatorakenne

Kaikkein näkyvin ja helpoin tapa edesauttaa transformaatiota perinteisestä elinkaarimallista ketteriin prosessimalleihin on tarjota työyhteisölle sitä tukevat työskentelyvälineet ja toimintaympäristö. Organisaation rakenteen tulee tukea uutta ketterää prosessimallia. Mikäli yrityksen rakenne ja toimintamallit eivät tue ketteryyttä, on transformaatiolla huonot mahdollisuudet onnistua. Esimerkiksi palkkiojärjestelmän yksilökeskeisyys, informaatiovälityksen vaikeus tai turha byrokratia voivat muodostua ylitsepääsemättömiksi esteiksi ketterien metodien toimintatapaan tutustuttaessa. Samalla voidaan päätyä tilanteeseen, jossa ketterille metodeille otollisimmat, asiantuntivimmat työntekijät turhautuvat toimintaa hidastavasta byrokratiasta sekä kontrollista ja lopulta lähtevät yrityksestä. Tämä taas aiheuttaa helposti kierteen, sillä pätevien asiantuntijoiden lähtiessä tehottomuus kasvaa, minkä takia byrokratiaa ja kontrollia lisätään tehokkuuden ylläpitämiseksi. Pahimmillaan yritys voi tuhota tällä tavoin työyhteisönsä innovaatiokyvyn, motivaation ja lopulta koko kilpailukykyä [Hig04, Sch99].

Asiantuntijaorganisaatiolle sopivaa johtamismallia tarkasteltiin lähemmin jo johtajuuskappaleessa, seuraavaksi pureudutaan lähemmin organisaation rakenteeseen ja sen mahdollisuuksiin tukea asiantuntijayhteisön tavoitteita. Toimintaa tukevan työympäristön kehittäminen on luonnollisesti merkittävä osa transformaation tukemista. Työympäristön kehittämistä käydään pintapuolisesti läpi, jonka jälkeen tarkastellaan lähemmin palkitsemisjärjestelmän roolia työympäristön johtamisessa haluttuun suuntaan. Koska organisaatiomuutos on hyvin laaja aihe, tyydytään muutosta tarkastelemaan henkilöstön hyvinvoinnin ja oppimisen näkökulmasta. Tätä ajattelumallia ehkä parhaiten tukee oppivan organisaation teoria, johon pureudutaan kappaleen lopussa. Näin ollen organisaatiohierarkiat, kuten matriisi-, projekti- ja verkosto-organisaatiot, sekä erilaiset organisaatiostrategiat sivuutetaan. Muutosta käydään kuitenkin läpi yrityskulttuurin näkökulmasta omassa kappaleessaan.

5.3.1 Työilmapiiri

Hyvä työilmapiiri on tiiviisti sidoksissa henkilöstön osaamispääomaan, hyvinvointiin, jaksamiseen ja tehokkuuteen. Kuten hyvinvointikappaleessa todettiin, organisaatio voi vaikuttaa henkilön hyvinvointiin ja sitä kautta työilmapiiriin usein vain välillisesti tukemalla henkilön mahdollisuuksia kehittää itseään eri tarvehierarkian tasoilla. Erityisesti psyykkisiin ja henkisiin hyvinvointitasoihin ei organisaatiolla kuitenkaan ole yleensä suoria vaikutuskeinoja.

Fyysistä hyvinvointia voidaan parantaa käytännön työympäristöjärjestelyillä ja takaamalla säännölliset työajat. Useissa yrityksissä on myös mahdollisuus työajan ulkopuoliseen liikuntaan, joko järjestettyinä liikuntapahtumina, yhteistyönä läheisen kuntokeskuksen kanssa, tai yrityksen itse kustantamissa liikuntatiloissa. Yritys voi myös tarjota toimivan lounasruokailun lisäksi esimerkiksi lounasseteleitä tai vastaavia taatakseen riittävän energiansaannin työntekijöilleen.

Sosiaaliseen hyvinvointiin voidaan vaikuttaa tiimitoiminnan ja muun yhteisöllisyyden tukemisella, sekä niiden hienovaraisella ohjauksella. Tämä on myös edellytys ketteriin prosesseihin siirryttäessä, joiden toiminta perustuu täysin tiimidynamiikkaan. Varsinkin prosessimuutoksen alussa on tärkeitä mahdollistavia henkilöiden liikkuvuus niin, että saadaan samanhenkiset työntekijät samoihin tiimeihin parhaimman yhteistoiminnan aikaansaamiseksi. Näin esimerkiksi XP-prosessimalliin siirryttäessä on hyvät lähtökohdat yhteisten arvojen ja periaatteiden omaksumisessa. Tiimitoimintatuen lisäksi hyvin tärkeä osa sosiaalista hyvinvointia on turvallisuuden tunne. Työntekijän näkökulmasta tämä tarkoittaa varmuutta työsuhteen jatkumisesta [Ota05, BeA04]. Kuitenkin jos yrityksen historia perustuu tiukan peräkkäiseen vesiputousmalliin, ovat työvälineet ja työympäristö todennäköisesti muotoutuneet hyvin byrokraattisiksi ja rutiininomaisiksi. Tämä tarkoittaa usein sitä, että jo pelkän prosessin ylläpitämisen ja hallinnan ympärille on muodostunut ns. suojatyöpaikkoja, joista tulisi päästä eroon. Esimerkiksi vesiputousmallissa yleisesti ohjelmointiryhmästä eriytetty testausryhmä kävisi ketteriin metodeihin siirryttäessä osittain, ellei kokonaan turhaksi. Tämä luonnollisesti aiheuttaa vastareaktion, koska pelätään

töiden menettämistä. Kuitenkaan näin ei tarvitse olla, sillä yleensä jokainen työntekijä voi olla jollain osa-alueella keskiarvoa pätevämpi [Hig04]. Haasteena on näin löytää näille henkilöille uudet työtehtävät. Aina tämä ei kuitenkaan ole mahdollista ja ketteriin metodeihin siirryttäessä onkin tunnusomaista organisaatorakenteen keventyminen sekä yrityksen tehokkuuden kasvaminen, mikä usein tarkoittaa henkilöstön vähentämistä [Hig04]. Toisaalta voidaan ajatella, että kiristyvässä kilpailussa koko yrityksen toiminta kävisi ennen pitkää kannattamattomaksi, ellei leikkauksia tehtäisi. Tehostus on kuitenkin hyvin haastava osa-alue, koska samalla tulisi taata henkilöstölle turvallinen toimintaympäristö ilman pelkoa työpaikkojen menetyksestä.

Ammatilliseen hyvinvointiin organisaatio voi vaikuttaa tasapuolisella ja oikeudenmukaisella toiminnalla. Samoin riittävät kehityskeskustelut ja työilmapiiritutkimukset auttavat organisaatiota mukautumaan kohti oikeaa työilmapiiriä. Tärkeätä näiden välineiden käytössä on kuitenkin työntekijöiden todellinen mahdollisuus vaikuttaa työilmapiiriin ja sitä kautta yhteiseen hyvinvointiin. On helppoa saada aikaan kyyninen ja epäluottava ilmapiiri, mikäli kehityskeskusteluja ja työilmapiiritutkimuksia tehdään vain näön vuoksi. Työntekijän ammatillisen hyvinvoinnin tunnuspiirteisiin eli kehityshaluihin ja motivoitumiseen voidaan matalammissakin organisaatioissa vastata työtehtävien kierrätyksellä. Näin tuetaan samalla ketterien metodien vaatimaa monitahoista asiantuntijuutta ja voidaan välttää avainosaajien dominoiva rooli tietyissä työtehtävissä, sekä vähentää henkilöresurssivajauksen aiheuttamaa riskiä [NMM05]. Työtehtävien kierrätys on myös ihanteellinen tapa luoda uutta tietoutta sisäisen ja ulkoisen tietämyksen spiraalimallin mukaan, mikä entisestään vahvistaa yrityksen valmiutta siirtyä kohti ketteriä metodeja [NoT95].

Hyvinvointitarvehierarkian mukaisesti työntekijä pystyy saavuttamaan seuraavan tason vain edellisten tasojen ollessa tyydytettyinä. Esimerkiksi ammatillista hyvinvointia on vaikea saavuttaa, mikäli työyhteistön sosiaalinen toiminta on välttävää. Erityisesti tämä korostuu hyvinvointihierarkian korkeimmilla tasoilla eli psyykkisellä ja henkisellä tasolla, joihin yritys ei pysty suoraan vaikuttamaan. Esimerkiksi psyykkisen tason energinen, itsevarma ja positiivinen työasenne on seurausta fyysisestä, sosiaalisesta ja ammatillisesta hyvinvoinnista. Samoin henkisen hyvinvoinnin tasapainoisen arvomaailman ja selkeän elämäntavoitteen saavuttaminen on alempien hyvinvointitasojen summa.

Erityisesti XP:ssä tämän hyvinvointimekaniikan merkitys korostuu, sillä sen toiminta lähtee arvoista, jotka ovat Sydänmaalakan tulkinnessa ylimmällä hyvinvointihierarkian tasolla [Syd03]. Siirtyessään kohti ketterää toimintamallia organisaation tulee näin ollen hienovaraisesti tukea oikealla tavalla kaikkia alempia hierarkian tasoja, jotta työntekijälle muodostuu haluttuja arvokäsityksiä henkisellä hyvinvointitasolla. Nämä ketteryyttä tukevat arvokäsitykset luovat sitten tulevan yrityskulttuurin, joka on koko toiminnan perusta. Tästä näkökulmasta XP ja Scrum poikkeavat merkittävästi toisistaan. Scrumissa kehoitetaan lähtemään prosessimallimuutokseen käytännön toiminnalla, kun taas XP:ssä korostetaan uutta toimintamallia tukevan työilmapiirin rakentamista. Toki etenemistavat ovat yrityksen itse päätettävissä, eivätkä Scrum ja XP pakota tiettyyn muutosprosessiin, mutta vain XP tarjoaa selkeän kuvauksen ketterää toimintalogiikkaa tukevasta arvomaailmasta ja periaatteista. Jokainen organisaatio on kuitenkin ainutlaatuinen omine haasteineen ja vahvuuksineen, mitä käsitellään lähemmin yrityskulttuuri-kappaleessa.

5.3.2 Käytännön työympäristö

Käytännön työympäristön on myös vastattava uusiin haasteisiin. Yksityiset työhuoneet tulisi muuttaa paremmin tiimitoimintaa tukeviksi avokonttoreiksi yhteisine ryhmätyötiloineen, virkistäytymismahdollisuuksineen ja kommunikaatiovälineineen. Beck [BeA04] jopa esittelee XP:tä parhaiten tukevan työtilan, jossa tietokoneet on järjestetty pareittain kahteen kolmen parin kolmiomuodostelmaan, jonka ympärillä on yksittäisiä työpisteitä yksityisyyttä vaativiin töihin sekä koko tilaa hallitseva tarinataulu (story board). Muistettava toki on, että kokonaisuuden tulee kannustaa tiimitoimintaan, mutta samalla taata myös tarvittava työrauhan. Tässä korostuu ketterien metodien pienien tiimien painotus, sillä suurten tiimien mahdollistaminen samaan tilaan on tehtävänä haastava riittävästä työrauhasta ja kehittävästä yhteistyöstä puhumattakaan. Samalla on huomioitava asiakkaiden ja tiimin muiden sidosryhmien mahdollinen ellei jatkuva läsnäolo osana toteutusprosessia. Nopean kehitystahdin takia kaikkien nähtävillä tulisi olla mahdollisimman reaaliaikainen kuva prosessin kulloinestakin tilasta esimerkiksi magneettitaululla tai valkokankaalla. XP:ssä yksi ehdotettu tapa esittää projektin eteneminen iteraation aikana on asiakkaan

toivomuksien eli tarinoiden toteutumismuutos ja Scrumissa päivittäinen tilannekaavio (burn down chart) [BeA04, ScB02].

