

INHIMILLISET TEKIJÄT OHJELMISTOTUOTANNOSSA

Marika Toivanen
Pro gradu -tutkielma
Tietojenkäsittelytieteen ja
sovelletun matematiikan laitos
Informaatioteknologian ja
kauppatieteiden tiedekunta
Kuopion yliopisto
Kesäkuu 2002

KUOPION YLIOPISTO, informaatioteknologian ja kauppatieteiden tiedekunta
Tietojenkäsittelytieteen koulutusohjelma
Tietojenkäsittelytiede

TOIVANEN MARIKA: Inhimilliset tekijät ohjelmistotuotannossa

Pro gradu -tutkielma, 93 s., 1 liite (3 s.)

Pro gradu -tutkielman ohjaajat:

FT, professori (mvs), Anne Eerola

FT, professori, Martti Penttonen

Kesäkuu 2002

Avainsanat: ihmiset, inhimilliset tekijät, ohjelmistotuotanto, projektityö, tehokkuus, vuorovaikutustaidot

Ohjelmistotuotannon kiihtyminen on lisännyt kiirettä ohjelmistotaloissa. Ohjelmistojen suunnittelijat ja toteuttajat työskentelevät haastavissa tehtävissä. Haasteellinen ja kiireinen työ voi olla stressaavaa, mutta myös motivoivaa.

Tutkielmassa käydään läpi ohjelmistotuotannon keskeiset osa-alueet, kuten ohjelmiston elinkaari, vaihejakomallit, suunnittelu- ja ohjelmointiparadigman valinta sekä projektityöskentely, inhimillisten tekijöiden näkökulmasta. Huomioon otetaan ensisijaisesti ohjelmistotyön tekijät, joiden mahdolliset roolit selvittää myös. Tutkimuksen kohteena ovat sekä kognitiiviset että sosiaaliset taidot. Tutkielmassa huomioidaan myös asiakkaan, markkinoiden sekä tehokkuuden ja tuottavuuden tärkeys ohjelmistotuotantoprosessissa.

Tutkielmassa tuodaan esiin ihmisten erilaisten taitojen tarpeellisuus. Ohjelmistotalalla tarvitaan monenlaisia ihmisiä. Suurin osa heistä tarvitsee työssä selviytyäkseen teknisten taitojen lisäksi sosiaalisia taitoja, kuten viestintä- ja vuorovaikutustaitoja. Myös johtamistaitojen tarve on kasvanut organisaatioiden madaltuessa ja muotoutuessa tiimiorganisaatioiksi. Tutkielmassa tarkastellaan tällaisia inhimillisiä tekijöitä ohjelmistotuotannon näkökulmasta.

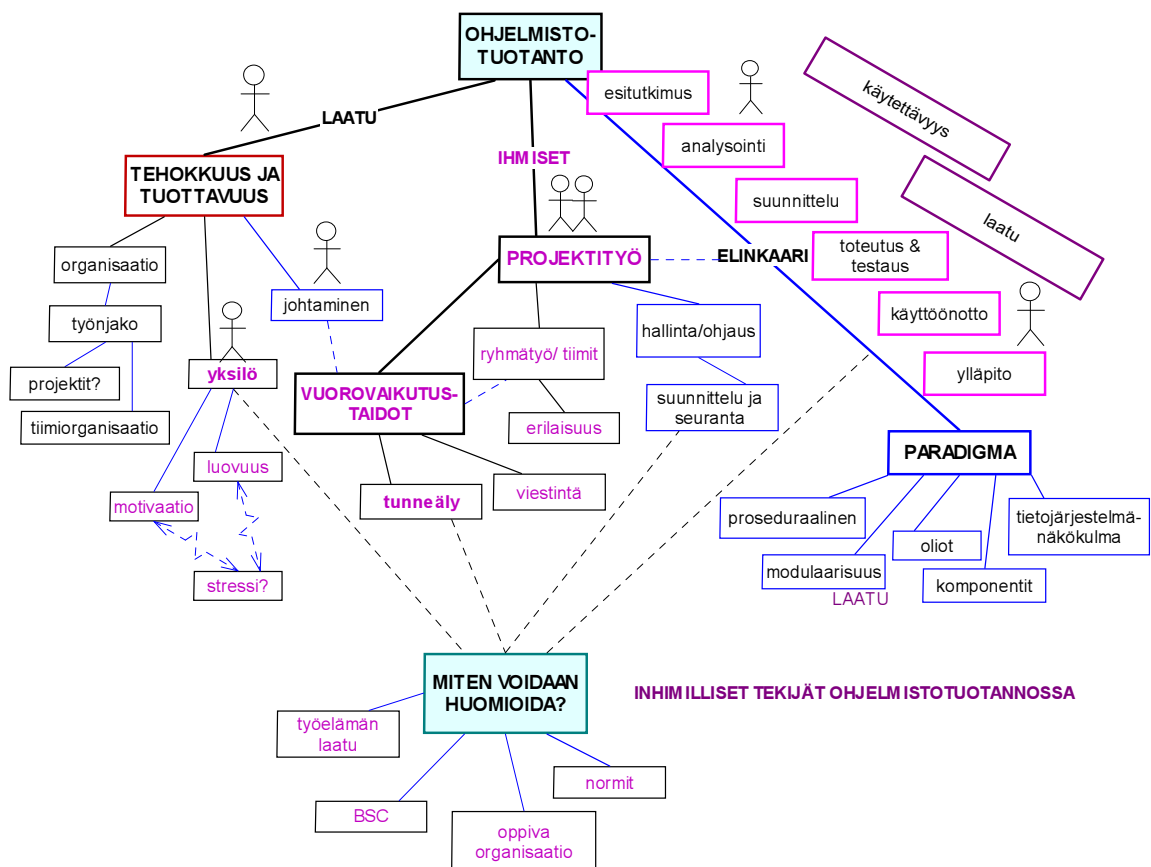
SISÄLLYS

1	JOHDANTO.....	4
2	OHJELMISTOTUOTANTO.....	6
2.1	ALAKOHTAISET ERITYISPIIRTEET	6
2.2	OHJELMISTON ELINKAARI	8
2.3	VAIHEJAKOMALLIT	9
2.4	LAATU	13
2.5	OHJELMISTOTUOTANTOPROSESSIN TOIMIJAT	14
2.6	IHMISET.....	17
2.6.1	<i>Sosiaaliset taidot.....</i>	<i>17</i>
2.6.2	<i>Kognitiiviset taidot.....</i>	<i>19</i>
3	SUUNNITTELU- JA OHJELMOINTIPARADIGMAN VALINTA.....	22
3.1	ENNEN OLIOKESKEISYYDEN LÄPIMURTOA	22
3.2	OLIOKESKEINEN SOVELLUSKEHITYS	23
3.2.1	<i>Periytyminen ja riippuvuudet</i>	<i>24</i>
3.2.2	<i>Käyttötapaukset.....</i>	<i>26</i>
3.3	MODULAARISUUS	27
3.4	KOMPONENTTIPOHJAINEN SOVELLUSKEHITYS	28
3.4.1	<i>Itsenäiset komponentit.....</i>	<i>28</i>
3.4.2	<i>Komponenttituotannon toimijat</i>	<i>29</i>
3.5	TIETOJÄRJESTELMÄNÄKÖKULMA	31
4	TEHOKKUUS JA TUOTTAVUUS.....	34
4.1	ORGANISAATIO	34
4.1.1	<i>Organisaatioarkkitehtuuri.....</i>	<i>35</i>
4.1.2	<i>Epäviralliset organisaatiot.....</i>	<i>37</i>
4.2	JOHTAJUUS.....	37
4.2.1	<i>Tilannejohtaminen</i>	<i>39</i>
4.2.2	<i>Erlaiset esimiehet.....</i>	<i>40</i>
4.2.3	<i>Avoin kommunikaatio.....</i>	<i>41</i>
4.2.4	<i>Kehityskeskustelut.....</i>	<i>42</i>
4.3	YKSIÖ	44
4.3.1	<i>Motivaatio</i>	<i>44</i>
4.3.2	<i>Työtyytyväisyys.....</i>	<i>47</i>
4.3.3	<i>Työaika, vapaa-aika ja työvire</i>	<i>47</i>
4.3.4	<i>Stressi.....</i>	<i>49</i>
4.3.5	<i>Luovuus.....</i>	<i>50</i>

5	PROJEKTITYÖSKENTELY	52
5.1	PROJEKTINHALLINTA	53
5.1.1	<i>Projektiorganisaatio</i>	53
5.1.2	<i>Suunnittelu ja käynnistäminen</i>	54
5.1.3	<i>Riskien hallinta</i>	55
5.1.4	<i>Toteutus ja päättäminen</i>	56
5.1.5	<i>Automatisointi</i>	57
5.2	TIIMIT	57
5.2.1	<i>Ryhmän kehitysvaiheet</i>	58
5.2.2	<i>Ryhmä vai tiimi?</i>	59
5.2.3	<i>Erilaisuuden hyödyntäminen</i>	59
5.2.4	<i>Erilaisuuteen ja muutoksiin suhtautuminen</i>	62
6	VUOROVAIKUTUSTAIDOT	64
6.1	VIESTINTÄ	65
6.1.1	<i>Sanaton viestintä</i>	67
6.1.2	<i>Manipulaatio</i>	67
6.1.3	<i>Palaute</i>	68
6.2	TYÖVÄLINEIDEN VAIKUTUS	70
6.3	TUNNEÄLY	71
6.3.1	<i>Tunteet</i>	72
6.3.2	<i>Tunteiden tunnistaminen ja ilmaiseminen</i>	74
6.3.3	<i>Tunteiden säätely</i>	75
6.3.4	<i>Konfliktit</i>	76
7	MITEN INHIMILLISET TEKIJÄT VOIDAAN OTTAA HUOMIOON?	77
7.1	YRITYSSTRATEGIA	77
7.1.1	<i>Työelämän laatu</i>	77
7.1.2	<i>Strategiamalli</i>	78
7.2	NORMIT	79
7.2.1	<i>Työmoraali</i>	80
7.2.2	<i>Kirjoitetut säännöt</i>	80
7.3	OPPIVA ORGANISAATIO	81
8	POHDINTA	84
	LÄHTEET	87
	LIITE: TIETOTEKNIIKAN AMMATTILAISEN EETTISET SÄÄNNÖT	94

1 JOHDANTO

Ohjelmistotuotannon tarkoitus on *tuottaa asiakkaille ohjelmistotuotteita*. Tuotteista saadaan rahaa, jolla yritykseen voidaan palkata työntekijöitä. *Tuotteiden laatu* vaikuttaa siihen, saako yritys lisää tuotteita kaupaksi. Tuotteet ovat kuitenkin hyvin monien *inhimillisten tekijöiden summia*, kuten tutkielman käsittekartasta (kuva 1) voidaan havaita. Tuotteen laatuun vaikuttavat niin asiakkaan kuin tuotteentekijöidenkin tiedot, taidot, kokemukset ja asenteet.



Kuva 1: Tutkielman käsittekartta

Taloudelliset tavoitteet ja tehokkuusvaatimukset voivat jakaa omistajan ja henkilökunnan vastakkaisille puolille, koska laatua mitataan yleensä rahasummiksi muutettuna. Pitkällä tähtäimellä voi olla edullisempaa, jos asetutaan yhteiseen rintamaan. Herää kysymys:

"Olisikohan ihmisten onnellisuudella jotain tekemistä prosessien tehokkuuden kanssa?"

Työelämän laadun ja tehokkuuden suhde onkin ajankohtainen tutkimuksen kohde [Työ02].

Ohjelmistotuotannon kiihtyminen on lisännyt kiirettä ohjelmistotaloissa. Ohjelmistojen suunnittelijat ja toteuttajat työskentelevät haastavissa tehtävissä. Haasteellinen ja kiireinen työ voi olla *stressaavaa*, mutta voiko se olla myös *motivoivaa*?

Tutkielmassa käydään läpi ohjelmistotuotannon keskeiset osa-alueet, kuten *ohjelmiston elinkaari, vaihejakomallit, suunnittelu- ja ohjelmointiparadigman valinta* sekä *projektityöskentely*, inhimillisten tekijöiden näkökulmasta. Huomioon otetaan ensisijaisesti ohjelmistotyön tekijät, joiden mahdolliset roolit selvittää myös. Mielenkiinnon kohteena ovat sekä *kognitiiviset* että *sosiaaliset taidot*. Tutkielmassa huomioidaan myös asiakkaan, markkinoiden sekä *tehokkuuden ja tuottavuuden* tärkeys ohjelmistotuotantoprosessissa.

Tutkimuksen kohteena on myös ihmisten *erilaisten taitojen tarpeellisuus*. Ohjelmistoalalla tarvitaan monenlaisia ihmisiä. Suurin osa heistä tarvitsee työssä selviytyäkseen teknisten taitojen lisäksi sosiaalisia taitoja, kuten viestintä- ja *vuorovaikutustaitoja*. Myös johtamistaitojen tarve on lisääntynyt organisaatioiden madaltuessa ja muotoutuessa *tiimiorganisaatioiksi*.

2 OHJELMISTOTUOTANTO

Ohjelmistotuotannossa on monia osa-alueita. *Ohjelmistotuote* käy kehitysvaiheessa läpi määrittelyn, suunnittelun, ohjelmoinnin ja testauksen. Tämän jälkeen tuote otetaan käyttöön ja alkaa ylläpitovaihe. Näitä vaiheita tukemaan tarvitaan mm. laadunvarmistusta, tuotteenhallintaa, dokumentointia ja vaatimustenhallintaa, jotka kaikki ovat osa *projektinhallintaa (project management)*. Ohjelmistotuotteet syntyvät usein projektien tuloksina, joten yhdessä ohjelmistotalossa on yleensä meneillään monta projektia, jolloin tarvitaan myös näiden hankkeiden välistä koordinoitua. Tätä kaikkea ohjaa yrityksen *liiketoiminta ja johtaminen*, jotka taas osaltaan vaikuttavat yrityksen laatujärjestelmään. *Laatujärjestelmä ohjaa kaikkia yrityksen toimia.* [HaM01]

2.1 ALAKOHTAISET ERITYISPIIRTEET

Ohjelmistotuotanto on melko nuori teollisuuden ala, joten sille ei ole määritelty vielä tarkkoja toimintamalleja ja standardeja. Vähitellen standardeja ja laatuvaatimuksia syntyy ja sitä mukaa niitä otetaan käyttöönkin. Alan nuoruudesta ja yllättävästä kasvusta johtuen alalla ei ole riittävästi päteviä henkilöitä, joilla olisi tarvittavaa koulutusta ja kokemusta. Useimpia alan yrityksiä vaivaa siis *henkilöstöpula* ja osittain tästä johtuen *kiire*. Ohjelmistotuotannon käyttöön tulee koko ajan uusia teknologioita ja uusia työkaluja. Näitä kaikkia ei voi kukaan hallita, mutta henkilöstöä on lähes pakko kouluttaa jatkuvasti, jotta osaamisen taso pysyy edes kilpailevien yritysten tasolla. [AvF95]

Kiire johtuu myös alalle tyypillisistä kiireisistä aikatauluvaatimuksista. Riippumatta yrityksen resursseista tuotteen on oltava asiakkaalla *sovittuna päivänä (deadline)* [AvF95]. *Nopeus markkinoilla* on tärkeä kilpailutekijä [Szy98]. Jos toimeksiannosta on kilpailua, aikataulut arvioidaan usein jo tarjousvaiheessa liian kireiksi, sillä usein yritys, joka arvioi aikataulunsa eniten alakanttiin, saa toimeksiannon. Tyypillisesti ongelmia ilmenee tuotteen ollessa lähes valmis eli testausvaiheessa, jolloin joudutaan tekemään ylitöitä, jotta asiakas saa tuotteensa toimivana [HaM01].