Teknisessä mielessä sovellusprojektien yhteisen säilytyspaikan (repository) merkitys korostuu ketterissä metodeissa. Vesiputousmallissa henkilöille yleensä nimetään työtehtävät projektin alussa, joita he noudattavat projektin loppuun asti. Tehtäväjako on useimmiten helpointa tehdä arkkitehtuuri- ja rajapintasuunnittelun jälkeen, jolloin henkilölle voidaan määritellä selkeä vastuualue, josta hän vastaa. Periaatteessa muut projektin jäsenet eivät siis ole vesiputousmallissa niin riippuvaisia toisistaan kuin ketterissä prosessimalleissa. Ketteriin metodeihin siirryttäessä tilanne muuttuu radikaalisti, sillä kehityssyklit ovat lyhyitä, työtehtävät voivat vaihtua tiuhaan ja testausta tehdään jatkuvasti. Scrum määrittelee 8-16 tunnin tehtäväjaottelun toteutusvaiheen aikana, ja XP esitteli testausvetoisen kehityksen, jossa toteutus perustuu esikirjoitettuihin testeihin. XP esitteli myös kymmenen minuutin testikäytännön, jossa kaikki automatisoidut testit tulee voida ajaa kymmenen minuutin sisään. Käytännössä tämä asettaa kovia vaatimuksia toimintaa tukevalle tekniikalle. Muutokset on saatava yhteiseen säilytyspaikkaan mahdollisimman nopeasti ja kaikenkattavat testit, joita siis voi olla tuhansia, on pystyttävä ajamaan mahdollisimman nopeasti läpi jokaisen muutoksen jälkeen.

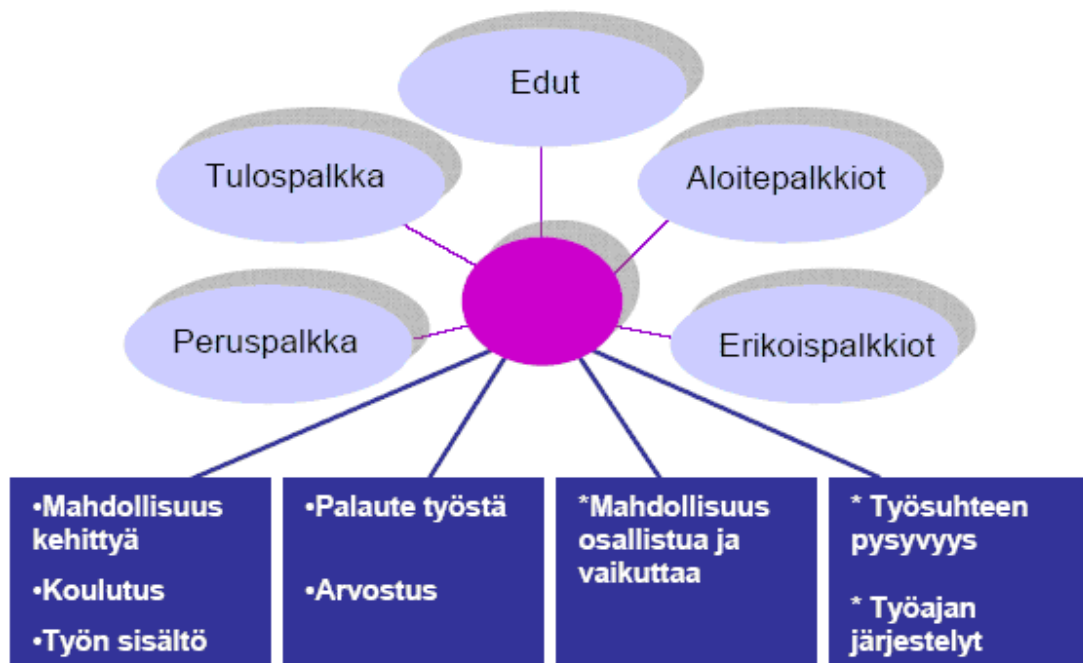
Muita teknisiä sosiaalista kanssakäymistä tukevia apuvälineitä ovat erilaiset elektroniset yhteydenpitovälineet, kuten keskustelupalstat, blogit, pikaviestimet, wiki, videokeskustelulaitteet sekä yhteiset "projektihiikkalaatikot" eli projektin sisäiset testaus- ja kokeiluympäristöt. Näin esimerkiksi fyysisesti tiimistä erillään olevat sidosryhmät voivat ottaa helpommin osaa prosessiin ja saada helposti ajantasaista tietoa projektin etenemisestä.

5.3.3 Palkitsemisjärjestelmät

Kannustimet ovat tärkeä osa työyhteisön motivointia. Etenkin ketterien prosessimallien mukaisissa työyhteisöissä, joissa organisaatorakenne on matala, on palkitseminen käynyt entistä tärkeämmäksi vähäisempien urakehitysmahdollisuuksien takia. Itse palkitsemistapoja on useita aina normaalista markkinapalkkauksesta palveluvuosilisiin ja osakepalkkioihin. Palkitsemisen tehokkuus riippuu kuitenkin paljon kohteen olevasta yksilöstä.

Toinen arvostaa lastenhoitomahdollisuutta ja toinen taas joustavia työaikoja. Monella isolla yrityksellä onkin käytössään niinsanottu palkitsemisen voileipäpöytä (cafeteria plan), joista työntekijät voivat valita mieleisensä palkitsemistavat [Ota05]. Palkitsemistavasta riippumatta tärkeintä on kuitenkin palkitsemisperusteiden kohdennus siten että ne tukevat yrityksen toimintatapaa.

Asiantuntijaorganisaatiossa palkitsemisen tulisi liittyä tiiviisti osaamiseen ja tiimitoimintaan. Esimerkiksi rakennusalalla yleiset urakkapalkat sopivat huonosti asiantuntijayritykseen, sillä ne eivät sisällä elementtejä, jotka kannustaisivat auttamaan muita työntekijöitä tai jakamaan informaatiota [Uus02]. Samoin yleisesti käytettyjä yksilökeskeisiä palkitsemisjärjestelmiä tulisi välttää, koska ne sotivat koko tiimitoiminnan filosofiaa vastaan [Hig04, Ota05, BuH06]. Osaamisen palkitsemista on tutkittu laajalti Teknillisen korkeakoulun Työpsykologian ja johtamisen laboratoriossa. Tutkimuksen yhtenä tuloksena eri palkitsemistavat on jaoteltu aineettomiin ja aineellisiin motivaatiotekijöihin (Kuva 8).



Kuva 8, Palkitsemisen jaottelu aineellisiin ja aineettomiin tapoihin [VHH98]

Tutkimuksissa on todettu, että aineelliset palkkiot vähentävät työntekijän tyytymättömyyttä vain hetkellisesti, kun taas aineettomat palkitsemistavat saavat ihmisen kehittämään työtään ja motivoitumaan paremmin työhönsä

[VHH98, Ota05]. Asiantuntijaorganisaation tulisikin tarjota perinteisten aineellisten palkkioiden lisäksi myös aineettomia palkkiotapoja, niin henkilö-, tiimi- kuin koko organisaation menestyksen perusteella.

Palkitsemisen merkitys korostuu kun se yhdistetään työhyvinvointiin. On todettu, että sopivan haasteelliseksi koettu työ, palautteen saaminen työstä sekä tyytyväisyys rahalliseen palkitsemiseen voivat lisätä yksilön työhyvinvointia [ScB04]. Erityisesti kun tavoitteena on sitoutunut ja hyvinvoiva henkilöstö aineettomilla palkitsemismetodeilla saavutetaan parempia tuloksia, kuin aineellisilla palkinnoilla, kuten palkalla [Ota05, VHH98].

5.3.4 Oppivan organisaation teoria

Oppivan organisaation käsite syntyi Chris Argyriksen ja Donald Schönin tutkimusten (1974–1982) pohjalta, mutta se popularisoitui vasta 1990-luvulla alussa [BuH06]. Oppivan organisaation mallia on usein käytetty työyhteistöjen toiminnan ja ilmapiirin kehittämässä [Ota05]. Oppivan organisaation määritelmä on ideaalinen tavoite, johon asiantuntijayrityksen tulisi pyrkiä. Käytännössä täydellistä oppivan organisaation esimerkkiyritystä tuskin löytyy, mutta sen määrittelyn pohjalta näkee mihin suuntaan toimintaa voisi kehittää. Erityisen moni määrittely osuu yhteen tietämyksenhallinnan tavoitteiden kanssa ja oppivaa organisaatiota pidetäänkin yhtenä idealistisena tapana toteuttaa tietämyksenhallinnan päämäärät [BuH06, Wiki10]. Koska ketterien metodien ja tietämyksenhallinnan symbioosi on jo aikaisemmin todettu, ei liene yllätys, että oppivan organisaation ominaisuudet käyvät niin hyvin yksiin ketterän liiketoimintamallin vaatimusten kanssa (Kuva 9). Itse asiassa ketterissä ohjelmistoyrityksissä vaaditaan jonkun tason organisaatiotason oppimistukea, sillä ketterät projektit ovat yleensä nopeatempoisia, jatkuvaa uuden oppimista ja muutosten keskellä elämistä, niin tekniikan kuin asiakasvaatimustenkin näkökulmasta. Tämän takia jokaisen ohjelmistoyrityksen, joka haluaa menestyä pitkällä aikavälillä, täytyy käyttää jo opittua tietoutta uusien samantapaisten projektien läpiviemiseen [ChM04]. Tämä vaatii tietämyksen jakamista ja organisaation tukea oppimisille [ChM04].

Oppiva organisaatio

- Oppiva organisaatio on työ- tai muu yhteisö, joka osaa jatkuvasti arvoida omaa toimintaansa ja verrata sitä ympäristön muutoksiin sekä uusia omaa toimintaansa ennakoivasti ympäristön vaatimusten mukaan.
- Kaikki organisaation jäsenet ottavat osaa strategian suunnitteluun, päätöksentekoon, arvomäärittelyihin ja ristiriitojen selvittelyyn
- Informaatioteknologiaa käytetään tiedon tasapuoliseen jakamiseen organisaatiossa ja auttamaan "eturintaman" työntekijöitä tekemään itsenäisiä päätöksiä.
- Raportoinnin, budjetoinnin ja kirjanpidon tarkoituksena on auttaa ihmisiä ymmärtämään organisaation taloudellista liiketoimintaa.
- Yksiköt sekä osastot toimivat asiakkaina ja toimittajina yhtiön sisäisessä toimitusketjussa oppien toinen toisiltaan.
- Palkitsemisjärjestelmät ovat joustavia ja jaoteltu aineellisiin sekä aineettomiin osa-alueisiin vastatakseen mahdollisimman hyvin yksilön tarpeisiin ja toiminnan tehokkuuteen.
- Organisaatiokaaviot, rakenteet ja prosessit ovat vain väliaikaisia ja voivat muuttua vastaamaan paremmin muuttunutta toimintaympäristöä.
- Kaikki tahot, jotka ovat tekemisissä asiakkaiden, toimittajien ja loppukäyttäjien kanssa, nähdään arvokkaina informaatiolähteinä.
- Organisaatio oppii muilta organisaatioilta yhteisyritysten ja muiden yhteistoimintatapojen kautta.
- Johtajien pääasiallisena tehtävänä on tukea kokeilemista ja uuden oppimista kannustamalla, kyselemällä ja antamalla palautetta.
- Työntekijöille tarjotaan oppimista tukevat järjestelmät, työkalut ja työympäristö, jotta he voivat ottaa vastuun omasta oppimisestaan.
- Kierrätetään osaamista ja osaajia jotta kaikesta osaamisesta saadaan mahdollisimman suuri hyöty.

Kuva 9, Oppivan organisaation määritelmä [PBB97, Ota05 mukailen]

Suurin osa oppivan organisaation määrittelyistä on triviaaleja itsensäselittäviä lausahduksia. Listasta voidaan kuitenkin nähdä selkeät yhteydet henkilöstön

hyvinvointiin ja transformatiiviseen johtamiseen. Oppiva organisaatio pyrkii tehostamaan sisäisen tietämyksen ulkoistamista ja XP:n ajattelumallin mukaisesti edistää muutosta ”pehmeästi” arvomaailmaan vaikuttamalla. Henkilöstön hyvinvoinnin rooli on luettavissa rivien välistä, sillä työntekijän sosiaalisten ja ammatillisten kehitystarpeiden täyttäminen on hyvin esillä. Kaiken tämän ohjaamiseen vaaditaan byrokraattisen kontrolloinnin sijaan kevyttä valmentaa otetta transformatiivisen johtajuuden tapaan. Oppivan organisaation määritelmän ja ketterän liiketoiminnan yhtäläisyydet voidaan myös kääntää niin, että ketterää liiketoimintaa soveltava yritys mukautuu luontevasti oppivan organisaation ajattelumallin, koska sen kohtaama liiketoimintaympäristö vaatii jatkuvaa uuden oppimista ja innovatiivisuutta [MAL05].

Oppivaa organisaatio on käsitteenä saanut kritiikkiä muun muassa siitä, että oppiminen tarkoittaa muutosta, erilaisuutta ja epäjärjestyneisyyttä, kun taas organisaatio terminä käsittää järjestystä, rakennetta ja vakautta. Miten siis nämä kaksi käsitettä on saatu yhdistettyä? Peter Senge [Sen90] järjestää oppivan organisaation periaatteet tukemaan järjestyksen saavuttamista kirjassaan ”The Fifth Discipline”, jota pidetään yhtenä oppivan organisaation merkittävimmistä popularisointiteoksista [BuH06]. Teorian pääajatuksena on oppimismahdollisuuksien tarjoaminen kaikille työskentelytasoille, jolloin työntekijöiden suurempi mahdollisuus vaikuttaa ja ottaa vastuuta aikaansaavat lisääntyvää sitoutumista, joustavuutta ja luovaa ajattelua [Sen90].

Vaikka oppivan organisaation malli vaikuttaakin melko ihanteelliselta ratkaisulta eritoten ketterää ohjelmistoliiketoimintaa harrastavalle yritykselle, on se pitkstä historiastaan huolimatta edelleen lähinnä idealistinen päämäärä innovaatiutta ja joustavuutta kaipavaalle organisaatiolle [BuH06, MAL05]. Ongelmia aiheuttaa erityisesti yrityksen yhteisen päämäärän mielessä pitäminen kaiken oppimisen ja muutoksen keskellä. Hieman nurinkurisesti oppivassa organisaatiossa vaaditaankin yleensä joku yksittäinen vetäjätaho, joka pitää huolen, että projektit etenevät ja eri kehityshankkeet tukevat toisiaan [MAL05, Ota05], mutta kuten edellä on todettu, oppivan organisaation malli on kuitenkin hyvä esikuva ja päämäärä ketterien prosessimallien kaltaiselle henkilöstökeskeiselle liiketoiminnalle.

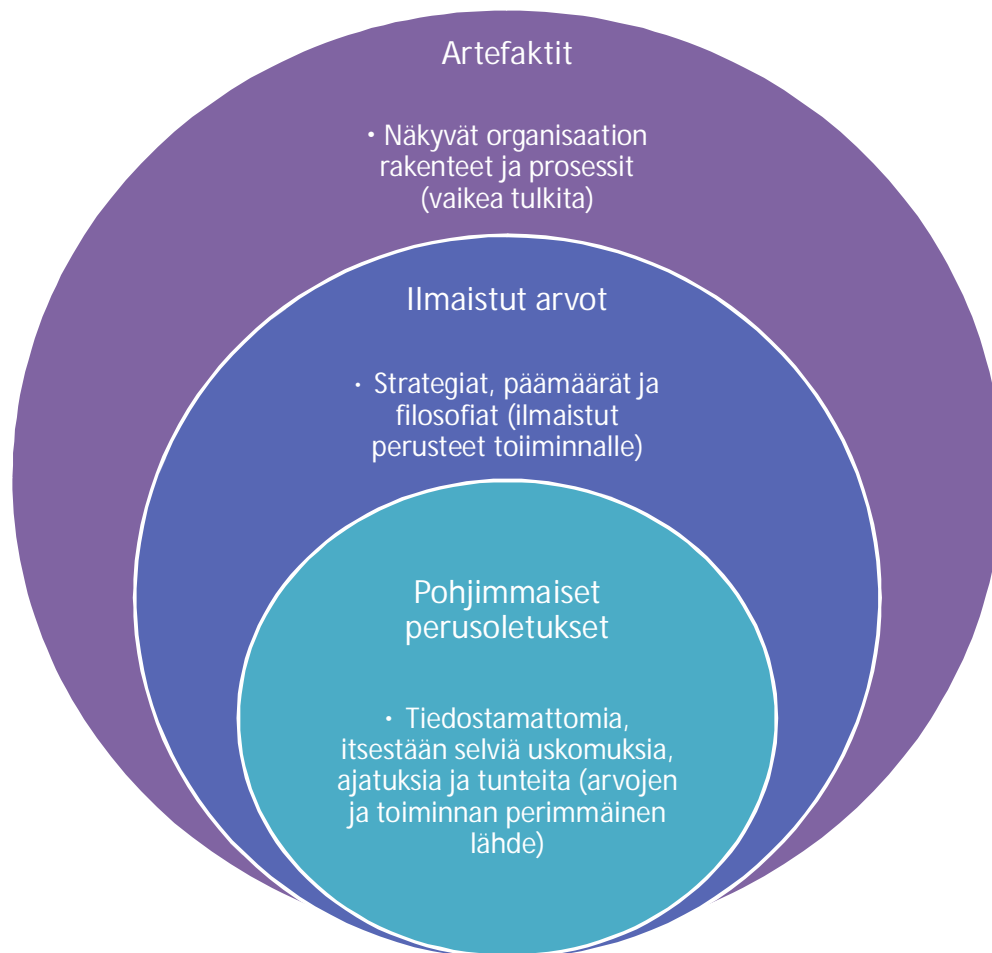
6 Yrityskulttuurin näkökulma muutokseen

Edellä on käyty lyhyesti läpi pientä osaa yrityksen haasteista sen siirtyessä byrokraattisesta elinkaarimallin mukaisesta liiketoiminnasta kohti joustavampaa ketterää prosessimallia. Aiheeseen käsitellessä melkein kaikkia liiketoiminnan alueita aina yrityksen strategiasta, sen teknisiin työvälineisiin, on tarkastelunäkökulma yritetty pitää hyvin tiukkaan rajatulla henkilöstöalueella. Tämä on myös se tärkein osa-alue ohjelmistoliiketoiminnasta ja erityisesti ketteristä metodeista puhuttaessa, sillä kuten aikaisemmin on todettu, ketterä ohjelmistoliiketoiminta on tiukasti osaavan ja muutosmyönteiseen henkilöstön varassa. Muutokseen on siis lähdettävä henkilöstön ehdoilla, sillä pakotettu muutos ei pelkästään sodi itse ketterää ajattelumallia vastaan, vaan myös pitkällä aikavälillä saattaa aiheuttaa vain enemmän sekaannusta kuin hyötyä [Sch99]. Erityisesti XP on alun perin suunniteltu tätä muutosta silmällä pitäen henkisen pääoman ja hyvinvoinnin näytellessä keskeistä osaa sen määrittelyssä. Scrum soveltunee ehkä parhaiten hieman hienovaraisempaa muutosta hakevalle yritykselle, jonka tarkoituksena on hakea joustavuutta luopumatta suuresta osasta vanhaa toimintamallia. Toki myös Scrumin mukainen toimintamalli voi olla lähellä XP:n ideologiaa, mutta Scrum antaa paremman mahdollisuuden hienovaraiseen siirtymiseen esimerkiksi toimimalla vain "käärepaperina" perinteiselle vesiputousmallille. Yhteistä kummallekin on kuitenkin niiden ketteryysmanifestin mukainen painotus eli toimintaa dokumentoinnin kustannuksella, yksilöt ja interaktio ennen prosesseja ja työkaluja, asiakasyhteistyö ennen sopimusneuvotteluita ja muutokseen vastaaminen suunnitelmissa pysyttämisen sijaan [Bec01]. Koska lähtökohtana tutkielmassa pidetään byrokraattista elinkaarimallin mukaista toimintaprosessia Macrosoftin esimerkin mukaan, siirtyminen ketterään ajattelumalliin on yritykselle vaativa ja aikaavievä transformaatio. Edellä on muutosta pyritty kuvaamaan eri näkökulmista, niin johtamisen, tietämyksenhallinnan, kuin organisaatorakenteenkin kannalta. Itse muutosprosessia ei kuitenkaan ole vielä tarkasteltu lähemmin, vaan sitä sivuutettiin pikaisesti työilmapiiri-kappaleen lopussa. Tämän kappaleen tarkoituksena on kuvata koko muutosta yrityskulttuurin näkökulmasta, jonka ymmärtäminen on olennainen osa lähes kaikissa yrityksen kohtaamissa toimintaympäristön muutoksissa.

6.1 Yrityskulttuurin määritelmä

Yrityskulttuurin isänä pidetään organisaatiopsykologian alan perustajaa Edgar H. Scheinia, jonka yrityskulttuuria käsittelevään kirjaan [Sch99] tämä kappale pitkälti perustuu.

Yrityskulttuurimuutosta tutkittaessa on aluksi ymmärrettävä mistä yrityskulttuuri muodostuu. Schein jakaa yrityskulttuurin kolmeen tasoon, jotka ovat artefaktit, ilmaistut arvot ja pohjimmaiset perusoletukset (Kuva 10). Tasot ovat kaksisuuntaisesti sidoksissa keskenään muodostaen jatkuvasti muotoutuvan perustan yrityksen toiminnalle.



Kuva 10, Yrityskulttuurin kolme tasoa [Sch99]

6.1.1 Artefaktit

Artefaktit ovat yrityskulttuurin kaikkein näkyvin osa. Esimerkiksi ketterässä liiketoiminnassa artefakteja voisivat olla avoimet tiimitilat, epäformaali käytös, voimakkaasti välittyvä yhteenkuuluvuuden tunne ja nopeatahtinen toiminta. Perinteisen byrokraattisessa suunnitteluvetoisessa organisaatiossa taas artefakteina voisivat olla suljetut tilat ja yleinen vaikutelma harkitsevasta sekä hitaasta etenemisestä. Yrityskulttuurin näkökulmasta on tärkeää ymmärtää, etteivät artefaktit kerro varsinaista syytä vallitsevalle toimintatavalle [Sch99]. Vaikka olisikin helppoa määritellä yrityksen liiketoimintamalli ketteräksi artefaktiensa perusteella, voivat ulkoiset puitteet olla vain kulissi toisenlaisille arvoille ja ajattelumalleille. On ymmärrettävä miksi työntekijät toimivat nykyisellä tavalla.