Asiakkaalle voidaan joutua tarjoamaan ala-arvoista tuotetta myös siksi, että sovitulla hinnalla ei pystytä parempaan. Tämä johtuu ohjelmistojen monimutkaisuudesta ja tarvittavan henkilöstön ja ajan puutteesta. Myös markkinointiosaston vähäinen tietämys

tuotteen ominaisuuksien ja valmistuskustannusten suhteesta voi aiheuttaa ongelmia aikataulujen kanssa. Erittäin tärkeä tekijä alalla on *markkinointi*, koska kilpailevia yrityksiä on enemmän kuin ennen. Pitkää tuotteenvalmistusprosessia ei kannata aloittaa ennen kuin markkinat on varmistettu. Markkinoiden puuttumien voi haitata muuten erittäin innovatiivista projektia, koska yritysjohto ei anna sille tukeaan. Tämä korostuu etenkin uudessa komponenttituotannossa. Huono tuote markkinoilla voidaan tehdä paremmaksi, mutta hyvä tuote ilman markkinoita on tarpeeton. Jos tuotteella on markkinoita, johtaa se yrityksen taloudellisen tilanteen kohenemiseen, mikä johtaa myös henkilöstön taloudelliseen ja henkiseen hyvinvointiin [Szy98].

Jos ohjelmiston valmistusta verrataan esimerkiksi auton valmistusprosessiin, ei riitä, että tuote saadaan asiakkaalle, vaan sen jälkeen alkaa *ylläpitovaihe (maintenance)*, joka voi olla hyvinkin työläs riippuen tuotteen toteutuksen onnistumisesta. Ohjelmistotalossa on tuotteita elinkaaren eri vaiheissa. Jotkut tuotteet ovat ideointitasolla ja projektiin liittyvää henkilöstöä koulutetaan. Toiset tuotteet taas ovat jo vuosia pyörineet ylläpitovaiheessa omalla painollaan. On selvää, että työntekijät voivat tuntea kateutta toisten töitä kohtaan, vaikka he kaikki tekevät tärkeää työtä. Johdon vastuulle jääkin kertoa kaikkien työpanoksen tärkeys, jottei talon sisällä synny ristiriitoja ainakaan tämän takia. [TaG96]

Kun tuotetta ryhdytään tekemään, ei se aina suju niin kuin on suunniteltu. Vaikka tuotteen vaatimusmäärittelyt on tehty ja ”jäädetyt”, on turhaa kuvitella, että vaatimukset eivät muuttuisi. Organisaatio ei ole stabiili, vaan liiketoimintaideat voivat muuttua ja ohjelmiston tulee tukea organisaation senhetkistä toimintatapaa. Yleensä asiakas ei myöskään heti alussa osaa määritellä, mitä haluaa. Mitä pidemmälle prosessissa päästään, sitä enemmän asiakaskin osaa vaatia. Siksi on yhtä tärkeää pystyä *hallitsemaan vaatimukseen* tulevia *muutoksia* kuin saada vaatimukset määriteltyä ja analysoitua. [Ja199]

Uusi ohjelmisto, joka otetaan käyttöön vaatii aina opettelua ja uuteen totuttelua. Voi olla, että samalla opetellaan myös uusia liiketoimintaprosesseja. On selvä, että loppukäyttäjät voivat olla muutosta vastaan. *Muutosvastarinta* on yleistä organisaatioissa. Osa henkilöstöstä saattaa pelätä myös, että tuleva ohjelmisto tekee hänen työnsä eikä häntä enää tarvita. Asiakkaaseen yhteydessä olevien ohjelmistoalan ammattilaisten onkin tärkeää pystyä osoittamaan, että tuotteen tarkoitus on helpottaa tulevia käyttäjiä eikä suinkaan korvata heitä tai vaikeuttaa heidän työtään [Haw01].

2.2 OHJELMISTON ELINKAARI

Ohjelmiston elinkaari koostuu *esitutkimuksesta (feasibility study / requirement study)*, *analyysistä (analysis)*, *suunnittelusta (design)*, *toteutuksesta (programming / implementation)*, *testauksesta (testing)*, *käyttöönotosta (conversion / operation)* ja *ylläpidosta (maintenance)*.

Esitutkimusvaiheessa pyritään osoittamaan, *miksi* koko ohjelmisto pitäisi toteuttaa tai miksi jotakin jo olemassa olevaa ohjelmistoa pitäisi muuttaa. Lisäksi selvitetään toteutettavuus huomioiden inhimilliset, tekniset ja taloudelliset kysymykset. Ennen analyysivaihetta ja osittain sen kanssa päällekkäin puhutaan määrittelystä, varsinkin vaatimusmäärittelystä. *Vaatimusmäärittely (requirements engineering)* on näkökulma siihen, mitä prosessissa on tehtävä eikä sitä voi suoranaisesti laittaa yhteen elinkaaren kohtaan. Vaatimusmäärittely painottuu kuitenkin prosessin alkuun.

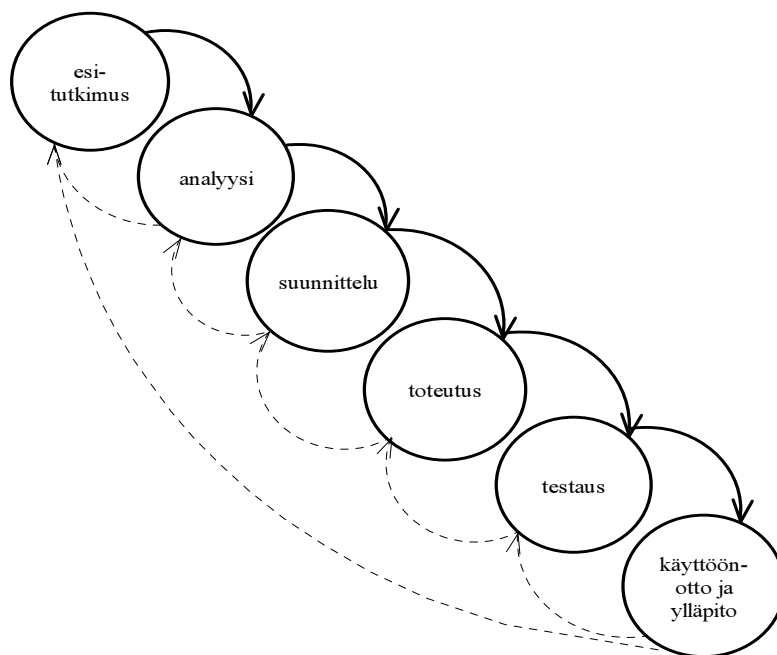
Analyysivaiheen tarkoitus on määritellä kohteena oleva järjestelmä mahdollisimman täydellisesti. Analyysivaiheen jälkeen tulee olla selvillä, *mitä* ohjelmisto sisältää. Suunnitteluvaiheessa keskitytään kysymykseen, *miten* määritellyt ominaisuudet ja toiminnot toteutetaan. Analyysi- ja suunnitteluvaiheisiin on olemassa erilaisia menetelmiä. Menetelmä syntyy yksikäsitteisestä notaatiosta ja systemaattisesta tavasta käyttää sovitua kuvaustapaa. Analyysi- ja suunnitteluvaiheiden menetelmät ovat kehittyneet sitä mukaa, kun ohjelmointikielet ovat monipuolistuneet ja ohjelmistot laajentuneet. Menetelmien avulla muokataan kerättyä tietoa vastaamaan kehitettävää ohjelmistoa. Apuna käytetään erilaisia kaavioita, joita voidaan vähitellen tarkentaa esimerkiksi kysymällä asiakkaalta yksityiskohtaisempia kysymyksiä. Erilaisista menetelmistä kerrotaan *luvussa kolme: Suunnittelu- ja ohjelmointiparadigman valinta*.

Toteutusvaiheessa suunnitelmat toteutetaan eli ohjelmisto koodataan. Tässä yhteydessä suoritetaan myös testaus. Kun jokainen ohjelmiston osa ja osista muodostuva kokonaisuus on testattu, voidaan tuote ottaa käyttöön. Käyttöönottovaiheessa huolehditaan koulutuksesta ja riittävästä tuesta [Som95], [Ken87]. Tämän jälkeen alkaa ylläpitovaihe. Ylläpitovaiheesta voidaan tehdä myös erillinen projekti, jolle suunnitellaan omat tavoitteensa [TaG96]. Mitä tahansa vaihejakomallia käytettäessä on syytä kiinnittää huomiota tulevaan ylläpitovaiheeseen, koska noin 70 % ohjelmiston elinkaarikustannuksista tulee tästä vaiheesta [Sch99, HaM01].

2.3 VAIHEJAKOMALLIT

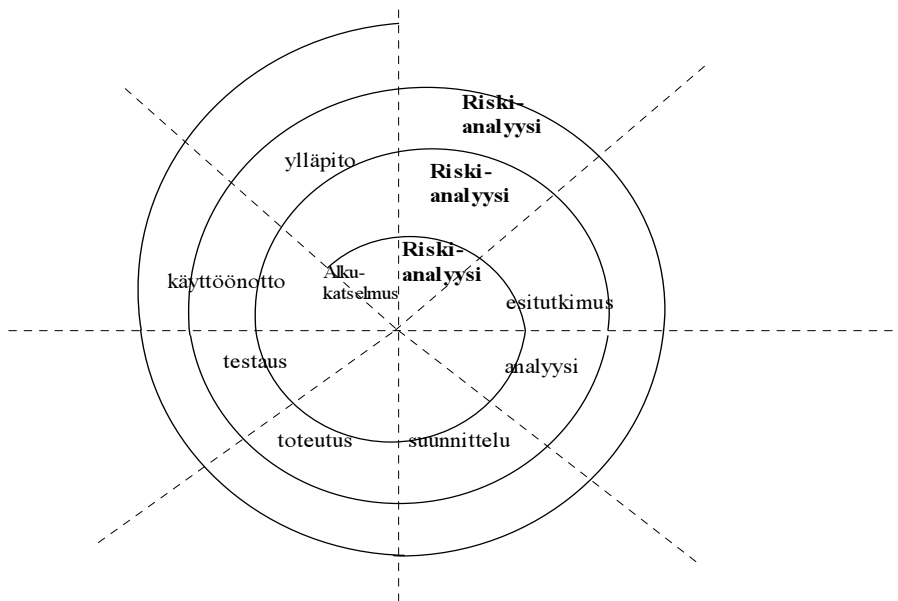
Riippuen vaihejakomallista ohjelmiston elinkaaren vaiheet voivat seurata toisiaan esimerkiksi peräkkäin, limittäin tai toistuen. Vaihejakomallien avulla kehitysprosessista on tehty *läpinäkyvämpi* (*visible*) ja täten hallittavampi kuin ennen mallien määrittelyä [Som95].

Ensimmäinen vaihejakomalli oli *vesiputousmalli* (*the waterfall model*). Sen esitti Royce W.W. vuonna 1970 [Roy70, Som95]. Vesiputousmalli (kuva 2) noudattaa peräkkäisyyttä niin, että jokainen vaihe on melko itsenäinen ja päättyy ennen seuraavaa vaihetta. Jokaisesta vaiheesta tehdään vaiheeseen liittyvät dokumentit. Mallia voidaan kutsua "dokumentointiorientoituneeksi". Vaiheistus jakaa prosessin hallittavissa oleviin palasiin, joiden väliin sopii hyvin *tarkastuspisteet* (*checkpoints*).



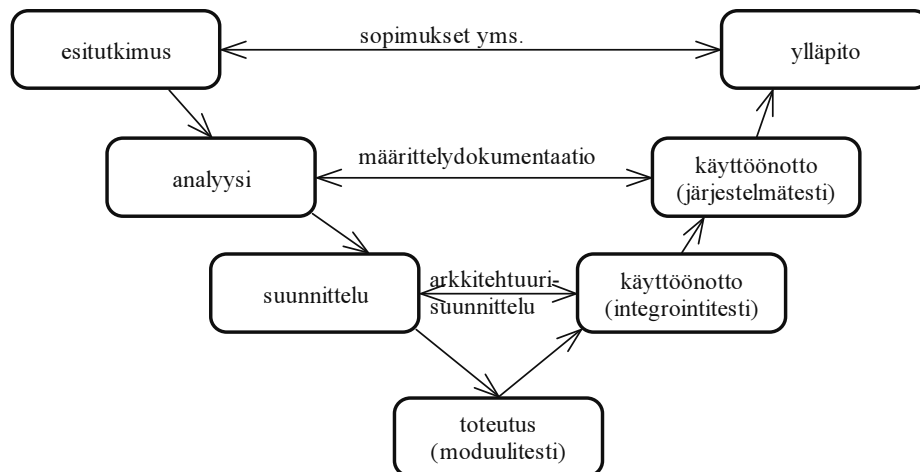
Kuva 2: Vesiputousmalli [Som95],[Ken87]

Spiraalimallin (kuva 3) esitti Barry Boehm 1988. Malli toteuttaa vaiheiden toistumisen niin, että kierros käydään läpi kaikkine vaiheineen ja aikaansaatu tulos otetaan taas määrittelyn kohteeksi. Spiraalimallissa painotetaan erityisesti *riskien hallintaa* [Som95]. Tämä tasaa työpainetta. Kun työpaine on kohtuullinen, ihminen pääsee parhaaseen tulokseen. Liian matala työpaine voi lamauttaa ja liian suuri stressaa [Toi94].