6.1.2 Ilmaistut arvot

Ilmaistut arvot ovat yleisesti sovittua toimintatapoja ja arvomalleja, jonka mukaan yritys pyrkii toimimaan [Sch99]. Esimerkiksi XP:n mukaista liiketoimintaa harrastava yritys voi kertoa noudattavansa kommunikoivaa, kunnioittavaa ja humaania liiketoimintamallia, joilla perustellaan näkyviä artefakteja kuten avokonttoreita. Nämä ilmaistut arvot ovat yleensä omalla tavallaan myös artefakteja, sillä useimmilla yrityksillä on kaavioksi ja "huoneentauluiksi" kirjoitettuja dokumentteja, jotka pyrkivät kuvaamaan minkälaiseen toimintaan yritys pyrkii. Paradoksaalisesti myös traditionaalista, kontrolloivaa elinkaarimallia soveltavalla yrityksellä voi olla XP:n mukaiset arvot, vaikka ulkoiset artefaktit ovatkin täysin erilaiset. Esimerkiksi elinkaariprosesimallin mukainen yritys voi kertoa arvokseen asiakasuuntuneisuuden, mutta käytännössä toiminta poikkeaa huomattavasti ketterien metodien asiakaskeskeisyydestä. Näin ollen ilmastujen arvojenkaan perusteella ei vielä voida päätellä yrityksen todellista arvomaailmaa. On mentävä syvemmälle ja tutkittava alla piileviä yhteisiä oletuksia.

6.1.3 Pohjimmaisets perusoletukset

Artefaktien ja ilmaistujen arvojen tuottamat ristiriitaisuudet kertovat, että ajattelun ja käsitysten syvempi taso ohjaa näkyvää käyttäytymistä [Sch99]. Jotta tämä syvempi taso pystyttäisiin ymmärtämään, on organisaatiota tarkasteltava historiallisesta näkökulmasta. Tämä voidaan kiteytyä kysymykseen: Mitkä ovat olleet yrityksen historian merkittävimpien perustajien ja johtajien näkemykset, arvot ja uskomukset, jotka ovat tehneet sen menestyksekkääksi? Seuraavaksi kuvataan tämä nuoren yrityksen kulttuurin syntyprosessi kuvitteellisesti byrokraattista elinkaarimallia soveltavan esimerkkiyrityksen Macrosoftin näkökulmasta.

Oletetaan, että ohjelmistoyrityksen perustajana on kontrollia ja järjestystä kannattava henkilö. Hänellä on selvä visio siitä, kuinka hän haluaa ohjelmistoja tuotettavan ja kuinka prosessista saadaan mahdollisimman tehokas, että kannattava. Hän on myös palkannut alaisikseen tähän toimintamalliin hänen mielestään parhaiten sopivat työntekijät. Jotta yrityksen toiminta voisi jatkua ja kehittyä, tulee näiden toimintamallien olla tarpeeksi tehokkaita tuottaakseen yritykselle tarpeeksi tuloja toiminnan ylläpitämiseen. Luonnollisesti yrityskulttuuri yrityksineen kaatuu, mikäli näin ei ole. Yrityksen kasvasessa ja menestyessä työntekijät omaksuvat pikkuhiljaa nämä perustajan kurinalaista toimintaa kannattavat arvot, sillä ne aletaan nähdä ainoana oikeana tapana toimia yrityksen toimintaympäristössä. Mikäli yritys pysyy menestyvänä, tämä arvomaailma muuttuu työntekijöille itsestään selväksi sisäiseksi tietämykseksi, eivätkä he lopulta pysty suoraan sitä kuvaamaan sen enempää, kuin kalat vettä ympärillään. Näin esimerkin Macrosoftissa ketterä liiketoimintamalli nähtäisiin vain yksinkertaisesti vääränä tapana toimia heidän toimintaympäristössään.

Näin opittu näkymätön arvomaailma vaikuttaa siten kaikkeen yrityksen toimintaan. Mikäli yrityksen ilmaisemat arvot poikkeavat paljon pohjimmaisista perusolettamuksista, kohdataan edellä mainittu ristiriita ulkoisten artefaktien ja ilmaistujen arvojen välillä. Samalla on huomattava, ettei yrityksellä varsinaisesti ole oikeaa tai väärää kulttuuria. Yrityksen toimintaympäristö määrää sille parhaiten sopivan toimintamallin ja rakenteen. Esimerkiksi sairaalaohjelmistoja tuottavalle yritykselle oppivan organisaation malli olisi todennäköisesti liian järjestäytymätön ja kuriton, niin luotettavuuden, kuin asiakassuhteidenkin näkökulmasta. Kuitenkin, vaikka yritys toimitaisikin menestyksekkäästi, ei sen yrityskulttuuri ole välttämättä kaikkein optimaalisin yrityksen

toimintaympäristöön nähden. Tätä korostaa ohjelmistoteollisuudessa jatkuva toimintaympäristön muutos, mikä pakottaa yrityksen aika ajoin uudelleenarvioimaan sen toimintamallia. Näin vuosikaudet toimivaksi koettu vesiputousmalli voi käydä liian jäykäksi ja tehottomaksi kilpailijoiden siirtyessä joustavampiin ketteriin metodeihin.

6.2 Transformatiivinen muutos

Kulttuurimuutoksessa voidaan yrittää pakottaa ihmisiä toimimaan uudella tavalla, mutta tällainen pakotettu toimintamallin muutos ei kuitenkaan ole pysyvää elleivät alla olevat näkymättömät perusolettamukset ole kokeneet mitään muutosta. Ennen kuin voidaan tarkastella lähemmin muutosta itse yrityksen kannalta, tulee henkilön käymä sisäinen muutosprosessi ymmärtää. Tätä muutosprosessia Schein kutsuu transformatiiviseksi muutokseksi [Sch99]. Scheinin transformatiivinen muutoskehys jaotellaan kolmeen toimintavaiheeseen, jotka ovat motivaation luominen muutokselle, uusien käsitteiden ja vanhojen käsitteiden uusien merkitysten oppiminen, sekä uusien käsitteiden ja merkitysten sisäistäminen

6.2.1 Motivaation luominen muutokselle

Muutos tarkoittaa uuden oppimisen lisäksi myös sellaisten asioiden poisoppimista, jotka ovat jo olemassa ja mahdollisesti esteenä uuden oppimiselle. Koska ihminen vastustaa luonnostaan muutosta ja pyrkii etsimään tasapainoa ja vakautta, tarvitaan muutoksen alkuunpanijaksi joku kyseenalaistava voima [BuH06, Sch99]. Näitä voimia voivat esimerkiksi olla taloudelliset uhkatekijät, teknologisen jälkeenjäämisen vaarat, fuusiot, yritysostot, koulutuksen tuoma uudistunut ajattelumalli ja sisäinen tyytymättömyys omien ihanteiden sekä tavoitteiden suhteen.

Kun henkilö on hyväksynyt kyseenalaistavat tiedot tarpeeksi suurina voimina muutokseen lähtemiselle, kokee hän joko eloonjäämisahdistusta tai syyllisyyttä. Eloonjäämisahdistuksen Schein kuvaa "en selviydy jossain mielessä ellen muutu"-tyyppisenä voimana, kun taas syyllisyyden hän kuvaa "en saavuta omia ihanteitani tai pyrkimyksiäni ellen muutu"-tyyppisenä ajatuksena [Sch99].

Samalla kun henkilö hyväksyy muutoksen, hän tuntee sen mukanaan tuoman pakotteen luopua joistakin vanhoista ajattelutavoista ja tottumuksista, sekä ymmärtää uuden tiedon tarpeellisuuden ja sen oppimisen välttämättömyyden. Eloonjäämisaahdistuksen tai syyllisyyden lisäksi hän alkaa kokea oppimisaahdistusta. Oppimisaahdistuksen Schein määrittelee koostuvan yhdistelmästä eri pelkoja, jotka ovat väliaikaisen osaamattomuuden pelko, osaamattomuudesta rankaisemisen pelko, henkilökohtaisen identiteetin menettämisen pelko ja ryhmän jäsenyyden menettämisen pelko [Sch99]. Erityisesti ryhmän jäsenyyden menettämisen pelko voi olla hyvinkin vaikea voitettava, koska kehittäessään uusia ajattelutapoja tiimin jäsen poikkeaa muusta ryhmästä ja hänet saatetaan torjua tai jopa sulkea tiimin ulkopuolelle. Näin koko tiimin tulee muuttaa toimintamallejaan samanaikaisesti, jottei yksilö poissulkemisen pelon takia vastustaisi muutosta. Sama muutosvastaisuus koskee kaikkia oppimisaahdistuksen muodostavia pelkoja. Oppimisaahdistuksen pysyessä korkealla, yksilö pyrkii vastustamaan muutosta klassisen selviytymissyklin kolmen ensimmäisen tason mukaan [Kub69, Sch99]. Aluksi yksilö pyrkii torjumaan muutoksen vakuuttamalla itselleen, että kyseenalaistavat tiedot ovat paikkansapitämättömiä tai vain tilapäisiä. Seuraavalla tasolla henkilö syyttää muutoksesta muita ja pyrkii välttelemään vastuuta. Kolmannella tasolla yksilö pyrkii tekemään kauppaa muutoksella kysymällä mitä erikoisetuja hän saavuttaa, jos lähtee muutokseen mukaan. Jotta välttyttäisiin tämänlaiselta muutosvastarinnalta, tulee oppimisaahdistus minimoida erityisesti eloonjäämisaahdistukseen nähden, sillä oppimisaahdistuksen ollessa korkealla yksilö pyrkii vastustamaan muutosta, vaikka se tarkoittaisi eloonjäämisaahdistuksen ja syyllisyyden tuntemista. Transformatiivisessa muutoksessa onkin olennaista että eloonjäämisaahdistus tai syyllisyys ylittää oppimisaahdistukseen. Kuitenkin eloonjäämisaahdistuksen tai syyllisyyden kasvattaminen ei ole suotavaa, vaan on pyrittävä oppimisaahdistuksen vähentämiseen

Oppimisaahdistusta voidaan laskea kasvattamalla henkilön psykologista turvallisuuden tunnetta. Schein [Sch99] jakaa psykologisen turvallisuudentunteen kasvattamisprosessin kronologisesti kahdeksaan vaiheeseen:

1. Luodaan vastustamaton positiivinen visio, jonka tehtävänä on vaikuttaa työyhteisö muutoksen tarpeellisuudesta.

2. Järjesteään kaikki tapreellinen koulutus oli sitten kyseessä uusien tekniikoiden, arvojen, kuin toimintatapojen oppiminen. Esimerkiksi ketteriin metodeihin siirryttäessä tulee järjestää tiimidynamiikan sekä sitä tukevien ajattelumallien ja -tapojen mukaista koulutusta.
3. Koska jokaisella on omat oppimistekniikkansa, on oppilaille annettava mahdollisuus vaikuttaa myös oman oppimisprosessinsa suunnitteluun.
4. Annetaan ryhmätasolla epävirallista koulutusta, jotta koko ryhmä voi yhteisesti omaksua uudet normit ja oletukset. Näin pyritään vähentämään yksilön oppimisahdistuspelkoa ryhmästä syrjäytymisestä.
5. Pidetään ohjattuja harjoitustilanteita ja jaetaan palautetta. Ei ole mahdollista oppia jotain täysin uutta ellei anneta resursseja, aikaa ja palautetta omasta toiminnasta. Yksilöllä tulee olla mahdollisuus kokeilla uutta oppimaansa niin, että hänelle annetaan mahdollisuus oppia virheistään ilman välittömiä rangaistuksia.
6. On tarjottava positiivisia roolimalleja, joiden kautta oppijat voivat nähdä muutoksen käytännössä. Etenkin jos muutos on suuri ja ajattelutavaltaan täysin uusi, roolimallien merkitys korostuu. Oppijat voivat näin samaistua näihin edelläkävijöihin.
7. Perustetaan tukiryhmiä, joissa voidaan keskustella uuden toimintatavan ongelmista ja haasteista. Näin lisätään yhteenkuuluvuuden tunnetta ja helpotetaan ongelmien ylitsepääsemistä yhteisin voimin.
8. Organisaation tulee tarjota yhdenmukaiset järjestelmät ja rakenteet, jotka tukevat uutta työskentelytapaa. Palkitsemisjärjestelmät-kappaleessa tätä sivuutettiin kuvaamalla tiimityöskentelyn ja yksilökeskeisen palkitsemiskohdennuksen välistä ristiriitaa.

Luettelosta voidaan nähdä transformatiivisen muutoksen huomattavat aika- ja resurssivaatimukset, joihin varautuminen on ensiarvoisen tärkeää muutoksen onnistumiselle. Transformatiivisessa muutoksessa kannustetaan hienovaraisesti työntekijöitä muuttamaan ja oppimaan uusia asioita antamalla heille aikaa omaksumiselle, kokeilumahdollisuuksille, yhdessäkehittymiselle ja palautteenjaolle. Tästä voidaan nähdä melko suora yhtenevyys uuden tiedon syntysykliin, joka perustui sosiaalistamisen, ulkoistamisen, yhdistämisen ja sisäistämisen nelijakoon [NoT95].

6.2.2 Uusien käsitteiden ja vanhojen käsitteiden uusien merkitysten oppiminen

Psykologisen turvallisuuden lisäämisprosessin kahdeksan vaihetta voidaan tiivistää kahteen muutosmekanismiin, jotka ovat roolimallien jäljittely tai kokeilemalla oppiminen.

Yritys voi tuoda esiin henkilöitä, joiden toimintatapa jo tukee uutta haluttua työskentelytapaa tai jotka ovat ensimmäisinä omaksuneet sen. Muita voidaan sitten kannustaa ottamaan selvää näiltä henkilöiltä kuinka he ovat päämäärän saavuttaneet. Roolimallit sopivat useimmiten muutoksiin, joissa opetettavat käsitteet ja tavoitteet ovat selkeitä. Monimutkaisemmissa ja usein myös yksinkertaisemmissa muutoksissa voi käydä niin, että roolimallin esimerkki poikkeaa liiaksi oppijan ajattelutavoista tai persoonallisuudesta. Tällaisissa tapauksissa yksilö palaa helposti vanhoihin toimintatapoihinsa roolimallin poistuessa. Jos siis halutaan työntekijöiden oppivan asiat heidän persoonallisuutta tukevalla tavalla, tulee roolimallien käyttöä välttää ja antaa mahdollisuus kokeilemalla oppimiseen. Tällöin yritys tarjoaa vain tuen ja puitteet muutokselle antaen työntekijöiden löytää oma tapansa oppia uudet työskentelytavat. Erityisesti on muistettava, ettei henkilöllä varsinaisesti ole valinnanvapautta vaikuttaa itse päämäärään, vaan vain keinoihin kuinka hän päämäärän saavuttaa.

6.2.3 Uusien käsitteiden ja merkitysten sisäistäminen

Viimeisenä vaiheena transformatiivisessa muutoksessa on saavutetun tilanteen vakiinnuttaminen sisäistämällä opitut asiat. Henkilön yhdistäessä opitut asiat identiteettinsä ja käsitykseen omasta itsestään on muutos saavuttanut yrityskulttuuri-määritelmän alimman tason eli pohjimmaiset olettamukset ja arvot. Tämä kuitenkin edellyttää, että uudet ajattelumallit sopivat työntekijän omaan persoonaan ja ovat yhdenmukaisia hänen työ- ja sosiaalisessa ympäristössä olevien muiden tärkeiden ihmisten kanssa. On kuitenkin huomattava, että mikäli työntekijä oppii uusia asioita, jotka eivät ole yhteensopivia nykyisen työympäristön kanssa hän joko palaa vanhoihin toimintatapoihin tai jättää ryhmän. Ryhmän jättäminen voi tulla kyseeseen myös tilanteessa, jossa muu ryhmä on omaksunut uudet toimintatavat, mutta henkilö ei ole halukas

seuraamaan niitä. Käytännössä tämä voi tarkoittaa siirtymistä uusin työtehtäviin talon sisällä tai sitten työpaikan vaihtoa. Yrityksen kannalta tämä tarkoittaa osaamispääoman menetystä, mutta myös samalla muutosta hidastavien voimien vähenemistä.

6.3 Yrityskulttuurimuutos keski-ikäisessä yrityksessä

Keski-ikäiseksi yritykseksi Schein laskee yritykset, joiden johtamisprosessit ovat tehtävänsä nimitettyjen yleisjohtajien eivätkä yrittäjien, perustajien tai perustajaperheiden luomia kuten nuorissa yrityksissä. Keski-ikäisessä yrityksessä kulttuureja voi olla monia, erityisesti jos yritysostoja tai fuusioita on sen historiassa useita. Käsitteet toimintatapojen oikeellisuudesta ovat itsestäänselviä ja syvälle juurtuneita sekä yrityskulttuuri kokonaisuutena vahva. Tämä tarkoittaa myös voimakasta vastareaktiota kulttuurimuutokseen lähdetessä.

Yrityksen kasvaessa maantieteellisesti syntyy helposti voimakkaita alakulttuureja, joiden muuttaminen voi olla työlästä. Aina ei välttämättä edes ole järkevää lähteä yhdistämään näitä kaikkia alakulttuureja yhdeksi, koska tällöin on vaarana luoda joihinkin tilanteisiin täysin sopimattomia oletuksia. Usein voikin olla järkevää pyrkiä löytämään oikeat tavat vahvistaa jo olemassaolevien alakulttuurien moninaisia voimia samalla kun muutetaan kulttuurien toimimattomia elementtejä. Jos kulttuurillisia elementtejä tarvitsee muuttaa, on kyseessä edellisessä kappaleessa esitelty transformatiivinen muutos. Kuten transformatiivisessa muutoksessa todettiin, on se hidas, muutosvastarintaa herättävä ja resursseja vaativa prosessi. Tätä keski-ikäisen yrityksen hidasta kulttuurimuutosta Schein kutsuu suunitelluksi ja johdetuksi kulttuurimuutokseksi.

Suunitellulle ja johdetulle kulttuurimuutokselle on tunnusomaista pyrkimys yhdistää muutokset jo vallalla oleviin kulttuurisiin oletuksiin. Välttämättömien muutosten yhdistäminen vallalla oleviin kulttuurisiin teemoihin on myös paljon helpompaa, kuin lähteä varsinaisesti julistamaan uutta kulttuuria. Tehokkaat organisaatiot pystyvät kehittämään menettelytapojaan tällä tavoin muutamien tärkeiden ydinarvojen- ja oletusten ympärille, jotka eivät muutu [KoH92, CoP94] Tällöin kulttuuri jo itsessään tukee ja kannustaa uuden oppimiseen [NoT95]. Oli

sitten kyseessä täydellinen tai täydentävä kulttuurimuutos, on suunniteltu ja johdettu kulttuurimuutos yleensä aina hyvin pitkä prosessi. Jos organisaatiota ei hajoteta tai jälleenrakenneta, muutos vie Scheinin kokemuksen mukaan noin 5-15 vuotta.

Mikäli organisaation on toteutettava kulttuurimuutos taloudellisista syistä hyvin nopealla aikataululla, on yrityksellä vaihtoehtoina joko muuttaa kulttuurin osia nopeasti tai organisaation täydellinen tuhoaminen uudelleenorganisoimisprosessin kautta fuusioitumalla, yritysostolla tai konkurssimenettelyllä. Kummassakin tapauksessa on muutos henkilöstömielessä tuskallinen, koska useimmiten on helpointa ja nopeinta päästä vanhoista kulttuurillisista oletuksista eroon irtisanomalla henkilöt, jotka vahvimpina sitä ylläpitävät. Kuten hyvinvointikappaleessa jo todettiin, pelko oman työpaikan jatkuvuudesta vaikuttaa helposti negatiivisesti yksilön hyvinvointiin ja sitä kautta motivaatioon. Tämän takia voimakkaan kulttuurimuutoksen toteutuksessa on erittäin tärkeää suunnata voimavaroja muutoksen jälkeiseen henkilöstötoimintaan, jotta ilmapiiri ja motivaatio saadaan säilytettyä kilpailukykyisinä. Aina tämä ei ole mahdollista ja yrityksen onkin hyvä tarkkaan punnita todellisia resurssejaan suorittaa voimakas kulttuurimuutos, koska lopputuloksena voi olla pahasti velkaantunut yritys, jonka tärkeimmät osaajat ovat sen jättäneet. Tällöin ainoaksi vaihtoehdoksi jää edellä mainittu uudelleenorganisoimisprosessin mukainen toiminta.

6.4 Kulttuurimuutosesimerkki: siirtyminen elinkaarimallista ketteriin metodeihin

Kuten Schein [Sch99] totesi, kaikki lähtee pohjimmaisista perusoletuksista. Hieman vastaavasti myös Beck [BeA04] viittasi epäsuorasti näihin perusoletuksiin XP:n arvomäärittelyillä. Jos yrityksen arvot poikkeavat merkittävästi XP:n arvoista, tuo muutos todennäköisesti vain lisää ongelmaa [BeA04]. Vain vaikuttamalla näihin perusoletuksiin, saadaan muutoksesta pysyvä. Esimerkiksi ilmoittamalla vain uudet toimintatavat ja määrittelemällä kirjalliset sitä tukevat arvot, yritys voi saada aikaan näennäisen muutoksen, mutta joka lopulta kuitenkin toimii joko tehottomasti tai palautuu pian vanhaan toimintamalliin [Sch99]. Tässä kappaleessa käymme läpi kuvitteellisesti Macrosoftin kohtaaman

yrityskulttuuriprosessin sen siirtyessä byrokraattisesta elinkaarimallistaan kohti ketteriä liiketoimintamalleja.

6.4.1 Nykytilanteen kartoittaminen

Microsoftin asema internetpankki-sektorilla on heikentynyt entisestään. Kilpaileva yritys Fastcorp on kehittänyt uusia kilpailevia tuotteita ilman Microsoftia vaivaavaa menneisyyden painolastia. Toisaalta tämä vaikeuttaa asiakkaiden siirtymistä uuteen järjestelmään, mutta kilpailevan yrityksen nopeampi reagointikyky palautteeseen ja muuttuviin markkinatilanteisiin sekä tuotteen edullisemmat kustannukset ovat painamassa vaakakuppiä Fastcorpin puolelle. Microsoft ymmärtää tilanteen vakavuuden ja aloittaa laajamittaiset kartoitukset kilpailijan ja oman liiketoiminnan välisten erojen selvittämiseksi.