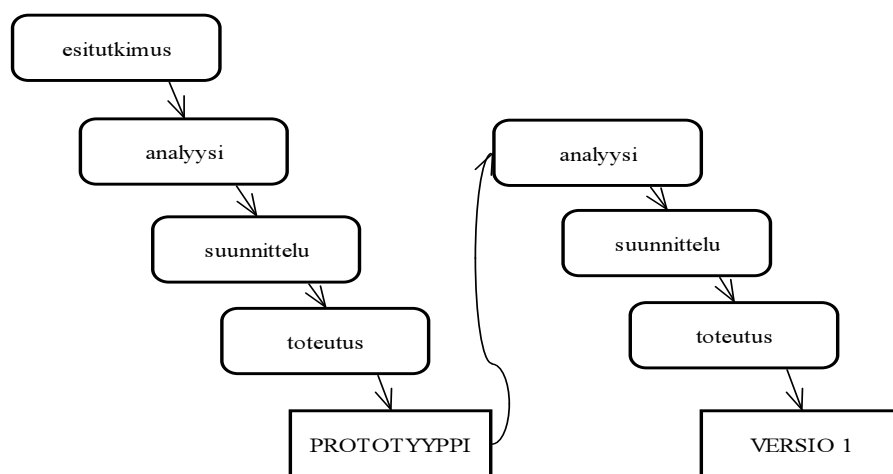
Kuva 3: Spiraalimalli [Som95], [TaG96]

V-mallin (kuva 4) mukainen vaihejako korostaa testausta siten, että analyysin ja suunnittelun tuotoksia (esim. hyväksymiskriteerit) käytetään suoraan hyödyksi elinkaaren loppupäässä. Jokaisen vaiheen jälkeen pidetään tarkastustilaisuudet, joissa käydään läpi kyseisessä vaiheessa tuotetut dokumentit ja muut tuotokset. Esimerkiksi toteutusvaiheessa tarkastuksen kohteena voi olla koodia, joka käydään läpi ryhmässä. Tarkastusmenettely riippuu yrityksen omasta laatujärjestelmästä. V-mallin mukaiset testaus- ja tarkastusmenetelmät antavat varmuutta ja vähentävät työn kasautumista prosessin loppupäähän. [HaM01]



Kuva 4: V-malli mukailtuna lähteestä [HaM01]

Protoilumalli (kuva 5) muistuttaa spiraalimallia, jossa vaiheet toistuvat. *Protoilussa (prototyping)* on tarkoitus saada tuote eli tuotteen prototyyppi asiakkaan käyttöön mahdollisimman nopeasti, jotta sitä voidaan kehittää asiakkaan haluamaan suuntaan. Tämä on hyvä tapa välttää asiakkaan pettymyksiä. Kun asiakas on tyytyväinen, häneltä tulee useimmiten kiitosta, mikä motivoi sovelluksen kehittäjiä yhä parempiin suorituksiin. Tässä mallissa korostetaan nopeaa prototyypin tekemistä eikä niinkään dokumentointia. Dokumentit viimeistellään sitten, kun ohjelmiston ensimmäinen versio on asiakkaalla. [AvF95]



Kuva 5: Protoilumalli [HaM01]

Protoilumallissa voidaan sanoa asiakkaan roolin korostuvan. Jos asiakas ei anna riittävästi palautetta prototyypistä, ei protoilusta ole hyötyä. Protoilumalli ei kuitenkaan edellytä asiakkaalta mitään määrittely- tai suunnittelutaitoja, kuten voidaan olettaa esimerkiksi vesiputousmallin ollessa kyseessä. Tuossa mallissa asiakkaan tulee olla tiiviisti mukana elinkaaren alkupäässä, jotta tuote täyttää valmistuessaan asiakkaan tarpeet. Protoilu konkretisoi asiakkaan ajatukset ennen valmista tuotetta ja mahdollistaa vielä halutut muutokset. Tästä johtuen prototyyppejä tehdäänkin myös muissa elinkaarimalleissa, vaikkakaan niiden rooli ei ole yhtä voimakas kuin protoilumallissa. Protoilua voidaan käyttää myös osana vaatimusmäärittelyä, ja käyttöliittymäsuunnittelussa se on erittäin hyödyllinen apuväline [Haw01].

EVO-malli (Evolutionary) on idealtaan samantyyppinen kuin protoilumalli (kuvassa 5). Siinä prototyypin sijaan asiakas saa ensimmäisen version tuotteesta. Lopullisen version toimittamispäivää ei välttämättä tiedetä. Seuraavan version toteuttaminen tosin voidaan aloittaa jo ennen kuin edellinen on asiakkaalla [HaM01]. EVO-malli on *kiivastahtinen* eikä siinä siksi ole hyödyllistä pitää dokumentaatioiden päivittämistä etusijalla. EVO-mallin ongelma on lähinnä sen nopeudessa. Kun tuote saadaan asiakkaalle, sen korjaamiseen asiakasta tyydyttäväksi menee paljon aikaa. [HaM01], [AvF95]

Uusin prosessimalli *RUP (Rational Unified Process)* [Kru00], on EVO- ja spiraalimallien yhdistelmä. Kyseessä on iteraatioihin perustuva malli. Mallin vaiheet ovat: *inception, elaboration, construction* ja *transition*. Näistä jokainen muodostaa oman vesiputousmallinsa siten, että jokaisessa vaiheessa käydään läpi yksi tai useampia iteraatioita. Tämä malli ottaa kantaa koko ohjelmistotuotantoprosessiin eikä vain tuotteen tekemiseen. Siinä on jokaisessa vaiheessa mukana mm. seuraavan vaiheen projektisuunnitelman tekeminen ja onnistumiskriteerien arviointi. RUP onkin enemmän kuin pelkkä vaihejakomalli; se antaa tuen ja välineistön kaikkiin ohjelmiston elinkaaren vaiheisiin ja tukee samalla mm. projektin hallintaa ja laadun toteutumista [JBR99].

Voidaan havaita, että vaihejakomallilla on merkitystä ainakin asiakkaan tietotaitojen kannalta. Asiakkaalla on prosessissa keskeinen rooli eikä se saisi olla liian vaativa, koska asiakkaan on tehtävä myös varsinaiset työnsä. Voi olla tarpeellista miettiä asiakkaan resursseja valittaessa vaihejakomallia. Ohjelmistotuotannon ammattilaiset näyttävät kuitenkin tarvitsevan tiettyjä toimintatapoja ja taitoja, kuten suunnitteluun ja toteutukseen liittyvät sekä tekniset että sosiaaliset taidot riippumatta vaihejakomallista.

Vaihejakomallin tietoisella valinnalla voidaan prosessiin tuoda tärkeää läpinäkyvyyttä ja selkeyttä ja tätä kautta turvallisuuden tunnetta. Tietoisuus sovitusta toimintatavoista on tärkeää [Hum98]. Tietoisuuteen päästään muun muassa dokumentoimalla yrityksen toimintatavat laatukäsikirjaan, kehittämällä kuvaavia malleja prosessista ja keskustelemalla ohjelmistojen kehitykseen osallistuvien henkilöiden kanssa [Haw01]. Vaikka henkilö osallistuisikin vain pieneen osa-alueeseen koko prosessiin nähden, kuten esimerkiksi ohjelmoija, motivoituu hän sitä paremmin, mitä enemmän hän ymmärtää kokonaisuudesta ja omasta osuudestaan tähän kokonaisuuteen. Yksilön motivaatioon palataan kappaleessa 4.3.1 Motivaatio.

2.4 LAATU

Ohjelmiston laatua kuvataan usein esimerkiksi sanoilla: luotettava, turvallinen, ymmärrettävä, testattava, käytettävä, tehokas, opittava ja modulaarinen [Som95]. Nämä adjektiivit kohdistuvat *lopputuotteeseen*. Laadun toteutuminen lopullisessa tuotteessa edellyttää kuitenkin laadun huomioimista jo ohjelmistotuotantoprosessin aikana. Toisaalta laatuvaatimuksia voidaan kohdistaa myös *tuotantoprosessiin*. Tällaisia laatuvaatimuksia voi olla esimerkiksi tiettyjen työvaiheiden järjestys sekä tarkastustilaisuudet ja katselmukset ennen seuraavaan vaiheeseen etenemistä.

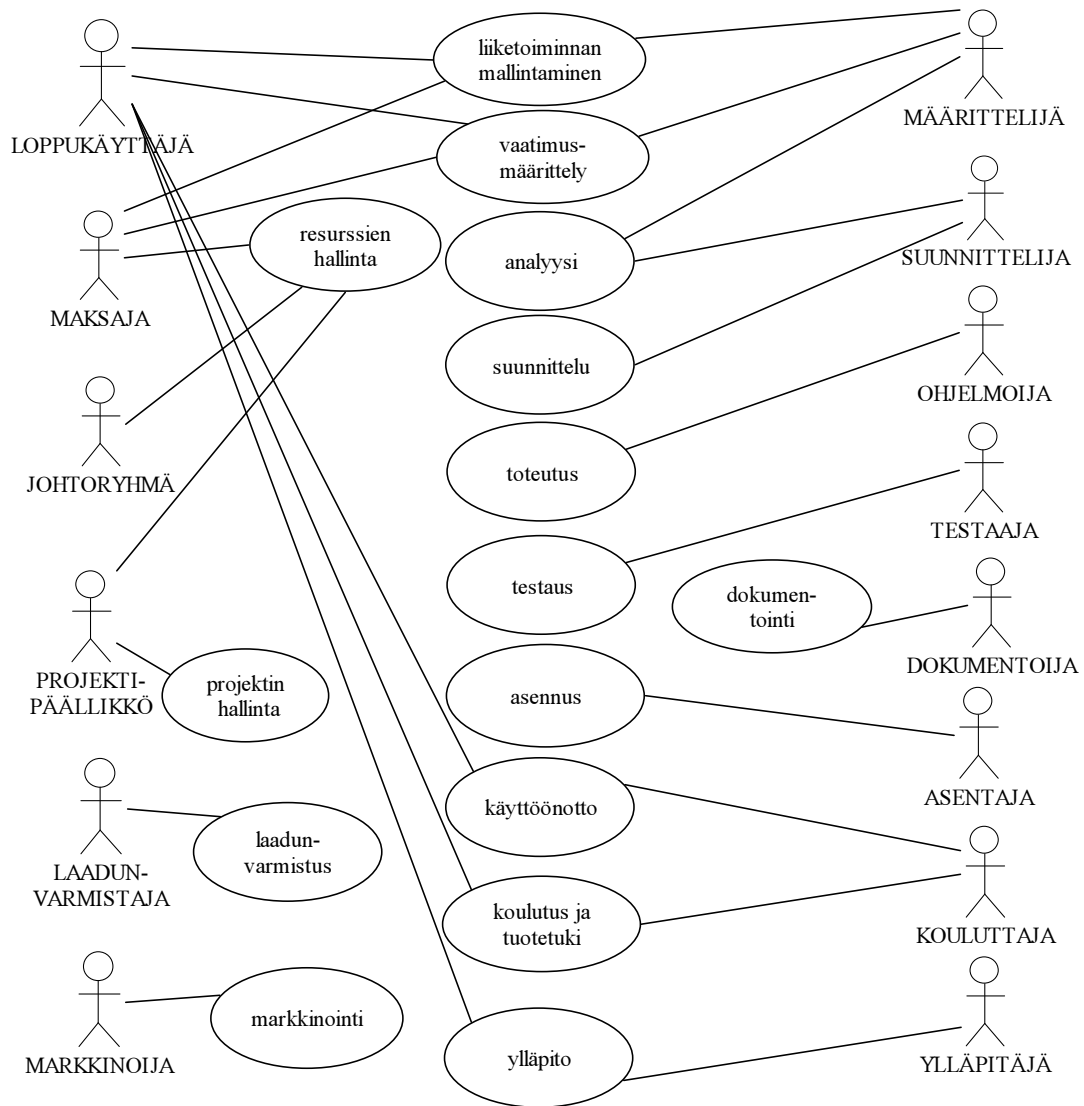
Ohjelmistoprosessissa laatuun liitetään käsite *laadunvarmistus*, joka on riittävää testausta, *verifiointia* (=todentaminen) sekä *validointia* (=kelpoistaminen). Verifiointi tarkoittaa tarkastusta, jolla selvitetään, vastaako tuote sille asetetut vaatimukset eli vastaako saatu tulos vaatimuksia. Validoinnissa varmistetaan, että ollaan tehty oikeanlainen tuote sen käyttötarkoitus huomioiden [Som95]. Testaus on osa laadunvarmistusta, vaikka se voidaan nähdä myös yhtenä vaiheena ohjelmiston elinkaareissa. Testauksen painotus riippuukin valitusta vaihejakomallista.

Yrityksessä valitaan laatupolitiikka, joka määrittelee yrityksen laadun tason ja joka on kirjattu yrityksen laatukäsikirjaan. Yrityksen laatujärjestelmä voi noudattaa jotakin standardia tai olla sellaisten yhdistelmä sovellettuna yrityksen omiin tarpeisiin. [HaM01] Laatua arvioidaan laatuvaatimusten toteutumisen mukaan. Jos yritys saavuttaa tietyn laatutason, voi se hakea itselleen laatusertifikaattia. Myös yrityksen tuottavuus koetaan laatuna, jolloin laatua mitataan rahasummiksi muutettuna. Laadun mittaaminen voisi kuitenkin olla "tavoitteiden saavuttamisen seuraamista, poikkeamien tunnistamista ja

tarvittavien korjaustoimenpiteiden tekemistä" [Pyl99]. Tähän pyritäänkin esimerkiksi laatujohtamisen perusteissa [Mau96].

2.5 OHJELMISTOTUOTANTOPROSESSIN TOIMIJAT

Ohjelmistotuotantoprosessiin osallistuu useita ihmisiä, joista vain osa on ohjelmistotuotannon ammattilaisia. Tässä kappaleessa käsitellään ohjelmistotuotantoprosessiin osallistuvia ihmisiä *toimijoina (actors)*, koska tämä on luonteva tapa ilmaista, että kyse ei ole aina eri ihmisestä, vaikka puhutaan eri toimijoista. Kyseessä on eräänlainen rooli suhteessa prosessiin. Kuvassa 6 käytetään *UML:n (Unified Modelling Language)* [Fow00] mukaista notaatiota ja *käyttötapauskaavion (use-case-diagram)* mukaista periaatetta, jotta ohjelmistotuotantoprosessiin tarvittavat toimijat ja aliprosessit tulevat esiin.



Kuva 6: Ohjelmistotuotantoprosessin aliprosessit ja niihin osallistuvat toimijat

Prosessiin tarvitaan asiakas. Asiakas on tuotteen *maksaja* ja määrää siten resursseista ja osallistuu vaatimusmäärittelyyn. Asiakkaan puolelta tulisi olla edustettuna myös tuleva *loppukäyttäjä*, joka osallistuu vaatimusmäärittelyyn ja käyttöönottoon sekä tiedottaa ylläpidolle mahdollisista virheistä ohjelmistossa [RoR99].

Ennen ohjelmistolle asetettavia vaatimuksia tulee tietää asiakkaan liiketoimintaprosesseista riittävästi, jotta tuote vastaisi yrityksen tarpeita. Liiketoimintaprosessit mallinnetaan ennen vaatimusmäärittelyä ja tämä voi tapahtua esimerkiksi haastattelujen ja työn seuraamisen avulla [Haw01]. Jotta ohjelmisto valmistuessaan vastaisi asiakkaan tarpeita, on

määrittelijöiden ja asiakkaan ymmärrettävä toisiaan. Tyypillisesti joka alan ammattilaisilla on omat ammattiterminsä. Ohjelmistoalan ihmisten on puhuttava asiakkaan ”kieltä” [Haw01].

Voidaan olettaa, että loppukäyttäjä osallistuu ohjelmistotuotantoprosessissa muihinkin aliprosesseihin kuin vaatimusmäärittelyyn ennen tuotteen valmistumista, mutta se voidaan kuvata vaatimusten tarkentumiseksi eli jatkuvaksi *vaatimustenhallintaprosessiksi*. Vaatimusmäärittelyn ja analyysin välinen raja on hieman epämääräinen. Tämä johtuu siitä, että analysointi aloitetaan jo ennen kuin kaikkia vaatimuksia on saatu määriteltyä. Mitä pidemmälle prosessi etenee sitä enemmän siirrytään kohti analyysiä ja vaatimusten *määrittely* vaihtuu vaatimusten *hallinnaksi* [RoR99].

Tuotteen valmistajan puolelta tarvitaan projektille tiimi, jossa toimii *projektipäällikkö, määrittelijä, suunnittelija, ohjelmoija ja testaaja*, joista jokainen on myös *dokumentoija ja laadunvarmistaja*. *Dokumentoijan* tehtävä on dokumentoida huolella jokaisen vaiheen tulokset sovitun käytännön mukaan sekä pitää dokumentit ajan tasalla. *Laadunvarmistaja* huolehtii tarkastuksista ja katselmuksista sovitun laatupolitiikan mukaisesti. *Projektipäällikkö* on vastuussa projektinhallinnasta ja tiedonkulusta johtoryhmän suuntaan. (Projektinhallinnasta kerrotaan *luvussa viisi: Projektityöskentely*.) Tuotteen valmistumisen jälkeen tarvitaan *asentaja*, joka asentaa ohjelmiston ja käyttöönoton yhteydessä tarvittavat laitteet ja sovellukset sekä *kouluttaja*, joka huolehtii loppukäyttäjien koulutuksesta ja tuotteen onnistuneesta käyttöönotosta. *Ylläpitäjä* voi olla tuotteen tekemiseen osallistunut tai ulkopuolinen toimija. Ylläpitäjä kommunikoi loppukäyttäjän kanssa.

Johtoryhmään osallistuu ohjelmistotalon ja asiakaspuolen johtotason ihmisiä, jotka tarkkailevat resursseja yhdessä projektipäällikön kanssa [Ste96].

Yksi tärkeäksi muodostuva aliprosessi on markkinointi, sillä tuote täytyy aina myös myydä. Tuotteen määrittelijät ja toteuttajat voivat periaatteessa olla välillä *markkinoijan* roolissa, mutta useimmiten markkinoinnin hoitaa henkilö, joka ei tiedä juuri mitään valmistusprosessista tai siitä, mitä on mahdollista toteuttaa tai miten paljon työtä tietynlaisen ominaisuuden toteuttaminen teettää. Tämä voi muodostua ongelmaksi, jos tiedonkulusta toteuttajien ja markkinoijan välillä ei huolehdi. Markkinoija on perinteisesti tekemisissä asiakkaan johtoportaan kanssa.

Edellä on määritelty ohjelmistotuotantoprosessin läpiviemiseen ja ylläpitoon tarvittavat toimijat. Jokaiselta toimijalta odotetaan erilaisia taitoja ja ominaisuuksia. Esimerkiksi

määrittelijän tulee osata hahmottaa kokonaisuuksia ja testaajan tulee olla pikkutarkka. Selvää on kuitenkin tarve vuorovaikutustaitojen hallintaan ja *vastuullisuuteen* jokaisella toimijalla tiiminjäsenenä. Vaikka yksi ihminen voi olla useammassa roolissa prosessin aikana, ei hän pysty viemään koko prosessia yksin läpi, koska ohjelmistot ovat tänä päivänä monimutkaisia ja isoja kokonaisuuksia. Vuorovaikutustaitoihin palataan *luvussa kuusi: Vuorovaikutustaidot*.

2.6 IHMISET

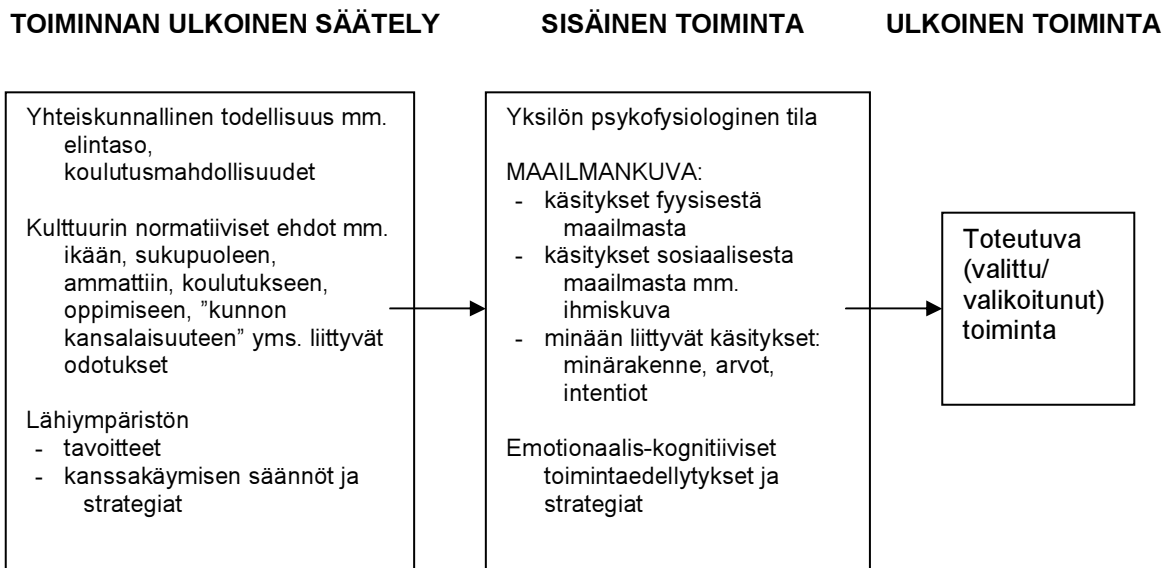
Ohjelmistotuotannon tärkein tekijä on sitä tekevät ihmiset. Ohjelmistotuotanto on *sosiaalista* ja *kognitiivista* toimintaa. Tämän ymmärtäminen on edellytys menestyksekkäälle projektin hallinnalle ja sitä kautta myös laadukkaalle tuotantoprosessille. [Som95]

Ohjelmistotuotannon laatua ja tuottavuutta on pyritty parantamaan monin teknisin apuvälinein kuten projektinhallintatyökaluin. Koska tekniset apuvälineet eivät yksin auta, on tärkeää keskittyä ihmisiin. *Usein ohjelmistoprojektit epäonnistuvat juuri inhimillisten tekijöiden vuoksi* [HaM01]. Ihmisten tulee toiminnallisen ja teknisen osaamisen lisäksi hallita projektityö sekä sitoutua toimimaan valitun tavoitteen ja menetelmän mukaisesti. ”Osaavien ihmisten motivointi tapahtuu kouluttamalla heidät toimiviksi tiimeiksi ja kytkemällä heidän henkilökohtaiset tavoitteensa projektin tavoitteisiin [Py199].”

2.6.1 Sosiaaliset taidot

Ihminen on pienin yksikkö sosiaalisten systeemien hierarkiassa. Yksilö voi olla jäsen perheessä, harrastusyhteisössä, työyhteisössä, projektiryhmässä ja asiantuntijatiimissä, jotka ovat osa organisaatioita, yhteiskuntaa ja ihmiskuntaa. Tästä johtuen kaikki vuorovaikutustilanteet ovat sidoksissa laajaan sosiaaliseen kokonaisuuteen.

Kuvassa 7 havainnollistetaan yksilön toiminnan *sosiaalista säätelyä* ja sitä selittäviä tekijöitä. Esimerkiksi vuorovaikutustilanteeseen työtoverin kanssa vaikuttavat henkilön sisäisen toiminnan, esim. maailmankuvan, lisäksi kulttuuritekijät, kuten koulutustaso, ikä ja sukupuoli.



Kuva 7: Toiminnan sosiaalinen säätely Rauste-von Wrightin mukaan [RaW97]

Sosiaaliset taidot kehittyvät elämän sosiaalisissa tilanteissa. Kotona, koulussa, työpaikoilla ja harrastuksissa saadaan palautetta käytöksestä. Tähän palautteeseen vaikuttavat tietenkin monimuotoiset tekijät, kuten yhteiskunnan elintaso, ammatilliset näkökulmat ja lähiympäristön omaksumat normit eli ”sallitut käytöstavat”. Tämä palaute muokkaa yksilön omia sosiaalisia normeja sekä *sisäistä maailmankuvaa* (ks. kuva 7).

Ihminen on päivittäin vuorovaikutuksessa *ystävien, tuttujen* ja *vieraiden* ihmisten kanssa. Ystävien ja tuttujen kanssa tilanteet pysyvät useimmiten normien mukaisina. Vieraat ihmiset voivat yllättää tekemällä jotakin ns. normien vastaista, kuten tupakoida kielletyllä alueella, soittaa musiikkia liian kovaa tai istua ruuhkabussissa invalideille ja vanhuksille varatulle paikalle. Luultavasti vieraan ihmisen käytökseen ei puututa niin helposti kuin tutun ennen kuin sääntöjen rikkomisesta näyttää seuraavan jotakin tuhoisaa. Toisaalta sääntöjen rikkoja ei aina välttämättä edes tiedä tai ymmärrä rikkovansa sääntöjä. [Duc86]

Vaikka organisaatioissa voidaankin pyrkiä viileän asialliseen vuorovaikutukseen, sosiaalisessa vuorovaikutuksessa ovat useimmiten mukana myös *tunteet*. Emotionaalis-kognitiiviset toimintaedellytykset ja strategiat opitaan tahtomattakin ja ne muodostuvat jokaiselle yksilöllisesti. Sosiaaliseen toimintaan vaikuttaa myös psykofysiologinen tila. Jos henkilöllä on nälkä tai hän on väsynyt, voi vuorovaikutustilanne olla jotakin muuta kuin siinä tapauksessa, että tämä henkilö olisi kylläinen ja levännyt.

Yleisimpiä *sosiaalisiiin tilanteisiin* liittyviä tunteita ovat rakkaus ja mustasukkaisuus sekä hämmentyneisyys, ujous ja yksinäisyyden tunne [Duc86]. Rakkautta on siis romanttisen rakkauden lisäksi arvostukseen ja huolenpitoon perustuvaa ”ystävyyssrakkautta”, käytännöllistä asenteisiin, uskontoon ja harrastuksiin liittyvää rakkautta, omistushaluista mustasukkaista rakkautta ja epäitsekästä, kaiken antavaa rakkautta. Myös mustasukkaisuus voi ilmetä monella tavalla. Esimerkiksi tiimin vaihtuessa voidaan olla mustasukkaisia vanhoista tiiminjäsenistä, koska he tutustuvat uusiin ihmisiin ja viettävät aikaansa uusien ihmisten kanssa. Myös esimiehen huomiosta voidaan olla mustasukkaisia. Hämmentyneisyys ja ujous liittyvät pelkoon kasvojen menetyksestä [Duc86]. Varsinkin suomalaiset ovat perinteisesti mieluummin hiljaa, jollei heiltä nimenomaan ole kysytty jotakin [RaW97]. Tunteista ja niiden vaikutuksesta erilaisiin tilanteisiin kerrotaan *kappaleessa 6.3 Tunneäly*.

2.6.2 Kognitiiviset taidot

Kognitiivisilla toiminnoilla tarkoitetaan tiedonkäsittelyyn liittyviä tiedollisia, mielensisäisiä toimintoja, kuten muistaminen. Kognitiiviset taidot mahdollistavat maailman havaitsemisen, ajattelun ja *ongelmanratkaisun*. Ympäröivästä todellisuudesta syntyy aivoihin *yksilöllisiä kuvia (representaatioita)* eli edustuksia, jotka taas suuntaavat *tarkkaavaisuutta* ja havainnointia. Jokaisella on omanlaisensa *skeemat* asioista. Silti emme voi sanoa jollain toisella olevan oikeammat tai paremmat representaatiot tai skeemat kuin toisella [EyK95].

Skeemojen avulla laajat ja monimutkaiset työt pystytään hahmottamaan helpommin. Esimerkiksi kirjan kirjoittamisen voi aloittaa hahmottelemalla joko sisällysluettelo tai käsitekarttaa. Ensimmäinen luonnos toimii jatkossa skeemana, jota tarkennetaan ja korjataan tarvittaessa. Myös ohjelmistosuunnittelu on skeemojen hahmottelua ja tarkentamista erilaisten kuvien ja kaavioiden avulla.

Muisti on kognitiivisten toimintojen tukijalka. Tiedonkäsittely ei olisi mahdollista, jos käsiteltäviä asioita ei voisi laittaa muistiin. *Pitkäkestoisessa muistissa* on mielikuvia, eräänlaisia muistilappuja ja lisäksi propositionaalisia esityksiä eli abstrakteja malleja ympäröivästä maailmasta. Tämän lisäksi pitkäkestoiseen muistiin on säilytetty *tietorakenteet*.

Työmuisti puolestaan asettaa omat haasteensa esimerkiksi oppimiselle. Kerrallaan voidaan käsitellä n. 5-7 yksikköä tietoa. Tämä tarkoittaa sitä, että uuteen asiaan tutustuttaessa voidaan omaksua vain vähän tietoa. Seuraavan kerran käsitellessä asiaan liittyvää tietoa on jo olemassa tuo 5 yksikön skeema pitkäkestoisessa muistissa, josta se voidaan palauttaa mieleen ja jokaiseen yksikköön voidaan liittää lisää tietoa [EyK95]. Ohjelmistosuunnittelussa käytettävät analyysimallit huomioivat työmuistin rajoitukset sallimalla yhteen kaavioon vain tietyn määrän asioita, esimerkiksi tietovirtakaavioissa näytetään 5-9 prosessia yhdellä tasolla.

Tutkimuksissa on osoitettu, että iän myötä kognitiiviset taidot heikkenevät [SuR99]. Näin käy varsinkin nopeudelle ja uudenoppimiselle. Hidastumisesta ei kuitenkaan ole yleensä haittaa arkipäivän askareissa. Romahdusmaista hidastumista tapahtuu vasta 80 vuoden iässä tai sairauksien myötä. Jotain varsin pysyvääkin kuitenkin on, kuten kiteytynyt älykkyys ja kielellinen lahjakkuus. Nämä näyttävät olevan pysyviä kognitiivisia taitoja. Esimerkiksi yli 80-vuotiaan kielellinen lahjakkuus voi olla 25-vuotiaan tasolla [SuR99]. Kognitiiviset taidot voivat heikentyä hetkellisesti myös esimerkiksi stressin tai masennuksen seurauksena.

Voidaan asettaa vastakkain myös kognitiivisten taitojen *nopeus* ja *laatu*, jolloin iän myötä laatu paranee, vaikka vasteajat pitenevät. Vaikuttaa siltä, että ei kannata enää kovin vanhana vaihtaa alaa varsinkaan sellaiseksi, jossa pitää oppia paljon ja nopeasti. Toisaalta kehittyä voi edelleen omalla vahvalla alallaan [EyK95]. Kokonaan uuden alan opettelu vaatii paljon työtä jo 50-vuotiaalta, mutta toisaalta tätäkin vanhempi pystyy omaksumaan uutta tietoa omalta alaltaan. Esimerkiksi proseduraalisen ohjelmointiparadigman oppineet oppivat helpommin olio-ohjelmoinnin kuin esimerkiksi terveydenhuollon ammattilaiset.