Järkevästi Microsoftin johto on sisäistänyt ajatuksen ulkoisten artefaktien pinnallisuudesta ja harhaanjohtavuudesta. Tämän takia Microsoft pyrkii tutkimaan kaiken mahdollisen Fastcorpin toiminnasta, mutta ymmärtää, ettei voi näiden perusteella päätellä kilpailijansa varsinaisia voimavaroja ja vahvuuksia. Tiukan kilpailutilanteen takia Microsoftilla ei ole paljon vuosikertomuksia ja satunnaisia epävirallisia tarinoita kummempaa informaatiota saatavilla kilpailijastaan. Nopeasti Microsoftilla ymmärretään keskittyä vain oman liiketoiminnan tutkimiseen ja kehittämiseen, sillä kilpailijasta saatu vaikutelma voi olla harhaanjohtava ja yrityskulttuurien yksilöllisyydestä johtuen kilpailevan kulttuurin kopiointi mahdotonta omaan toimintaympäristöön. Toki summittainen arvio kilpailijan vahvimmista alueista ja toimintatavoista voi olla lähtökohtana muutokselle, mutta näkökulma on pidettävä oman yrityksen heikkouksissa ja vahvuuksissa. Selvää kuitenkin on, että Fastcorpin toiminta poikkeaa merkittävästi Microsoftin tiukasta prosessikeskeisestä ja kontrollia suosivasta toimintamallista. Sellaiset termit, kuin XP, ketterät menet, iteratiivinen toimintamalli ja asiakaslähtöisyys ovat tulleet useassa yhteydessä vastaan Fastcorppia tutkittaessa. Enemmän ketterää toimintamallia tutkittuaan Microsoftin johtajat ymmärtävät edessä olevan muutoksen valtavuuden. Toisaalta he samalla myös ymmärtävät muutoksen välttämättömyyden yrityksen mahdollisuuksille selvitä kiristyvästä kilpailusta.

Macrosoft päättää palkata ulkopuolisia konsultteja selvittämään yrityksen nykytilanteen. Tähän ratkaisuun päädyttiin kahdesta syystä: ulkopuoliset henkilöt pystyvät objektiivisesti tarkastelemaan yrityksen toimintaa ja tarjomaan uusia ratkaisuja ilman Macrosoftissa opittujen pohjimmaisten arvojen ja käsitysten vääristävää vaikutusta. Kulttuuria ei voida arvioida valmiilla kyselyillä tai tutkimuksilla, koska ei tiedetä mitä kysyä, eikä vastausten todenperäisyyttä voida arvioida [Sch99]. Parhain tapa lähteä kartoittamaan yrityksen nykytilannetta artefaktien, ilmaistujen arvojen ja pohjimmaisten olettamusten suhteen, onkin pitää yhteisiä ryhmäkeskusteluja, joissa työntekijät saavat vapaasti ilmaista mielipiteensä ilman pelkoa jälkiseuraamuksista. Keskustelujen avulla saadaan yleiskuva yrityksen sisällä vaikuttavista näkymättömistä olettamuksista ja voimista. Näillä löydetyillä olettamuksilla voidaan myös selittää useita ristiriitaisuuksia näkyvien artefaktien ja ilmaistujen arvojen välillä. Otteita Macrosoftin ryhmäkeskusteluista on lueteltu allaolevassa taulukossa (taulukko 2, kts. myös liite 2).

Mielipide	Ongelma	Kehitysehdotus
Ohjaavia järjestelmiä on aivan liikaa. On turha toimia nopeasti, koska prosessin useat välivaiheet, portaat ja kokoukset pakottavat tiettyyn etenemisvauhtiin. Tämän takia teenkin töitä vain sen verran kuin on pakko, liika yrittäminen vain turhauttaa.	Valvonta ja kontrolli laskevat työntekijöiden tehokkuutta ja heikentää heidän motivaatiotaan	Ketteryysmanifestin mukaisesti korostetaan yksilöitä ja yhteistyötä. Luotetaan henkilöihin työkalujen sijaan.
Johtajat ovat suurten yksiköiden ja paperihommien takia etäisiä ja vaikeasti lähestyttäviä. Tämä pätee myös projektipäälliköihin, jotka tekevät pitkiä päiviä saadakseen kaikki suunnittelu- ja laskentatyönsä tehtyä. Tämän takia ilmenee usein johto vastaan työntekijät -tyyppistä ajattelua.	Johtajien aika kuluu lähes kokonaan prosessin ylläpitämiseen ja valvontaan. Heillä ei ole aikaa toimia johtajuusmaisesti. Samalla syntyy formaalin kommunikoinnin avustamana automaattisesti vastakkainasettelu "käskevien johtajien" ja "orjallisesti toimivien" työntekijöiden välillä.	Pyritään keventämään prosessia niin, että johtajille jää aikaa myös johtajuustyyppiseen toimintaan pelkkien asioiden johtamisen sijaan. Johtajille järjestetään koulutusta transformatiivisen johtajuuden periaatteista.
Tuntiseurannat ja muutoksenhallintavälineet ovat iso vitsi. Jos et täytä tiettyä viikkotuntimäärää, sinun katsotaan laiskottelevan riippumatta siitä kuinka tehokkaasti työtä teet. Muutostenhallintajärjestelmässä katsotaan vain muutosten läpimeno- tai korjausaikaa riippumatta muutoksen luonteesta.	Käytettävien valvontatyökalujen tuloksia ja raportteja ei osata tulkita oikein, eivätkä ne kuvaa todellista toimintaa. Tunnuslukujen ylläpitäminen kasvattaa kustannuksia hukkaan menneiden työtuntien muodossa.	Jos ei löydetä luotettavia tunnuslukuja, luovutaan niistä, koska muuten ohjataan toimintaa kohti valheellisia päämääriä. Otetaan työntekijät mukaan tunnuslukujen määrittelyyn.
Omasta mielestäni olen erittäin asiantunteva työssäni. Tuskin kukaan talossa pystyisi tekemään töitäni yhtä pätevästi.	Asiantuntemus on keskittynyt tiettyihin avainosaajiin. Näin syntyy helposti suojatyöpaikkoja tiettyjen osaamisalueiden ympärille	Kannustetaan työtehtävien kierrättämiseen vähentämällä irtisanomisen ja erehtymisestä leimautumisen pelkoa.
Työpaikan tukema ruokailu, liikuntamahdollisuudet ja työpaikan ulkopuoliset sosiaaliset tapaamiset ovat erittäin hyviä lisiä. Palkitseminen on kuitenkin liian johtajavoittoista, ihan kuin he tekisivät kaiken työn. Muutenkin palkitseminen kannustaa vain yksilösuorituksiin.	Työntekijät ymmärtävät heille myönnettyjen etujen arvon, mutta palkitsemisen kohdennus on epäreilu.	Jatketaan hyvinvointihierarkian mukaista tukemista ja määritellään palkitsemisjärjestelmät tukemaan osaamisen kehittämistä ja tiimitoimintaa. Sisäistetään aineettomien ja aineellisten palkitsemistapojen ero.
Mitään ei voi tehdä ilman valtavaa määrää paperitöitä, lupahakemuksia ja allekirjoituksia.	Tietämyksenhallinta perustuu vain ulkoisiin artefakteihin, mikä hidastaa oppimista. Sisäistä tietämystä ei hyväksytä, koska se ei sovi tiukan kontrolloituun prosessiajatteluun.	Järjestetään koulutusta tietämyksenhallinnasta ja uuden tiedon syntymekanismeista. Hyväksytään sisäinen tietämys ja osaamispääoma yhtiön tärkeimmäksi omaisuudeksi.

Taulukko 2, Ote Macrosoftin lähtötilanteen kartoituksesta

6.4.2 Muutoksen toteutus

Yrityskulttuurin määritelmän mukaan muutoksen lähde on ensin luotava pakottava ja hyväksytty tarve eli visio, jonka suuntaan yrityksen toimintaa ja henkilökunnan ajatusmaailmaa hienovaraisesti ohjataan. Macrosoftin tapauksessa muutostarvetta voidaan perustella taloudellisilla pakotteilla ja sitä kautta toiminnan jatkumisen edellytyksenä. Nykytilanteen kartoituksessa myös huomattiin, että työntekijätasolla olisi kysyntää radikaalimmallekin muutokselle. Transformaatiota varten yritys palkkaa uusia muutosjohtajia ja kouluttaa jo olemassaolevaa johtohenkilökuntaansa toimimaan perinteisen käskytyksen ja kontrolloivan johtamistavan sijaan transformatiivisen johtajuuden periaatteiden mukaisesti. Ulkoisten muutosjohtajien rooli voi olla hyvinkin tärkeä, sillä yrityksen sisäiset kontrolliin ja valvontaan kasvaneet johtajat todennäköisesti kohtaavat isoja haasteita luopua vallastaan tai omaksuessaan kevyempiä ja pehmeämpiä tapoja vaikuttaa alaisiinsa [Gri05]. Muutosjohtajien tarkoituksena on kannustaa muita työntekijöitä kohti uusia toiminta- ja ajattelutapoja sekä toimia roolimalleina joihin samaistua. Koska Macrosoft on vielä vakavarainen yritys, ei muutosta tehdä orgaanisessa mielessä kivuliaalla eli voimakkaalla kulttuurimuutoksella, vaan pyritään toimimaan transformatiivisesti oppimisahdistusta vähentämällä eloonjäämisahdistukseen nähden.

Koska lähes aina suurin epäselvyys liittyy lähtötilanteen kartoittamiseen ja sen pohjalta uusien päämäärien asettamiseen [Sch99], voidaan muutokseen nyt keskittyä transformatiivisen muutoksen kahdeksan psykologista turvallisuutta lisäävän periaatteen mukaan. Selkeästi koulutustarpeita on Macrosoftin tapauksessa monia, kuten tiimidynamiikka, transformatiivinen johtajuus, yrityskulttuurimuutos, palkitsemisjärjestelmät ja jatkuvan muutoksen tuoman epävarmuuden hallinta. Macrosoftin idealistisena päämääränä voidaan pitää oppivan organisaation mallia, sillä sen pääperiaatteet tukevat uutta haluttua toimintatapaa. Transformatiivisen muutoksen mukaisesti on tähän perustavanlaatuisen muutokseen varattava useita vuosia, annettava mahdollisuus kokeilla uusia työmenetelmiä, varattava riittävästi resursseja, tuettava ja ohjattava muutosta hienovaraisesti oikeaan suuntaan sekä ennen kaikkea annettava työntekijöille mahdollisuus vaikuttaa oppimisprosessiin, jotta

he motivoituisivat ja sitoutusivat uuteen toimintamalliin. On kuitenkin ymmärrettävä keski-ikäisen organisaation yrityskulttuurin syvyys ja voimakkuus, eikä lähteä suoraan muuttamaan kaikkea mitä se edustaa. Scheinin [Sch99] ehdotusten mukaan on pyrittävä erottelemaan vanhan kulttuurin vahvuudet ja heikkoudet, minkä jälkeen on kehitettävä vahvuuksia ja transformatiivisesti muutettava heikkouksia kohti toivottua päämäärää. Lopputuloksena on vanhan ja uuden kulttuurin yhdistelmä, joka vastaa paremmin yrityksen nykyistä toimintaympäristöä. Macrosoftin tapauksessa voidaan olettaa, ettei kaikesta kontrollista ja valvonnasta koskaan päästä eroon ilman mittavia henkilöstöirtisanomisia. Tässä mielessä Scrum voisi olla parempi tavoite ketterälle toiminnalle kuin täysin uutta ajattelumaailmaa edustava XP. Mikään ei toki estä Macrosoftia sekoittamasta näitä kahta keskenään tai muiden ketterien metodien esittelemiä toiminta- ja ajatusmalleja. Tärkeintä on, että valittu malli tukee yrityksen kulttuuria ja saa siten henkilöstön hyväksynnän.

Huolimatta valitusta ketterästä metodista, suurimman muutoksen Macrosoft kohtaa henkilöstön hallinnassa. Tehokkuutta haetaan ketteryyshänifestin mukaisesti asiakaskeskeisyydellä, formaalien toimintamallien ja dokumenttien vähentämisellä sekä yhteistyötä ja muutosta korostamalla. Tämä johtaa väistämättä näkymättömän osaamispääoman ja sisäisen tietämyksen lisääntyvään kasvuun, minkä oikeaoppiseen ohjaamiseen tarvitaan johtajuustyyppistä hallintaa. Samalla myös hyvinvoinnin merkitys kasvaa, sillä tehokkuutta ei enää pystytä lisäämään valvontaa ja kontrollia tiukentamalla, vaan kasvattamalla työntekijöiden sitoutuneisuutta ja motivaatiota. Oikeaoppinen palkitseminen on yksi välillinen tapa lisätä hyvinvointia, kuten myös muut työympäristö- ja hyvinvointi-kappaleissa esitetyt keinot.