Sanonta: "vanha ja viisas", pitää paikkansa siinä määrin, että kokemusten ja tiedon karttuessa myös tiedonkäsittely paranee. Vaikka nopeus ei pysyisikään entisellään, niin asiantuntijalla on niin hyvät rutiinit työhönsä, että hän päihittää noviisin. Iän myötä lisääntyy myös kyky *harkita* asioita ja *nähdä laajempia kokonaisuuksia*, jolloin myös nuoruuden rohkeus häviää. Kokemukset ja esimerkiksi kouluttautuminen voivat harjoittaa ja kehittää kognitiivisia strategioita. Tietorakenteet kehittyvät sellaisiksi, että ne ottavat vastaan ja käsittelevät tietoa paremmin eli entistä tehokkaammin. [EyK95]

Eksperttien huikeat tietomäärät ovat mahdollisia *hyvien tietorakenteiden* ansioista. Lohdullista on todellakin se, että kyse ei ole ainoastaan lahjakkuudesta, vaan ekspertin pitkäkestoisesta, esimerkiksi 10 vuotta kestäneestä työn tekemisestä. [EyK95]

Ohjelmistotuotannossa tarvitaan varsinkin *havainnointia* esimerkiksi asiakkaan vaatimuksia kerätessä ja *ongelmanratkaisukykyä*, esimerkiksi ohjelmoidessa. Tilanteet ovat joka kerta erilaisia. Kuitenkin vanhan tiedon pohjalta osataan ratkaista uudet ongelmat. Uusiin ongelmatilanteisiin sovelletaan aikaisempia toimintamalleja. Joskus vanha toimintamalli toimii hyvin, mutta usein tarvitaan myös uutta mallia tai vanhan soveltamista uudella tavalla. Hyväksi havaittu toimintamalli voi rajoittaa luovuutta, kun samaa mallia yritetään soveltaa ongelmiin, joihin olisi parempiakin ratkaisuja [EyK95]. Mitä useampia yksittäisiä ongelmia henkilö on ratkaissut, sitä enemmän hänelle on kertynyt erilaisia ratkaisumalleja, mikä johtaa entistä tehokkaampaan ongelmanratkaisuun [Mod01].

Esimerkiksi ohjelmointia voi olla vaikea oppia ja ohjelmoidessa tulee usein virheitä. Tämä johtuu osittain siitä, että ohjelmointikielellä ongelmat voidaan ratkaista paljolti eri tavalla kuin todellisessa maailmassa. Ohjelmoijat voivat käyttää ohjelmointiongelmien ratkaisemiseen muussa yhteydessä oppimiansa ratkaisumalleja, jolloin virheitä syntyy. [EyK95]

3 SUUNNITTELU- JA OHJELMOINTIPARADIGMAN VALINTA

Suunnittelu- ja ohjelmointiparadigman valinnan voidaan olettaa vaikuttavan projekteissa tarvittaviin suuntaviivoihin ja projektityömenetelmiin. Riippuen paradigmasta voi työnjako olla erilainen. Paradigma vaikuttaa myös siihen, millaisia kognitiivisia ja sosiaalisia taitoja tarvitaan.

Rakenteinen (structured) analyysi ja suunnittelu johtaa proseduraaliseen ohjelmaan [DeM78]. Rakenteisen analyysin ja suunnittelun parhaita puolia lienee suoraviivainen eteneminen tarkkojen, tasoittain tarkentuvien määrittelyjen kautta ohjelman toteuttamiseen. Ongelmana on toimintopainotteisuus, joka johtaa runsaasti muutoksia tarvitseviin ja vaikeasti ylläpidettäviin ohjelmiin, koska toiminnot ovat nopeammin muuttuvia kuin tiedot [CoY90].

Rakenteisuuden rinnalle tuli 1990-luvun taitteessa voimakkaasti *oliokeskeinen (object-oriented)* analyysi ja suunnittelu (OOA/D). Tämä johtui osittain siitä, että 1980-luvun puolivälissä oli kehitetty toimivia olioperustaisia ohjelmointikieliä, esim. C-kieleen perustuva C++ sekä Pascalin ja Lispin oliolaajennokset [Kos00]. Tuolloin korostettiin sitä, kuinka oliopohjainen ajattelu mullistaa myös projektit [Oli92]. Oliokeskeisydessä pääpaino ja lähtökohta on tiedoissa ja käsitteissä, jolloin päästään stabiilimpiin ohjelmistoihin.

Tänä päivänä, 2000-luvulla, puhutaan *komponenttipohjaisesta* sovelluskehityksestä [Szy99], [Hes00]. Komponenttipohjainen sovelluskehitys mahdollistaa oliokeskeisten työmenetelmien tapaan reaali maailman käsitteiden käytön kehitystyössä. Lisäksi se mahdollistaa uudenlaisen työnjaon. Puhutaankin "ihmisen kokoisista" komponenteista. Tämä tarkoittaa sitä, että yksi komponentti on sen kokoinen, että yksi ihminen voi sen hallita. Nykyisin ohjelmistot alkavat olla niin suuria ja monimutkaisia, ettei niitä yksi ihminen voi hallita. Komponentteihin jako ja komponenteista koostaminen mahdollistaa paremman ohjelmistojen hallinnan.

3.1 ENNEN OLIOKESKEISYYDEN LÄPIMURTOA

1970- ja 1980-luvuilla ohjelmistot toteutettiin enimmäkseen proseduraaliseen-, funktionaaliseen- tai logiikkaohjelmointiin perustuvina ohjelmina, joiden suunnitteluun

käytettiin esimerkiksi käsiteanalyysiä ja rakenteista lähestymistapaa. Ennen näitä käytettiin myös esimerkiksi *ISAC:ia (Information Systems Work and Analysis of Change)*, jota voidaan sanoa ihmislähtöiseksi. ISAC-menetelmässä painopiste on muutosanalyysissä ja muutosprosessissa, joita tarkastellaan käyttäjien ongelmien ratkaisemisen kautta [AvF95].

Rakenteinen analyysi (Structured Analysis) [DeM78] keskittyy prosesseihin ja *käsiteanalyysi (Entity-Relationship Analysis)* [Che76] puolestaan tarvittaviin tietoihin. Myöhemmin kehitetty *moderni rakenteinen analyysi (Modern Structured Analysis)* sisältääkin myös käsiteanalyysin [You89]. Myös *JSD-menetelmään (Jacson's System Development)* [Jac83] kuuluu sekä toimintojen että käsitteiden tarkastelun. JSD-menetelmään liittyy myös ajatus reaali maailman käsitteiden viemisestä systeemiin.

Aikaisemmin oli tärkeää kiinnittää huomiota esimerkiksi muistiin tallennettavien tietojen kokoon ja toiminnot toteuttavien algoritmien nopeuteen. Teknologian kehittyessä ei enää tarvitse kiinnittää niin suurta huomioita noihin asioihin, vaan voidaan keskittyä paremmin loppukäyttäjän tarpeisiin ja muihin ohjelman ominaisuuksiin, kuten uudelleenkäytettävyys, testattavuus, ymmärrettävyys ja luotettavuus, käytettävyys ja opittavuus. Eli periaatteessa oliokeskeisyys ja teknologian kehitys ovat asettaneet ohjelmistoille uusia vaatimuksia vain siksi, että näiden laatuvaatimusten toteuttaminen on tullut mahdolliseksi.

3.2 OLIOKESKEINEN SOVELLUSKEHITYS

Ohjelmistojen *laadun* uskottiin paranevan oliokeskeisten systeemityömenetelmien myötä. Koska oliokeskeiset menetelmät mahdollistavat paremmin myös loppukäyttäjien ja liiketoiminnan asiantuntijoiden mukaantulon prosessiin, uskottiin oliokeskeisyyden ratkaisevan ongelmat, joita syntyy, kun lopputuote ei vastaa odotuksia. Ideana on, että reaali maailman *oliosta (object)* tehdään systeemin sisäinen olio, joka elää analyysi- ja suunnitteluvaiheista aina valmiiseen tuotteeseen asti samannimisenä [Put94]. Tämä auttaa systeemin ymmärtämisessä ilman teknisten yksityiskohtien hallitsemista ja täten mahdollistaa esimerkiksi loppukäyttäjän osallistumisen tuotantoprosessiin. Kun yrityksen liiketoiminnan käsitteitä siirretään sellaisenaan ohjelmistoihin, tukevat ne paremmin yrityksen tarpeita [Oli92].

Tuottavuus paranee, koska oliokeskeisillä työmenetelmillä saadaan paremmin uudelleenkäytettävää koodia eli luokkia ja luokkakirjastoja, joita voidaan tarvittaessa ostaa jopa valmiina. Tällöin käytettävissä olevat resurssit voidaan suunnata paremmin.

Uudelleenkäytettävyyden uskotaan ulottuvan suunnittelutyöhönkin. Hyvin suunniteltua voidaan käyttää uudelleen. Tänä päivänä olioparadigmasta on saatavissa eniten hyötyä monimutkaisten ohjelmistojen uudelleenkäytettävyyden ja ylläpidettävyyden haasteisiin vastattaessa. Pelkkä olio-ohjelmointikieli ei auta kuitenkaan uudelleenkäytön lisäämisessä, jos ohjelmia ei ole suunniteltu ajatellen nimenomaan mahdollista uudelleenkäyttöä [Kos00].

Olioparadigma ei ole myöskään liian tieto- tai vastaavasti liian toimintopainotteinen. Olio kapseloi sisäänsä tiedon sekä toiminnot, joilla kyseistä tietoa voidaan käsitellä, joten oliokeskeinen suunnittelu vastaa samalla kerralla tarvittavien tietojen ja toimintojen suunnittelusta [Kos00]. On kuitenkin syytä muistaa, että tällaisia menetelmiä on ollut jo aikaisemminkin, esim. JSD-menetelmä [Jac83].

Oliokeskeisyys toi tullessaan uudenlaisen työnjaon sovelluskehittäjille. Toiset voivat ohjelmoida oman erikoisosaamisensa mukaisia luokkakirjastoja ja toiset pystyvät hyödyntämään niitä ja kokoavat tarvittavia kokonaisuuksia valmiista luokista.

3.2.1 Periytyminen ja riippuvuudet

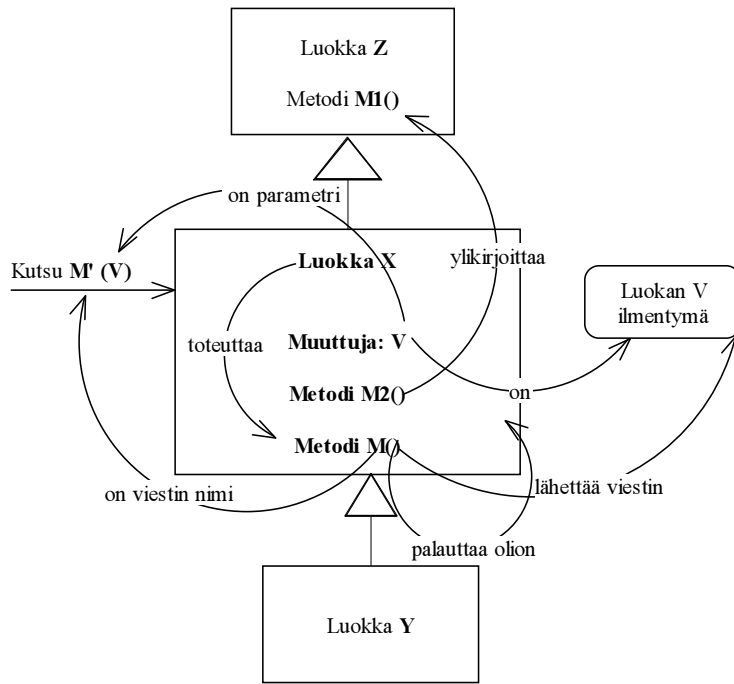
Oikeastaan ainut olioparadigman tuoma uusi piirre on *periytyminen* [Kos00]. Se pakottaa proseduraalisiin ohjelmiin tottuneet uudennlaiseen ajatusmalliin. Tämä olikin osasy s siihen, ettei 1960-luvulla kehitetty olioperustainen kieli, Simula [Dah68] eikä 1970-luvulla kehitetty ns. puhdas oliokieli, Smalltalk saavuttanut suosiota. Ajatustavan muuttaminen ei onnistu ilman ongelmia. Vanhan proseduraalisen ajattelutavan ”poisoppiminen” vaatii aikaa. On huomattava, että esimerkiksi C++-kielellä tehty ohjelma ei välttämättä tarvitse olla toteutettu aidolla olioperiaatteella, jolloin ei tietenkään saavuteta haluttuja ominaisuuksia, kuten ylläpidettävyyys. Sama ongelma seuraa muitakin hybridikieliä, eli kieliä, jotka ovat perinteisten kielten laajennuksia. Hybridikielet mahdollistavat olioiden käytön, mutta eivät pakota siihen, kuten ns. puhtaat oliokielet, esimerkiksi 90-luvulla kehitetyt Java ja Eiffel. [HaM01], [Kos00]

Olioita etsitään ensin reaali maailmasta ja ne sijoitetaan sopiviin luokkiin niin, että samankaltaiset oliot ovat samassa luokassa. *Periytyminen* tarkoittaa, että yläluokan ominaisuudet ja metodit voidaan periä sellaisenaan tai uudelleenmääriteltyinä aliluokalleen. Tästä seuraa, että kerran määriteltyä ominaisuutta ei tarvitse määritellä uudelleen, vaan se voidaan periä. Esimerkiksi jos asiakas-olio on jo määritelty, voidaan

sen ominaisuudet periyttää lapsiasiakas-oliolle samalla kun lisätään lapsiasiakas-olion omat ominaisuudet systeemiin [CoY90]. Koska kognitiivisessa ongelmanratkaisussa ihminen yrittää aina ensin jo ennestään tuttua ratkaisumallia, voi reaalimaailman mukainen ohjelmointitekniikka tuoda lisää laatua ja ymmärrettävyyttä. Olioparadigman mukaisen ohjelmiston kehittämisessä vastaan tulevia ongelmia voi ratkaista kuten ne on ratkaistu reaalimaailmassa [TeS01].

Periytymisestä seuraavat luokkien väliset *riippuvuudet* monimutkaistavat kuitenkin mm. testausta ja pienestä muutoksesta luokassa voi syntyä esimerkiksi *särkyvän ylikuokan ongelma (fragile base class problem)*. Tämä tarkoittaa, että aliluokan riippuvuus ylikuokastaan voi tehdä siitä epäkelvon, kun sen ylikuokkaa muutetaan. Kun jo proseduraalisissa ohjelmissa pyrittiin mahdollisimman vähiin riippuvuuksiin, rikkoo olioparadigma osittain tätä pyrkimystä [Kos00].

Riippuvuuksia ohjelmiston sisällä syntyy mm. seuraavasti (ks. kuva 8): luokka X on toisen luokan Y ylikuokka ja luokka X on toisen luokan Z aliluokka. Luokka X toteuttaa metodin M, metodi M palauttaa olion luokalle X ja luokka X ymmärtää kutsun M'. Luokan X ilmentymässä on muuttuja V, muuttuja V on luokan V ilmentymä ja V on parametri metodille M. Metodi M on viestin M' nimi ja metodi M2 ylikirjoittaa metodin M1 [WiH92]. Edellä kuvattujen riippuvuuksien lisäksi luokkien välillä on assosiaatio- ja osa/kokoonpanosuhteita.



Kuva 8: Olio-ohjelman riippuvuuksia

Riippuvuuksia on siis paljon verrattuna modulaarisuutta noudattavaan proseduraaliseen ohjelmaan. Koska kutsuhierarkia muodostuu oliokielissä metodien kutsuista, joudutaan tutkimaan pitkä kutsuketju jo yhden toiminnon jäljityksessä. Tästä johtuen *systemin korkean tason ymmärtäminen, mikä on tärkeää mm. ylläpitäjälle, voi tuottaa vaikeuksia* [WiH92].

3.2.2 Käyttötapaukset

Jacobson ja kumppanit toivat [Jac98] oliokeskeiseen ohjelmistokehitykseen *käyttötapaukset (use-cases)* vuonna 1992 kehittäessään *OOSE-menetelmän (Object-oriented Software Engineering)*. Nykyisin käyttötapaukset ovat ensimmäinen vaihe useimmissa oliokeskeisissä menetelmissä ja lähestymistavoissa (esim. *OMT++ (Object-Modelling Technique ++)* ja UML:n tukema RUP). Käyttötapauskavion notaatio on määritelty UML-standardissa.

Käyttötapaukset tuovat ohjelmistokehityksen lähemmäs loppukäyttäjää. Ideana on pitää *käyttötapauskavio (use-case diagram)* mahdollisimman yksinkertaisena, jotta asiakaskin voi osallistua sen tekemiseen. Lisäksi kaaviolla kuvataan nimenomaan loppukäyttäjien ja ohjelmiston välistä kommunikaatiota. Kaavio tuo analyysiin mukaan inhimillisen aspektin

jo sillä, että siinä kuvataan ihminen tikku-ukkona käyttäjäksi ja käyttötapaukset ovat kuvauksia siitä, miten ihminen käyttää ohjelmistoa. Ihmisellä on aina jokin rooli suhteessa ohjelmistoon. Käyttötapausten avulla määritellään koko ohjelmisto tai ainakin tärkeimmät ominaisuudet. Kun käyttötapaukset on löydetty, kuvataan niistä jokainen tarkasti, sovitulla tavalla, esim. tapahtumasekvensseinä, tekstinä tai kaaviona. Käyttötapausten määrittämisen jälkeen etsitään oliot, jotka voisivat toteuttaa määritellyt käyttötapaukset [Kru00].

Nykyisin käyttötapauksia käytetään vaatimusmäärittelyn lisäksi mm. käyttöliittymäsuunnittelun, käyttöohjeiden kirjoittamisen ja testauksen pohjana. Ne muodostavat yhden näkökulman myös arkkitehtuurisuunnitteluun [JBR99]. Käyttötapauksia voidaan joissakin tapauksissa käyttää myös projektisuunnittelun pohjana. Esimerkiksi EVO-vaihejakomallissa voidaan päättää, että ensimmäiseen versioon toteutetaan tietyt käyttötapaukset ja loput vasta versioon kaksi [HaM01].

3.3 MODULAARISUUS

Oliokeskeisyyden perusideana on toteuttaa modulaarisuus, joka on tuotu esille jo ennen olioita. Esimerkiksi Modula2-kieli, joka kehitettiin 1980-luvun alkupuolella, on modulaarinen [Szy98]. Modulaarisuus on yksinkertaisesti ilmaistuna sitä, että ohjelma koostuu itsenäisistä moduuleista, joiden välillä on mahdollisimman vähän *yhteyksiä (coupling)* ja nämä moduulit ovat mahdollisimman *kiinteitä (cohesion)* eli jakautumattomia. Meyer määrittelee [Mey88] viisi kriteeriä, jotka modulaarinen ohjelmisto täyttää. Nämä ovat: *modulaarinen osiinjaettavuus (modular decomposability)*, *modulaarinen koostettavuus (modular composability)*, *modulaarinen ymmärrettävyys (modular understandability)*, *modulaarinen jatkuvuus (modular continuity)* ja *modulaarinen suojaus (modular protection)*.

Meyer esittelee myös *viisi periaatetta*, joita modulaarisuus noudattaa. Nämä ovat:

- *lingvistinen itsenäisistä osista muodostuva alue (linguistic modular area)*,
- *harvat (riittävän monet) rajapinnat (few interfaces)*,
- *mahdollisimman pienet rajapinnat, joista seuraa heikko kytkentä (small interfaces (weak coupling))*,
- *eksplisiittisesti määritellyt rajapinnat (explicit interfaces) ja*
- *tiedon kätkeyttäminen (information hiding)*.

Modulaarisuus toi tullessaan uudenlaisen jaottelun ohjelmistolle. Enää ei ohjelmoijan tarvinnut hallita kokonaan kaikkea, vaan hän pystyi tarkastelemaan yhtä moduulia kerrallaan. Toki kokonaisuuskin täytyi hallita siinä määrin, että sieltä löysi tarvittavan moduulin. Kuten *viisi periaatetta* osoittavat, tulee tuntea myös moduulin tarvitsemat muut moduulit pääpiirteissään eli vähintään niiden määritellyt rajapinnat. Tiedon kätkennästä seuraakin se, että moduulista ei näy ulospäin kuin tarkasti määritellyt asiat.

Modulaarisuus liitetään aina ohjelmiston *laatuun*, koska modulaarisuus tuo mukanaan ymmärrettävyyttä, hallittavuutta ja ylläpidettävyyttä. Modulaarisuuden idea tuleekin ottaa mukaan jo suunnittelussa eikä vasta ohjelmoinnissa [Mey88]. Myös komponenttipohjainen sovelluskehitys pyrkii toteuttamaan edellä mainitut modulaarisuuden piirteet [Szy98].

3.4 KOMPONENTTIPOHJAINEN SOVELLUSKEHITYS

Komponenttipohjaisuuden uskotaan parantavan uudelleenkäytettävyyttä yhä enemmän. Toisaalta se voikin olla mahdollista kehittyvän teknologian myötä, kun saadaan sopivia *suoritusympäristöjä* (*component execution environment, CEE*) ja *väliohjelmistoja* (*middleware*), niin hyvinkin erilaisten komponenttien yhdessä toimiminen mahdollistuu. Uudelleenkäytettävyyden uskotaan helpottavan ohjelmistotoimittajien työtaakkaa. Toisaalta paljon täytyy oppia uuttakin, kuten erilaiset uudet tekniikat ja uudenlainen työnjako. Pitää myös oppia luottamaan toisen tekemiin komponentteihin eikä yrittää aina tehdä kaikkea itse [Szy98].

3.4.1 Itsenäiset komponentit

Komponenttipohjainen sovelluskehitys rakentuu ajatukselle *itsenäisinä* toimitettavissa oleville "ohjelmiston osille", joilla on sopimuksiin pohjautuvia *rajapintoja*, ja joiden välillä on vain eksplisiittisesti *määriteltyjä riippuvuuksia* [Szy98, s.34]. Komponentti, tai tarkemmin liiketoimintakomponentti, sisältää toimivan koodinsa lisäksi tarkan rajapintamäärittelyn sekä dokumentit käyttötarkoituksestaan ja ohjeet käyttöönotosta.

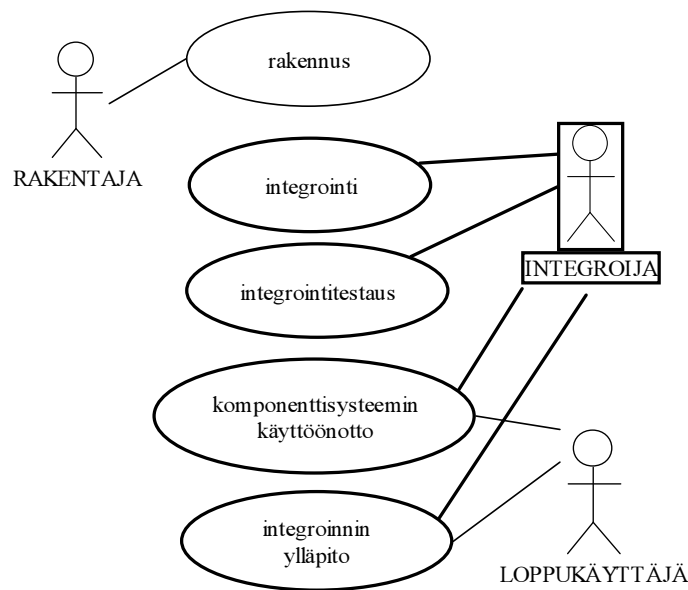
Komponenttipohjainen ohjelmointi noudattaa yhä paremmin modulaarisuuden määrittelyä. Szyperski [Szy98] on yhtä mieltä Meyerin [Mey88] kanssa määritellesään modulaarisuutta: "Linguistic modular area, few interfaces, small interfaces (weak coupling), explicit interfaces ja information hiding" -periaatteiden tulee toteutua myös komponenttipohjaisessa sovelluskehityksessä. Hyvin määritellyillä rajapinnoilla on siis

tärkeä osuus. Komponentin tulee tarjota riittävä ja hyvin määritelty rajapinta ja kätkeä epäolennainen tieto käytön kannalta.

Esimerkki *liiketoimintakomponentista*, joka voidaan toimittaa itsenäisenä voisi olla kalenteri-komponentti, joka ostetaan osaksi *liiketoimintakomponenttisysteemiä*. Sen hyvin määritellyt ja dokumentoidut rajapinnat auttavat sen käyttämisessä ja yhteensovittamisessa muiden systeemin komponenttien kanssa. Kalenteri-komponentin kanssa yhdessä voisi toimia jo aikaisemmin hankitut asiakkaan tietoja käsittelevä asiakas-komponentti ja tilauksia käsittelevä tilaus-komponentti.

3.4.2 Komponenttituotannon toimijat

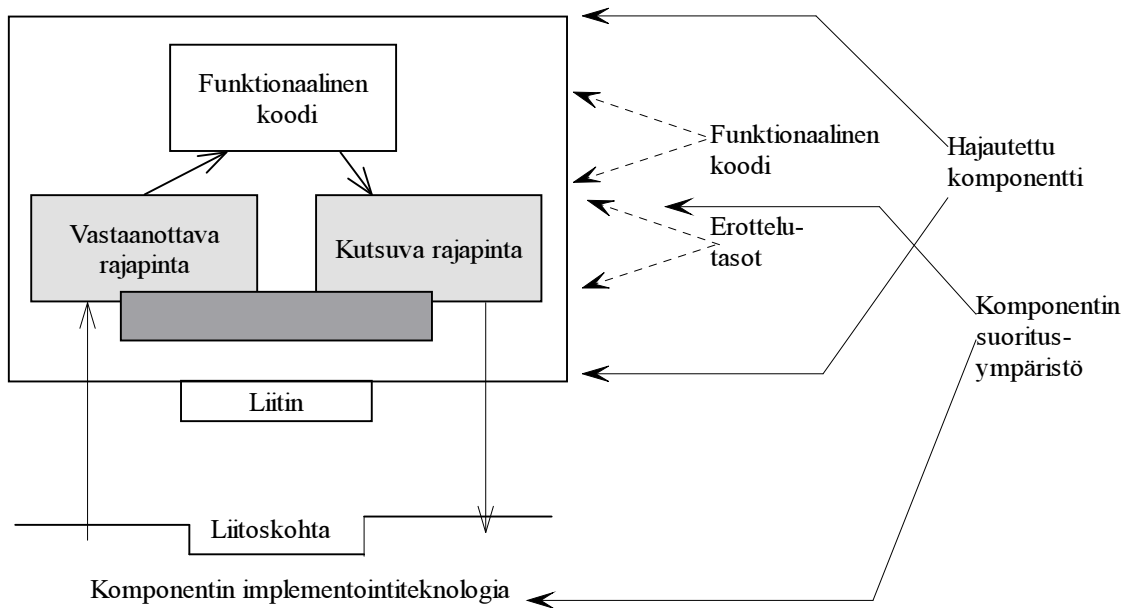
Komponentit tuovat uudenlaisen roolin suhteessa ohjelmistotuotantoprosessiin. Komponentin *rakentaja* voi olla ohjelmiston määrittelijä, suunnittelija, ohjelmoija ja testaaja kuten ennenkin. Uusi rooli syntyy, kun mukaan tulee *komponenttien integroija* (kuva 9), joka tarvitaan rakentajan ja *loppukäyttäjän* väliin. Integroija voi ostaa, vuokrata, toteuttaa itse ja räätälöidä komponentteja. Integroinnin tarkoitus on saada aikaan toimiva komponenttisysteemi. Integroijan tulee huolehtia siitä, että ostetut tai itse tehdyt komponentit tekevät, mitä on pyydetty. Lisäksi komponenttien tulee toimia integroituna yhdessä. Tämä vaatii integroijalta testaajan ominaisuuksia sekä paljon teknistä tietämystä, koska eri komponentit voidaan toteuttaa hyvin erilaisilla tekniikoilla ja kielillä. Integroijana voikin toimia esimerkiksi organisaation atk-osasto, joka tekee myös komponenttihankinnat.



Kuva 9: Komponenttituotannon tuoma lisänäkökulma ohjelmistotuotantoprosessin toimijoihin ja aliprosesseihin

Kun ajatellaan komponentteja sovelluskehityksen kannalta, voi komponentin rakentaja tehdä työtään hyvinkin itsenäisesti, jos kyseessä on komponentti, joka tulee esimerkiksi komponenttikirjastoon myytäväksi. Tosin, jos rakentaja ei tee minkäänlaista yhteistyötä toisten komponenttien kehittäjien tai mahdollisten *ostajien* ja integroijien kanssa, komponentti ei luultavasti saavuta markkinoita. Jos taas on kyse komponentista, joka tulee osaksi komponenttisysteemiä, on yhteistyö suorastaan pakollista.

Toisaalta komponenttitekniologia mahdollistaa myös funktionaaliseksi koodiksi kutsutun osan eli toiminnallisuuden tekemisen komponenttiin ilman *rajapintamäärittelyihin* (*interface* (=vastaanottava rajapinta) ja *proxy* (=kutsuva rajapinta)) tarvittavaa tietämystä. Tämä mahdollistaa erikoistumisen. Kaikkien ei tarvitse osata kaikkea. Kuvassa 10 havainnollistetaan komponentin mahdollista jakoa.



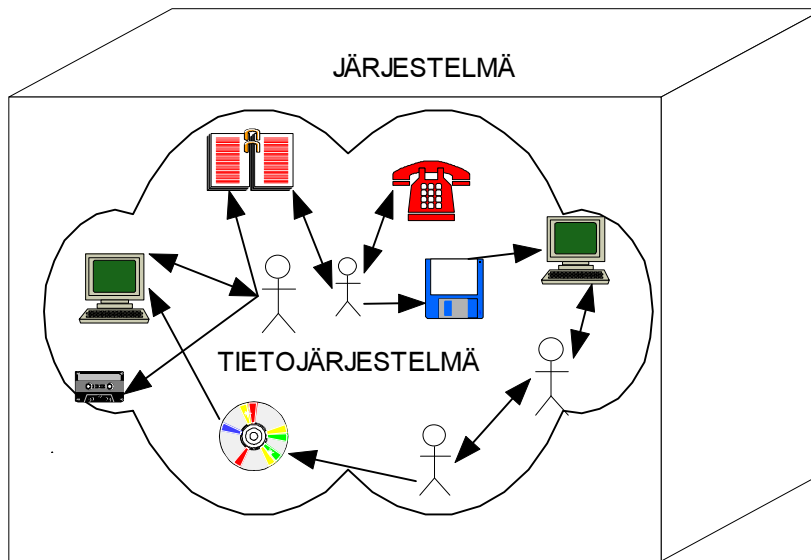
Kuva 10: Hajautetun komponentin erilaiset näkymät toteuttajien kannalta [HeS00]

Myös rakentajien kesken voi syntyä uudenlaisia rooleja. Toiminnallisen *koodin kirjoittajan* ei välttämättä tarvitse oppia paljoakaan uutta. Muut osat komponenteissa ovat uutta teknologiaa. Projektissa tehokkuus voi parantua, kun jokainen voi keskittyä vain yhteen asiaan. Yksi voi siis keskittyä funktionaaliseen koodiin, toinen voi tehdä *vastaanottavaa rajapintaa*, kolmas *kutsuvaa rajapintaa*, neljäs hallitsee *liittimeen ja liitoskohtaan liittyvän tekniikan* ja viides jäsen on *komponentin suoritust ympäristön asiantuntija*.