Organisaatorakenne kokee todennäköisesti rajun muutoksen prosessia ylläpitävien työpaikkojen menetyksen johdosta. Näille henkilöille voidaan tarjota vaihtoehtoisia töitä, mutta täysin ilman henkilöstöirtisanomia tuskin selvitään. Samaan aikaan tulisi ylläpitää mielikuvaa työpaikkojen jatkuvuudesta, minkä takia tilanne on haastellinen. Täysin oikeita ratkaisuja ei ole, kuten ei yrityskulttuurissa muutenkaan, mutta avoin, informatiivinen ja oikeudenmukainen toiminta lienevät turvallisia tapoja lähteä toteuttamaan tätä kivuliasta irtisanomisprosessia.

Kuten käytännön toimintaympäristö- ja transformatiivinen muutos-kappaleissa todettiin, on tärkeää tarjota yhtenäiset ja uutta toimintatapaa tukevat työskentelyolot sekä -välineet. Macrosoftin tapauksessa tämä tarkoittaa samalla useista, kalliisti kehitetyistä valvonta- ja kontrollointijärjestelmistä luopumista. Asiakkaiden mahdolliselle läsnäololle on varattava yhteistoimintaa tukevat tilat sekä tarjottava teknisesti mahdollisuus seurata ja vaikuttaa projektin etenemiseen myös etänä.

6.4.3 Muutoksen ylläpito ja saavutetun aseman säilyttäminen

Kuten transformatiivinen muutos-kappaleessa todettiin, riippuu muutoksen pysyvyys yleensä siitä onko sen merkittävin aikaansaaja ollut roolimalli vai henkilön oman kokeilemisen tulos. Jos muutos perustuu pelkästään roolimalleihin, on vaarana paluu vanhoihin toimintatapoihin roolimallin poistuessa. Osittain tämän takia varsinainen muutos on hyvä suorittaa riittävän pitkällä aikajänteellä, jotta kaikki työntekijät saavat mahdollisuuden löytää omalle persoonallisuudellensa sopivat tavat saavuttaa haluttu päämäärä. Mikäli uudet ajattelu- ja toimintamallit ovat ristiriidassa työntekijän persoonallisuuden ja sisäisten arvokäsitysten kanssa, hän todennäköisesti jättää nykyisen työpaikkansa ja siirtyy uusiin tehtäviin, joko talon sisällä tai sen ulkopuolella.

Macrosoftin siirtyessä hitaasti byrokraattisesta prosessikeskeisestä toimintamallistaan kohti ketterää liiketoimintamallia ja oppivan organisaation periaatteita, se luo samalla myös muutosmyönteisen ja itseäänkehittävän yrityskulttuurin. Näin tulevat muutokset eivät välttämättä ole niin haastellisia ja aikaavieviä, koska yrityskulttuuri jo itsessään tukee kehitystä ja nopeaa reagointia muuttuneisiin toimintaympäristövaatimuksiin. Tehokkaat organisaatiot pystyvät kehittämään menettelytapojaan tällä tavoin muutamien tärkeiden ydinarvojen- ja oletusten ympärille, jotka eivät muutu [KoH92, CoP94, NoT95].

7 Yhteenveto

8 Lähteet

- [AMK06] Artto K., Martinsuo M., Kujala J., *Projektiliiketoiminta*. WSOY Oppimateriaalit Oy, 2006.
- [Bak72] Baker F.,T., *Chief programmer team management of production programming*. IBM System Journal, 11,1(1972). [Myös: <http://www.research.ibm.com/journal/sj/111/ibmsj1101E.pdf>, 26.1.2009].
- [BeA04] Beck, K., Andres, C., *Extreme Programming Explained (2.painos)*, Addison-Wesley, 2004.
- [Bec01] Kent Beck, et.al., *Manifesto for Agile Software Development*. <http://agilemanifesto.org/> [16.3.2009]. [Myös liite 1].
- [Ber06] Joseph Bergin, *Patterns for agile development practice part 3 (version 4)*. PLoP '06: Proceedings of the 2006 conference on Pattern languages of programs, lokakuu 2006.
- [Boe02] Boehm, B., *Get ready for agile methods, with care*. Computer, tammikuu 2002, sivut 64–69.
- [BuH06] Buchanan, D., Huczynski, A., *Organizational behaviour: an introductory text*. Prentice-Hall, 2006.
- [Bur78] Burns, J., *Leadership*. Harper Torchbooks, 1978. [Myös: http://www.spsfg.org/data/files/News/AcademicsNews/Leadership_by_Burns1.pdf, 19.2.2009].
- [ChM04] Chau T., Maurer F., *Tool Support for Inter-team Learning in Agile Software Organizations*. LNCS 2096, LSO 2004, sivut 98-109. [Myös: <http://books.google.fi/books?id=QeLbLZHJNbgC&pg=PA98&dq=iso+agile&ei=4pBvSaa8FoiUzAT655nvDg#PPA98,M1>, 24.1.2009].
- [Coc06] Cockburn, A., *Agile Software Development: The Cooperative game*. Addison-Wesley, 2006.
- [CoH01] Cockburn A., Highsmith, J., *Agile Software Development: The People Factor*. Computer, marraskuu 2001, sivut 131-133.
- [Con01] L. Constantine, *Methodological Agility*. Software Development, kesäkuu 2001, sivut 67-69. [Myös:

http://books.google.fi/books?id=Wte1EGtvddoC&pg=PA232&lpg=PA232&dq=constantine+Methodological+Agility&source=bl&ots=gOfnDuGg71&sig=mom24o8xn7zZ9IRkxtoxNJpN0gl&hl=fi&sa=X&oi=book_result&resnum=1&ct=result#PPA235,M1, 16.3.2009].

- [CoP94] Collins, J.C., Porras, J.I, *Built to Last*, HarperBusiness, New York, 1994
- [Gri05] Griffiths, M., *Teaching agile project management to the PMI*. Agile Conference, Proceedings, 24,29(2005), sivut 318 – 322.
- [HiC01] Highsmith, J., Cockburn, A., *Agile software development: the business of innovation*, Computer, 34,9(2001), sivut 120-127.
- [Hig04] Highsmith, J., *Agile Project Management*. The Agile Software Development Series, Addison-Wesley, 2004.
- [Hig03] Highsmith, J., *Agile Project Management: Principles and Tools*, Cutter Consortium Reports, 4,2(2003).
- [KrI08] Krasteva, I., Ilieva, S., *Adopting an agile methodology: why it did not work*, APOS '08: Proceedings of the 2008 international workshop on Scrutinizing agile practices or shoot-out at the agile corral, toukokuu 2008.
- [KoH92] Kotter, J.P, Heskett, J.L, *Corporate Culture and Performance*. Free Press, New York, 1992.
- [Kub69] Kubler-Ross, E., *On Death and Dying*. Macmillian, 1969. [Myös: http://books.google.com/books?hl=fi&lr=&id=9yfJQBo_9GQC&oi=fnd&pg=PP7&dq=on+death+and+dying&ots=r6viJnq1RX&sig=exo2D38TIL1PMYG5pxoTUUIfAzk#PPP5,M1, 4.3.2009].
- [Leh04] Lehtonen, J-M (toim.), *Tuotantotalous*. WSOY, 2004.
- [Leh80] Lehman, M.M., *On Understanding Laws, Evolution and Conservation in the Large-Program Life Cycle*. Journal of Systems and Software 1,3(1980), sivut 213-231.
- [MAL05] Mintzberg H., Ahlstrand, B., Lampel, J., *Strategy Safari A Guided Tour Through the Wilds of Strategic Management*. The Free Press, New York, 2005.

- [MaM06] Mauer, F., Melnik G., *Agile methods: moving towards the mainstream of the software industry*. ICSE '06: Proceedings of the 28th international conference on Software engineering, toukokuu 2006
- [Mas43] Maslow, A.H., *A Theory of Human Motivation*. Psychological Review, 50,4(1943), sivut 370-396.
- [MeM02] Metcalfe, B.-A., Metcalfe, J.-A., *The great and the good*. People Management, 8,11(2002).
- [MeM03] Metcalfe, B.-A., Metcalfe, J.-A., *Under the influence*. People Management, 9,5(2003).
- [Min77] Mintzberg, H., *The Manager's Job: Folklore and Fact*. Harvard Business Review, 55,4(1977), sivut 49-61. [Myös: <http://www.uu.edu/personal/bnance/318/mintz.html>, 20.2.2009].
- [Mor05] Morien, R., *Agile management and the Toyota way for software project management*. IEEE International Conference, Industrial Informatics (INDIN), elokuu 2005, sivut 516 – 522.
- [NMM05] Nerur, S.P, Mahapatra R., Mangalaraj, G., *Challenges of migrating to agile methodologies*. Communications of the ACM , 48,5(2005).
- [NoT95] Nonaka, I., Takeuchi, H., *The Knowledge-Creating Company: How Japanese Companies Create The Dynamics*. Oxford University Press, 1995. [Myös: http://books.google.com/books?hl=fi&lr=&id=B-qxrPaU1-MC&oi=fnd&pg=PA3&dq=The+Knowledge+Creating+Company+&ots=XfTNNbVj9Z&sig=Oc9_YIiOoFF3dRtn3kvJF1ucQhU#PPA6,M1, 11.2.2009].
- [OtA05] Ojala L., Ahonen G., *Työhyvinvointi tuloksen tekijänä*. WSOY, 2005.
- [PBB97] Pedler, M., Burgoyne, J., Boydell, T., *The Learning Company: A Strategy for Sustainable Development (2.painos)*, McGraw-Hill, London, 1997.
- [Pre00] Pressman, R., *Software Engineering, A Practitioner's Approach European Adaptation*. McGraw-Hill, 2000.

- [Roy70] Royce, W.W., *Managing The Development of Large Software Systems*. Technical Papers of Western Electronic Show and Convention (WesCon), Los Angeles, USA, 25,28(1970). [Myös: <http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf>, 29.1.2009].
- [ScB02] Schwaber, K., Beedle M., *Agile Software Development with Scrum*. Prentice Hall, 2002
- [ScB04] Schaufeli, W., Bakker, A., *Job demands, job resources, and their relationship with burnout and engagement: a multi-sample study*. Journal of Organizational Behavior, 25(2004), sivut 293–315. [Myös: <http://www.fss.uu.nl/sop/Schaufeli/208.pdf>, 13.2.2009].
- [Sch99] Schein, E., *The Corporate Culture Survival Guide – Sense and nonsence about culture change*. San Fransisco Jossey-Bass, 1999.
- [Sen90] Senge, P., *The Fifth Discipline*, Doubleday Business, 1990.
- [SNM01] Sircar, S., Nerur, S.P., and Mahapatra, R. *Revolution or Evolution? A Comparison of Object-Oriented and Structured Systems Development Methods*. MIS Quarterly, 25,4(2001), sivut 457–471.
- [Syd03] Sydänmaalakka, P., *Intelligent Leadership And Leadership Competencies: Developing a leadership framework for intelligent organizations*, Teknillinen korkeakoulu, Tuotantotalouden laitos, 2003. [Myös: <http://lib.tkk.fi/Diss/2003/isbn9512263602/isbn9512263602.pdf>, 12.2.2009].
- [TaN86] Takeuchi, H., Nonaka, I., *The New New Product Development Game*. Harvard Business Review, tammi-helmikuu 1986, sivut 137-146. [Myös: <http://apln-richmond.pbwiki.com/f/New%20New%20Prod%20Devel%20Game.pdf>, 27.1.2009].
- [ThK96] Theorell, T, Karasek R., *Current Issues Relating to Physiological Job Strain and Cardiovascular Disease Research*. Journal of Occupational Health Psychology, 1,1(1996), sivut 9-26.