Määrittelijöiden tulee puolestaan opetella ajattelemaan komponenttikeskeisesti sen lisäksi, että he osaavat jo kenties komponenttien sisällä käytettävät menetelmät ja mallinnustavat kuten esimerkiksi oliokeskeisyys, luokkakaaviot ja peräkkäiskaaviot. Määrittelijöiltä vaaditaan yhä parempaa kokonaisuuksien hallintaa. Tosin heilläkin on mahdollisuus keskittyä kokonaisuuden hahmottamisen jälkeen yhteen komponenttiin. Tällöin toteutuu edellä mainittu periaate ”*yhden ihmisen kokoisista komponenteista*”. Tässä on huomioitava, että sama henkilö voi olla sekä määrittelijä että rakentaja.

3.5 TIETOJÄRJESTELMÄNÄKÖKULMA

Tietojärjestelmä on laajempi kokonaisuus kuin ohjelmisto. Ohjelmisto voi toimia osana automatisoitua tietojärjestelmää. Kuvassa 11 selvitetään tietojärjestelmä-käsitettä.



Kuva 11: Tietojärjestelmä on kokonaisuus, joka koostuu ihmisistä, laitteista ja ohjelmistoista, jotka vuorovaikutuksessa toistensa kanssa huolehtivat järjestelmän tarvitseman tiedon virroista, tallentamisesta, löytämisestä ja suojaamisesta.

Kuvassa 11 *laatikko* kuvaa *järjestelmää* (*system*), joka on olemassa toteuttaakseen omaa ”liikeideaansa”, esimerkiksi sairaala, koulu tai myymälä. Laatikon sisällä oleva pilvi kuvaa järjestelmän tarvitsemaa *tietojärjestelmää* (*information system*), jonka sisällä on kuvattuna tyypillisiä tietojärjestelmän osia; ihmisiä, tietokoneita, puhelin, kalenteri, disketti, cd-rom, kasetti. Kaikkien näiden tehtävä on huolehtia siitä, että tarvittu tieto on oikeaan aikaan oikeassa paikassa, jotta esimerkiksi päätöksenteko ja erilaiset toimenpiteet mahdollistuvat.

Myös ohjelmistoja voidaan kehittää tietojärjestelmänäkökulmasta, jolloin tarvitaan vain laajempaa kokonaiskuvaa. Tietojärjestelmien kehittäminen on prosessi, jolla helpotetaan kohteena olevaa työtoimintakokonaisuutta uuden teknologian avulla analyysin, suunnittelun, toteutuksen, käyttöönoton ja ylläpidon kautta. Kun tietojärjestelmää kehitetään täytyy nähdä olemassa oleva kokonaisuus ja sen kehittämismahdollisuudet aina arkkitehtuuria, tietoliikenne ratkaisuja, komponenttikokonaisuuksia, ohjelmistoja ja jopa toteutusluokkia myöten unohtamatta ihmistä ja liiketoimintaa [Haw01].

Tietojärjestelmien kehittämisessä voidaan nähdä suuntauksina esimerkiksi *käyttäjakeskeinen suunnittelu* (*user-centered design*), *osallistava suunnittelu* (*participatory design*) ja *toiminnan analyysi* (*activity analysis*). Nämä menetelmät ottavat ihmisen

paremmin huomioon kuin pelkän ohjelmiston suunnitteluun ja toteutukseen keskittyvät menetelmät. Toisaalta esimerkiksi jo aiemmin mainittua ISAC-menetelmää voidaan pitää tietojärjestelmän kehittämisen välineenä, koska se on *ihmislähtöinen (human-oriented)* menetelmä [AvF95]. Myös liiketoimintaprosessien ymmärtämiseen ja asiakasvaatimusten keräämiseen keskittyviä menetelmiä voidaan pitää tietojärjestelmien kehittämisen välineinä [Haw01].

Käyttäjakeskeinen suunnittelu tarkoittaa sitä, että analyysoija menee observoimaan eli tarkkailemaan tulevaa käyttäjää, jotta kehitettävä tietojärjestelmä tulee tukemaan hänen toimiaan. Tässä voidaan käyttää apuna esimerkiksi videointia. Esimerkiksi asiantuntijajärjestelmät tehdään käyttäjakeskeisellä lähestymistavalla. *Osallistavassa suunnittelussa* käyttäjä ja suunnittelija asettuvat samalle tasolle. Suunnittelija voi esimerkiksi mennä kohteena olevan työryhmän jäseneksi ymmärtääkseen työtoiminnan kokonaisuutta. Tällöin hän voi paremmin huomioida myös manuaaliset toiminnot eikä vain ohjelmiston käyttöä [Haw01]. *Toiminnan analyysissä* toimintakokonaisuuden elementeistä selvitetään:

- toimintatapa (eli onko se esimerkiksi tekniikkakeskeinen),
- kollektiivinen toimija (tiimi vai jakautunut),
- yhteistoiminnan ja kommunikoinnin välineet (esimerkiksi työn jaottelu ja roolit),
- toimijat ja tekijät,
- työvälineet ja -keinot (kuten koulutus), laitteistot ja työn merkitykset sekä
- työprosessit [KSO00].

Tietojärjestelmänäkökulman valitseminen ei sulje pois ohjelmistotuotannon menetelmiä. Esimerkiksi käyttäjakeskeinen suunnittelu täydentyy hyvin protoilun ja käyttötapauskaavioiden käytön kautta, kun taas pelkkään ohjelmistoon keskittyessä, voi ihminen unohtua. Kun muistetaan, että ohjelmistojä tekivät *ihmiset ihmisille*, tulee huomioitua sekä niiden tekijät että tulevat käyttäjät.

4 TEHOKKUUS JA TUOTTAVUUS

Työn tehokkuuden parantamiseksi on kehitelty *johtamis-* ja *organisaatioteorioita*. Näyttää siltä, että tehokkuuden ollessa kyseessä olisi *käyttäytymisteorioilla* kuitenkin suurempi merkitys johtamis- ja organisaatioteorioiden tulosten vaikuttaessa taustalla [Juu92]. Käyttäytymisteoriat kiinnittävät huomiota *yksilöön* ja yksilön asenteisiin työtä kohtaan johtamis- ja organisaatiostrategioiden sijaan.

Myös *työryhmän* vaikutus yksilön suoritustasoon on suuri. Ryhmä voi olla kannustava mutta myös rajoittava tekijä. Ryhmän suoritustasoa ei helposti haluta alittaa, mutta ei myöskään ylittää. Ryhmästä ei haluta erottua, mistä seuraa tasapäistyminen. Jos joku erottuu joukosta, on hänen kestettävä myös mahdollinen syrjintä ja hylkäämisen tunne. [Juu92]

4.1 ORGANISAATIO

"Organisaatio on sosiaalinen järjestelmä, jolla on tietyt tavoitteet ja sen jäsenten tiedostama, tarkoituksellisesti luotu rakenne." Organisaatiossa on siis enemmän kuin yksi ihminen ja sen ominaisuuksia ovat jatkuvuus ja vakaisuus toiminnassa [VLK98]. Organisaation rakenne muodostuu:

- tehtävärakenteesta, joka kertoo työnjaon,
- raportointi- ja ohjaussuhteista, jotka luovat organisaation jäsenten, ryhmien ja esimiesten välille hierarkisen rakenteen sekä
- koordinoitijärjestelmistä eli informaatio- ja ohjaustoiminnoista.

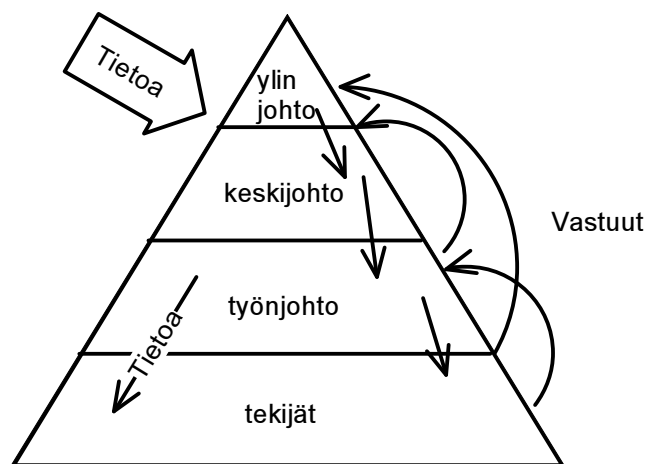
Tämän rakenteen sisällä vaikuttavat yksiköt ovat usein kiinteitä ja kommunikoivat enemmän yksikön sisällä kuin keskenään. Yksiköt voivat sijaita esimerkiksi konkreettisesti eri rakennuksissa.

Organisaation muodostaa konkreettisen puolen lisäksi siinä *toimivat ja ajattelevat ihmiset*. Tietotyössä jokaisen yksilön on ajateltava. Ajattelevat yksilöt voivat muodostaa tietyllä tavalla toimivia ja ajattelevia ryhmiä. Tämän vuoksi on huomioitava yksilö- ja ryhmäkäyttäytyminen, kun halutaan tarkastella organisaatiota.

Organisaation tehokkuuteen liittyy oikeanlaisen rakennetyypin valinta sekä johtamisote ja ohjaustyylit. Rakennetyypin valintaan liittyy esimerkiksi yrityksen koko, liiketoiminta, asiakasryhmät ja tuotantomenetelmät [VLK98]. Organisaatorakenteista valittavana on mm.: linja-organisaatio, toimintokohtainen organisaatio, tulosyksikköorganisaatio, projektiorganisaatio, matriisiorganisaatio tai verknomainen organisaatio [Juu92].

4.1.1 Organisaatioarkkitehtuuri

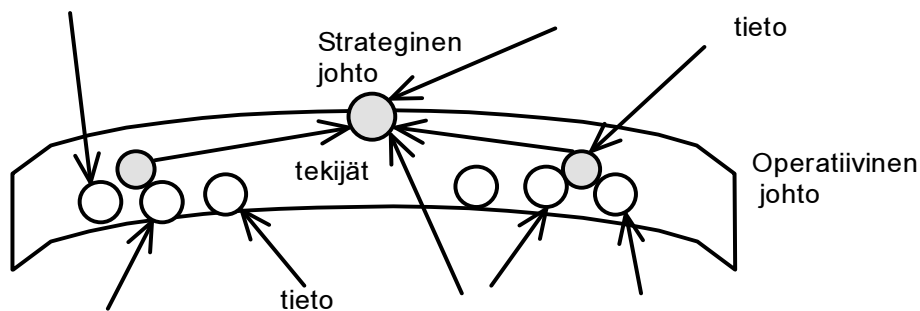
Organisaatio nähdään perinteisesti *hierakkisena* kokonaisuutena, joka muodostuu erilaisista byrokraattisista vastuu- ja tiedotuskanavista (ks. kuva 12).



Kuva 12: Perinteinen organisaatiohierarkia

Samanaikaisesti, kun yritykset 1990-luvulla alkoivat poistaa turhia tasoja organisaatioistaan, tuli hierarkisen näkökulman rinnalle *prosessiajattelu*, ns. horisontaali organisaatio. Tämä tarkoittaa sitä, että yrityksen ydinprosessit muodostavat ketjuja, joissa esimerkiksi toimittajalta tilatusta ja saadusta tavarasta suunnitellaan, valmistetaan ja markkinoidaan asiakkaalle sopiva tuote tai kehitetään yrityksen suorituskykyä ja laatua läpi organisaation tutkimus-, valmistus- ja markkinointiyksikön. Tällaiset prosessit läpäisevät organisaation. Ketjujen läpiviemiseksi voidaan muodostaa *tiimejä*, jolloin tiimitkin toimivat yli organisaatorajojen. Yksilöitä ei lokeroida enää niin tarkasti organisaatiohierarkiaan. [VLK98]

Ydinprosesseihin keskityttäessä mukaan otetaan prosessit, jotka lähtevät asiakkaan tarpeista ja päättyvät asiakkaan tarpeen tyydyttämiseen, ja jotka tuottavat asiakkaalle *lisäarvoa*. Tällä ajattelutavalla karsitaan turhat prosessit, jotka eivät tuota ja voidaan keskittyä tärkeisiin prosesseihin. Voidaan puhua myös *asiakslähtöisistä ja nopealiikkeisistä* verkosto-organisaatioista [Han94]. *Perinteinen organisaatiohierarkia (=pipomalli)* voi toimia, jos olosuhteet ovat kohtalaisen muuttumattomat, mutta tänä päivänä toimii paremmin *matalampi organisaatio (=baskerimalli)*. Baskerimallin (ks. kuva 13) mukainen organisaatio edellyttää vastuun jakamista ja valtuuttamista sekä tiimiorganisaatiota. (Tiimeihin palataan kappaleessa 5.2 Tiimit.) Voidaan sanoa, että baskerimallissa koko organisaatio on kuin ”hikinauhassa” [Han94].



Kuva 13: Organisaation baskerimalli [Han94]

Prosessimallin käytöstä on saatavissa kustannushyötyjä. Sen lisäksi, että turhat prosessit karsiutuvat, säästetään myös hallintokustannuksissa, koska vastuulliset itseohjautuvat tiimit vähentävät hierarkian tarvetta. Kun yritys siirtyy prosessiajatteluun, täytyy yrityksen ydinprosessit löytää ja mallintaa. Tähän voi kulua aikaa ja rahaa, mutta kun uusi ajattelumalli on saatu toimimaan, alkavat kustannushyödyt yrityksessä näkymään säästöinä. Kun huomio kiinnitetään ydinosaamiseen ja olennaisiin asioihin varmistetaan organisaation osaaminen ja *reagointikyky* muuttuvassa ympäristössä eli ollaan aina nopeasti myös ”markkinoilla”. Lisäksi ydinprosesseja voidaan käyttää tietojärjestelmän suunnittelun pohjana [Han94].