- [Uus02] Uusitalo, R., *Tulospalkkaus ja tuottavuus*. Valtion taloudellinen tutkimuskeskus, 2002,
http://www.vatt.fi/file/vatt_publication_pdf/k276.pdf. [13.2.2009]
- [VHH98] Vartiainen, M., Hakonen, N., Hulkko, K., *Ryhmien ja tiimien palkitseminen*. Metalliteollisuuden kustannus Oy, 1998.
- [WaB76] Wahba, A., Bridgewell, L., *Maslow reconsidered: A review of research on the need hierarchy theory*. *Organizational Behavior and Human Performance*, 15(1976) , sivut 212–240.

Liite 1. Ketteryysmanifesti

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

© 2001, the above authors
this declaration may be freely copied in any form,
but only in its entirety through this notice.

[Twelve Principles of Agile Software](#)

[Become a Signatory](#)

[View Signatories](#)

[About the Authors](#)

[About the Manifesto](#)

site design and artwork © 2001, Ward Cunningham

Liite 2. Microsoftin lähtötilanteen kartoitus

Mielipide	Ongelma	Kehitysehdotus
Ohjaavia järjestelmiä on aivan liikaa. On turha toimia nopeasti, koska prosessin useat välivaiheet, portaat ja kokoukset pakottavat tiettyyn etenemismuotoon. Tämän takia teenkin töitä vain sen verran kuin on pakko, liika yrittäminen vain turhauttaa.	Valvonta ja kontrolli laskee työntekijöiden tehokkuutta ja heikentää heidän motivaatiotaan	Ketteryysmanifestin mukaisesti korostetaan yksilöitä ja yhteistyötä. Luotetaan henkilöihin työkalujen sijaan.
Johtajat ovat suurten yksiköiden ja paperitöiden takia etäisiä ja vaikeasti lähestyttäviä. Tämä pätee myös projektipäälliköihin, jotka tekevät pitkiä päiviä saadakseen kaikki suunnittelu- ja laskentatyönsä tehtyä. Tämän takia ilmenee usein johto vastaan työntekijät -tyyppistä ajattelua.	Johtajien aika kuluu lähes kokonaan prosessin ylläpitämiseen ja valvontaan. Heillä ei ole aikaa toimia johtajuusmaisesti. Samalla syntyy formaalin kommunikoinnin avustamana automaattisesti vastakkainasettelu "käskevien johtajien" ja "orjallisesti toimivien" työntekijöiden välillä.	Pyritään keventämään prosessia niin, että johtajille jää aikaa myös johtajuus-tyyppiseen toimintaan pelkkien asioiden johtamisen sijaan. Johtajille järjestetään koulutusta transformatiivisen johtajuuden periaatteista.
Esimiehet eivät kuuntele alaisiaan tarpeeksi tai jos kuuntelevatkin, niin käytännön hyötyjä siitä ei tunnu olevan.	Esimiehet eivät anna työntekijöille mahdollisuutta osallistua ja sitoutua toimintaan.	Jaetaan vastuuta ja annetaan työntekijöille mahdollisuus vaikuttaa. Samalla heidän sitoutuneisuutensa ja motivaationsa kasvaa hyvinvointihierarkian mukaisesti.
Kehityskeskustelut ja tyytyväisyyskartoitukset eivät tunnu vaikuttavan mihinkään. Tuntuu kuin ne tehtäisiin vain näön vuoksi. Sen takia en oikeasti mieti mitä kyselyihin vastaan, kunhan vain täytän ne jos sitäkin.	Tyytyväisyyskartoitusten ja kehityskeskustelujen voimaan ei uskota, eikä niiden tuloksiin voi näin ollen luottaa.	Pidetään info- ja kehitystilaisuuksia löydettyjen epäkohtien korjaamiseksi. Myönnetään riittäviä määrärahoja muutosten toteuttamiselle.
Tuntiseurannat ja muutoksenhallintavälineet ovat iso vitsi. Jos et täytä tiettyä viikkotuntimäärää, sinun katsotaan laiskottelevan riippumatta siitä kuinka tehokkaasti työtä teet. Muutostenhallintajärjestelmässä katsotaan vain muutosten läpimeno- tai korjausaikaa riippumatta muutoksen luonteesta.	Käytettävien valvontatyökalujen tuloksia ja raportteja ei osata tulkita oikein, eivätkä ne kuvaa todellista toimintaa. Tunnuslukujen ylläpitäminen kasvattaa kustannuksia hukkaan menneiden työtuntien muodossa.	Jos ei löydetä luotettavia tunnuslukuja, luovutaan niistä. Otetaan työntekijät mukaan tunnuslukujen määrittelyyn.
Talon johto on keskittynyt valvomaan vain tiettyjä tunnuslukuja tehokkuuden mittareina. Isossa organisaatiossa tämä on usein ainoa mahdollinen tapa, mutta tunnuslukujen todenperäisyyttä tulisi valvoa tarkemmin. Nyt on liian helppoa toimia tunnusluvullisesti oikein, mutta käytännön työn kannalta väärin.	Määritellyt tunnusluvut eivät osu yksiin toiminnan kanssa ja niistä vedetyt johtopäätökset ovat virheellisiä. Työntekijät turhautuvat ja pyrkivät parantamaan vain tunnuslukuja.	Kts. edellinen ehdotus. Jos ei voida luotettavasti mitata jotain toimintaa, niin sen valvonnasta tulee luopua. Muuten ohjataan toimintaa kohti valheellista päämäärää ilman sanottavaa kustannustehokkuuden kasvua.

Häly ja taustameteli työpaikalla vaikeuttavat omaa työskentelyä. Avokonttorit ovat liian isoja.	Avokonttori-idea on toteutettu säästeliäästi ahtamalla liian paljon ihmisiä samaan tilaan.	Otetaan työtilaesimerkiksi Beckin[BeA04] määrittelemä optimaalinen tiimityöskentely-ympäristö.
Omasta mielestäni olen erittäin asiantunteva työssäni. Tuskin kukaan talossa pystyisi tekemään töitäni yhtä pätevästi.	Asiantuntemus on keskittynyt tiettyihin avainosaajiin. Näin syntyy helposti suojatyöpaikkoja tiettyjen osaamisalueiden ympärille	Kannustetaan työtehtävien kierrättämiseen vähentämällä irtisanomisen ja erehtymisestä leimautumisen pelkoa.
Työpaikan tukema ruokailu, liikuntamahdollisuudet ja työpaikan ulkopuoliset sosiaaliset tapaamiset ovat erittäin hyviä lisiä. Palkitseminen on kuitenkin liian johtajavoittoista, ihan kuin he tekisivät kaiken työn. Muutenkin palkitseminen kannustaa vain yksilösuorituksiin.	Työntekijät ymmärtävät heille myönnettyjen etujen arvon, mutta palkitsemisen kohdennus on epäreilu.	Jatketaan hyvinvointihierarkian mukaista tukemista ja määritellään palkitsemisjärjestelmät tukemaan osaamisen kehittämistä ja tiimitoimintaa.
Tieto on avain menestymiselle. Mitä enemmän tiedän, sen varmemmin saan pitää työpaikkani. Yrityksen työntekijät ovat asiantuntevia ja osaavia.	Uuden tiedon aktiivinen omaksuminen on hyvä asia, mutta jatkuva opiskelutarve ei saa perustua työpaikan menettämisen pelkoon.	Vahvistetaan tätä uuden tiedon oppimisasennetta tarjoamalla työntekijöille koulutus- ja muita kehittymismahdollisuuksia. Pyritään korvaamaan irtisanomisen pelko motivaatiolähteenä eri palkitsemiskeinoilla.
Miksi lähteä muuttamaan mitään? Hyvinhän me näitä tuotteita olemme tälläkin tavoin saaneet tehtyä. Onhan noita kilpailijoita ollut ja aina olemme niistä selvinneet. Ehkä tämä on vain väliaikaista...odotellaan?	Yrityskulttuurin perimmäinen muutosongelma: "Miksi muuttua, kun tähänkin asti on näin toimien selvitty?"	Transformatiivisen muutoksen periaatteiden mukaan, luodaan vastustamaton visio ja vähennetään oppimisahdistusta.
Kyllähän tämä vähän byrokraattista ja kankeaa on, mutta näin on aina toimittu. Millään pienellä muutoksella ei selvitäisi, koska kaikki liittyy kaikkeen.	Sama kuin edellinen ongelma, mutta havaittavissa tarve myös muuttua, koska myönnetään olemassaolevat ongelmat.	Tuetaan tätä tarvetta päästä irti byrokraattisesta valvontajärjestelmästä ja kannustetaan muutokseen. Tarjotaan roolimalleja ja tukijärjestelmiä oikean päämäärän saavuttamiseksi.
Täydellisyyteen pyrkivä prosessi leimaa virheidentekijät. Jo yksikin virhe voi leimata huolimatta useista onnistumisista ja tämä leima vaikuttaa menestymiseesi, eikä se helposti unohdu. Näin ei tee mieli kokeilla mitään uusia kehitysideoita tai innovaatioita.	Syylisistä yrityskulttuuri estää tehokkaasti uuden oppimisen ja sitä kautta kehittymisen.	Kehitetään tiimidynamiikkaa ja yhdessäoppimisen taitoja. Toimitaan ennakkoluulottomasti ryhmissä toisia tukien.

<p>Työ on usein rutiinia samojen asioiden hoitoa. Töiden kierrättämistä ei suosita ja henkilöt enemmänkin suojelevat omaa aluettaan kuin jakavat tietouttaan muille.</p>	<p>Kulttuuri tukee traditionaalista yksilökeskeistä suoritusta hidastaen näin oppimista uuden tiedon syntysyklin mukaisesti. Samalla ammatillinen hyvinvointi kärsii kehityksen ja haasteiden puuttuessa.</p>	<p>Korostetaan sosiaalisen kanssakäymisen merkitystä tiedon jakamisen lähteenä formaalien dokumenttien sijaan. Annetaan työntekijöille mahdollisuus kehittää ammatillista hyvinvointiaan ilman työpaikan menettämisen pelkoa.</p>
<p>Omia "seikkailuja" ei suvaita jos ne eivät välittömästi tuota jotain. Tämän takia on parempi vain odottaa johtajien sanovan mitä tehdään ja sitten toteuttaa ne mahdollisimman nopeasti.</p>	<p>Formaali prosessi ja kontrolli tukahduttavat innovoinnin ja itsensäkehittämisen tarpeet</p>	<p>Otetaan työntekijät sekä asiakkaat mukaan kehitysprosessiin ja annetaan näiden työryhmien punnita toteutettavia "tarinoita" ketteryysmanifestin mukaisesti yhteistyömäisesti sopimusneuvotteluiden sijaan</p>
<p>Mitään ei voi tehdä ilman valtavaa määrää paperitöitä, lupahakemuksia ja allekirjoituksia.</p>	<p>Tietämyksenhallinta perustuu vain ulkoisiin artefakteihin, mikä hidastaa oppimista. Sisäistä tietämystä ei hyväksytä, koska se ei sovi tiukan kontrolloituun prosessiajatteluun.</p>	<p>Järjestetään koulutusta tietämyksenhallinnasta ja uuden tiedon syntymekanismeista. Hyväksytään sisäinen tietämys ja osaamispääoma yhtiön tärkeimmäksi omaisuudeksi.</p>