4.1.2 Epäviralliset organisaatiot

Organisaatioiden sisällä muodostuu myös ns. epävirallisia organisaatioita. Henkilöllä, ei pelkästään esimiehellä, voi olla organisaatiossa *vaikutusvaltaa*. Jos henkilö voi kontrolloida toisen saamia tietoja, tehtäviä, palkkioita tai rangaistuksia, on hänellä valtaa suhteessa tähän toiseen henkilöön. Valta voi perustua esimerkiksi asiantuntemukseen, vuorovaikutukseen, suhteisiin tai tietoon. Rangaistusvalta voi ilmetä esimerkiksi epämiellyttävien tehtävien antamisena toiselle, kun taas palkitsemisvalta voi olla esimerkiksi arvostuksen ja etuisuuksien antamista. Valtarakenteet voivat muodostua esimerkiksi sen mukaan, kuinka pitkään on ollut työntekijänä, millainen on koulutustausta tai millainen on suhde esimieheen. [Juu92]

Yrityskulttuuriin kuuluvat ”näkymättömät” normit voivat heijastua epävirallisten organisaatioiden syntymisenä. Jos oikeat esimiehet eivät ole riittävästi läsnä, ei heitä välttämättä koeta esimiehinä. Esimiehen asemaan voi nousta esimerkiksi asiantuntijaksi luokiteltu tai pidempään yrityksessä ollut henkilö. Työntekijöiden luonteenpiirteet ja keskinäiset suhteet voivat olla epävirallisen organisaation syntymisen taustalla. Vahvat persoonat saattavat ”nousta valtaan”. Henkilökemioilla voi myös olla vaikutusta, jos alainen ei tule toimeen esimiehensä kanssa, hän voi etsiä ”korvaavan esimiehen” itselleen.

4.2 JOHTAJUUS

Johtamiseen kuuluu tärkeänä osana *henkilöstön johtaminen* organisaation ja rahavirtojen johtamisen rinnalla. Tämä on otettava huomioon, vaikka edustettaisiin "tavoitejohtamista, tilannejohtamista, muutosjohtamista, prosessijohtamista, arvojohtamista, tiimijohtamista, toimintojohtamista, resurssijohtamista, tulosjohtamista, merkityksen johtamista tai laatujohtamista" [Mau96]. Johtamiselle riittää siis nimityksiä. Osittain johtamisote on valittava organisaatiotyypin mukaan, mutta myös alaisten odotukset ja taitotasot voivat vaikuttaa erilaisten johtamisotteiden tarpeeseen. Johtajat ovat varmasti opiskelleet erilaisia oppeja koulutuksensa aikana, mutta silti he eivät aina tiedä, mitä heidän alaisensa odottavat heiltä.

Asiakkaat ovat tietenkin tärkeitä suunnannäyttäjiä johtajien toimille, koska heille tuotteet myydään. Johtajilla on kuitenkin myös *sisäisiä asiakkaita (=alaisia)*, jotka odottavat johtajiltaan *johtamispalvelua*. Hyvä johtamispalvelu huolehtii, että sisäiset asiakkaat saavat työltään sen, mitä odottavat. Johtajan tehtävänä on auttaa yksilöitä ja työryhmiä

saavuttamaan tavoitteensa sekä sitoa heidän tavoitteensa organisaation tavoitteisiin [Juu92]. Työltä odotetaan mm. työn jatkuvuutta, oikeudenmukaista kohtelua, miellyttäviä työtovereita, mielenkiintoista työtä, hyviä työolosuhteita ja kykyjä vastaavaa työtä (Pauli Juutin tutkimukset vuosina 1986 ja 1989 [Mau96]). Yksi tapa saada tietää alaisten odotuksia on *kehityskeskustelut*, jos ne käydään hyvässä hengessä.

Johtamisote ja ohjaustyyli on valittava tilanteen vaatimusten mukaan. Vaikka johtamisote olisi hyvin määritelty ja tarkka, se ei epävakaisissa olosuhteissa ole välttämättä riittävän hyvä joustamattomuutensa takia [VLK98]. Johtamistyyleistä voidaan erotella esimerkiksi *autoritaarinen-*, *demokraattinen-* ja ”*antaa mennä*” –johtamistyyli [Juu92]. Autoritaarinen johtamistapa on määräilevä. Alaisten sanomiset eivät paina päätöksentekotilanteissa. Demokraattinen johtaja sen sijaan ”laskeutuu” kuuntelemaan alaisiaan. ”Antaa mennä” –johtaja ei välitä asioiden kulusta ja antaa alaisilleen hyvinkin paljon päätäntävaltaa. Tämän johtamistyylin omaksunut johtaja voi olla tehnyt valintansa tietoisestikin ja olla hyvinkin kiinnostunut asioiden kulusta, mutta ei näytä sitä.

Huomionarvoista johtamistyylin valinnassa on myös, että autoritaarinen johtamistyyli opettaa alaiset avuttomuuteen. Jos käskyt on aina annettu ylhäältä käsin, ei opita ajattelemaan eikä pystytä itsenäisiin ratkaisuihin. Toimintakykyisyyttä syntyy, kun johtajuutta jaetaan. Usein voidaan ajatella, että paras johtamistapa on tietenkin demokraattinen, mutta asia ei ole näin suoraviivainen. Useimmiten tullaan lopputulokseen, joka puhuu *tilanteen mukaan johtamisen* puolesta. Joustavuus lienee ”päivän sana”. Johtajan tulisi olla mahdollistaja, joka takaa tasapuoliset työskentelymahdollisuudet kaikille työntekijöilleen.

Mitkä ovat sitten tekijöitä, jotka voivat tehdä tilanteista erilaisia? Tällaisia tekijöitä ovat esimerkiksi: esimiehen asema koko organisaatiossa, esimiehen omat taidot; pystyykö hän neuvomaan alaisiaan sekä alaisten taidot; ovatko he asiantuntijoita vai noviiseja. Lisäksi tulisi huomioida työtilanne; onko kyseessä todella tärkeä projekti, jolla on kiire tai onko nyt aikaa opetella ja opettaa sekä tehdä kenties vähän virheitäkin. Esimiehen taituruutta on huomioida myös alaisen henkilökohtainen tilanne. Toisaalta myös alainen voi huomioida esimiehensä kiireitä ja paineita. Jokainen ei aina ole parhaassa mahdollisessa työntekokunnossa johtuen tekijöistä, joille ei mahda itse mitään [HeB90]. Esimerkiksi läheisen vakava sairaus voi olla tällainen tekijä.

Selvää on, että asiantuntijoille täytyy antaa ”vapaammat kädet” ja heidän mielipiteensä täytyy ottaa huomioon. Kaikkia täytyy tietenkin kuunnella, mutta eniten asiasta tietävän sana painaa eniten. Kuitenkin kuka tahansa voi esittää sen ”parhaan idean”. [Hum98]

4.2.1 Tilannejohtaminen

Paul Hersey ja Kenneth Blanchard julkaisivat vuonna 1988 *The Management of Organisational Behavior* -teoksen, jossa he esittelevät tilannejohtamisen periaatteet. Kaksi tärkeintä periaatetta ovat: ensimmäiseksi johtamistapa tulee valita alaisen valmiuksien mukaan niin, että tämä pystyy suoriutumaan tehtävistään ja toiseksi johtamistavan tulee johdattaa alaista yhä itsenäisempään työskentelyyn [HeB90].

Tilannejohtaminen on malli, joka perustuu (1) *tehtäväkeskeisyyteen* eli esimieheltä tulevan ohjauksen ja ohjeiden määrään, (2) *ihmiskeskeisyyteen* eli esimiehen aikaansaamaan sosioemotionaalisen tuen määrään ja (3) alaisten valmiustasoon heidän suorittaessaan toimintoja ja pyrkiessään tavoitteisiinsa sekä (4) näiden kolmen tekijän vuorovaikutukseen [HeB90]. Taulukosta 1 nähdään yksinkertaistettuna, kuinka johtamistavan tulee muuttua alaisten tietojen ja taitojen tason mukaan.

Taulukko 1. Esimiehen sopiva rooli suhteessa alaisen valmiuksiin mukailtuna lähteestä [HeB90]

Itseohjautuva Kykenevä / hyvä itsetunto	<i>Konfliktin todennäköisyys lisääntyy</i>	←←←←←	←←←←←	T4=Alhainen tehtäväkeskeisyys alhainen ihmiskeskeisyys
Sitoutunut Kykenevä/ epävarma	↑ ↑		T3=Alhainen tehtäväkeskeisyys korkea ihmiskeskeisyys	↓ ↓
Kiinnostunut Kykenemätön/ hyvä itsetunto	↑ ↑	T2=Korkea tehtäväkeskeisyys korkea ihmiskeskeisyys		↓ ↓
Riippuva Kykenemätön/ epävarma	T1=Korkea tehtäväkeskeisyys alhainen ihmiskeskeisyys	→→→→→	→→→→→	<i>Epävarmuus & stressi lisääntyy</i>
Alainen Esimies	Auktoriteetti Ohjaava Neuvova	Motivoiva Myövä Selkyttävä	Osallistuva Avustaja Rohkaiseva	Delegoiva Havainnoiva Tarkkaileva

Ohjaavassa roolissa esimiehen tulee antaa tarkkoja, yksityiskohtaisia ohjeita ja valvoa ohjeiden noudattamista, koska alainen ei vielä tiedä eikä osaa. Riippuva alainen tarvitsee ensisijaisesti apua tehtävässä. Jos taas alainen on jo kiinnostunut asioista, muttei vielä oikein osaa, on esimiehen oltava tukena tehtävässä ja samanaikaisesti annettava mahdollisuus kysymyksille, jolloin hänen roolinsa on *myyvä*. Alaisen ollessa Riippuva tai Kiinnostunut tarvitaan aina esimiehen ohjeita ja päätöksiä. Kun taas Sitoutunut tai Itseohjautuva alainen on kykenevä tekemään päätöksiä myös itsenäisesti. *Osallistuvaa* esimiestä tarvitaan, kun alaisella on taitoja, mutta hän suhtautuu töihinsä hieman epävarmasti tai haluttomasti. Työnsä jo hallitsevat ja itseensä luottavat, motivoituneet alaiset eivät kaipaa jatkuvaa tukea tai ohjeita, jolloin esimiehen on parempi pysytelläkin *delegoivassa* roolissa, jakaen vastuullisia tehtäviä. [HeB90]

Jos johtamistyyli ei muutu alaisen valmiuksien mukaan kuten taulukossa 1 (T1, T2, T3, T4) on esitelty, voi seurauksena olla pahojakin konflikteja tai epävarmuudesta seuraavaa stressiä. Jos asiantuntija saa tarkkoja ohjeita ja ohjausta jatkuvasti tai hänen mielipiteitään ei oteta huomioon, voi hänestä tuntua, että hänen osaamistaan ei arvosteta tai että häneen ei luoteta. Itseohjautuvasta alaisesta voi tulla jopa riippuvainen, jos häntä koko ajan neuvotaan. Ohjeista riippuvainen alainen taas ajautuu kriisiin, jos häneltä odotetaan enemmän itsenäisyyttä kuin, mihin hän pystyy. [HeB90]

Tilannejohtaminen vaatii esimieheltä kykyä tunnistaa, ymmärtää ja arvioida alaistensa eroja. Lisäksi hänen tulee pystyä mukauttamaan toimintaansa kulloisenkin tilanteen edellyttämällä tavalla. Jokainen alainen saa näin tarvitsemansa määrän ohjausta (=tehtäväkeskeisyys) ja tukea (=ihmiskeskeisyys), jolloin alaisista on hyötyä yritykselle, heistä tulee enemmän itseohjautuvia ja tuloksia syntyy [HeB90]. Toisaalta esimies voi määrittellä myös projektiryhmästä tai tiimistä yksilölle sopivan ohjaajan, selkiyttäjän, rohkaisijan tai tarkkailijan, jos hän ei itse tunne kykenevänsä olemaan jatkuvasti läsnä tilanteen edellyttämällä tasolla.

4.2.2 Erilaiset esimiehet

Johtajille asetetaan paljon vaatimuksia, vaikka ihmisiähän hekin ovat ja siksi erilaisia keskenään. Maunula on eritellyt johtajat laadunvalvojiin, tasapainottajiin, toimeenpanijoihin ja järkeilijöihin [Mau96]. Oman johtamistyyliensä hyvyiden tai huonouden saa tietää alaisilta saadusta palautteesta. *Laadunvalvojat* vaativat tarkkuutta,

tasapainottelijat levittävät ympärilleen myönteistä ilmapiiriä, *toimeenpanijat* laittavat asiat vauhdilla eteenpäin ja *järkeilijät* harkitsevat asioita huolella. On selvää, että äärimmilleen vietyinä, mikä tahansa näistä tyyleistä on huono eikä mikään tyyli yksin ole hyvä. Tasapainottelija on alaisten mielestä mukava, mutta hommat voivat jäädä tekemättä. Toimeenpanija voi tehdä asioita pohtimatta erilaisten ratkaisujen vaikutuksia kun taas järkeilijä voi pohtia niitä loputtomiin. Laadunvalvoja voi olla jopa jarruttaja jatkuvien tarkastustensa kanssa. Hyvä esimies tuntee itsensä ja osaa hyödyntää omia hyviä puoliaan. Alaiset tuntevat esimiehensä hyvin, joten heiltä voi kysyä. Johtajalta vaaditaan toki hyvää itseluottamusta, jotta hän voi ottaa alaistensa todellista palautetta vastaan ja oppia siitä. [Mau96]

Hieman erilaisia taitoja tarvitaan myös organisaation eri tasoilla. Jos ajatelleen käsitteellisten, teknisten ja ihmissuhteisiin liittyvien taitojen tarpeellisuutta, niin ylin johto tarvitsee eniten käsitteellisiä taitoja, kuten taloushallinto ja liiketoiminnan johtaminen, kun keskijohto ja työnjohto tarvitsevat enemmän teknisiä taitoja. Alin taso eli työnjohto tarvitsee eniten teknisiä taitoja. Joka tasolla tarvitaan kuitenkin eniten ihmissuhteisiin liittyviä taitoja. [HeB90]

4.2.3 Avoin kommunikaatio

Organisaation rakenne muodostuu mm. erilaisista tiedotus- ja vastuukanavista. Esimiehiä voi olla organisaatioissa useilla eri tasoilla. Kaikilla tasoilla on tärkeää huolehtia tiedon jakamisesta. Tietoa voidaan pitää myös vallan välineenä [Juu92]. Humphrey pitää [Hum98] avointa kommunikointia tärkeänä. Jos alaisista tuntuu, että heiltä salataan jotain, se vähentää ainakin *turvallisuuden tunnetta*. Jos halutaan sitoa ihmisten omat tavoitteet organisaation tavoitteisiin, tulee ihmisille kertoa riittävästi ylemmän tason tavoitteista tai organisaation tavoitteita voidaan suunnata ihmisten tavoitteiden mukaan.

Ei riitä, että alaisia tiedotetaan, vaan tavoitteet ja suunnitelmat tulee jakaa heidän kanssaan tai mieluiten jopa tehdä yhdessä. Ainakin päätökset tulee *perustella*. Vain näin alaiset voivat löytää yrityksen tavoitteista omat persoonalliset tavoitteensa. Alaisten ideoita ja näkemyksiä tavoitteiden saavuttamiseksi tulee myös kunnioittaa. "Palkinnoksi jaetusta tiedosta ja alaisten kunnioittamisesta johtaja saa tuottoisat, lojaalit ja yhtenäiset tiimit." [Hum98]

Oikeanlainen kommunikaatio on kaksisuuntaista vuorovaikutusta. On jopa vaarallista jakaa vaikeaselkoista tai laaja-alaista tietoa varmistamatta, että se ymmärrettiin oikein. Väärinymmärretty tieto voi johtaa väärille raiteille ja aiheuttaa yleistä sekaannusta. Varsinkin johtotasolta saatua sekavaa tietoa on vaikea korjata jälkeenpäin. Vaikka asia selvitettäisiin uudelleen, voi olla, että enää ei tiedetä mihin uskoa. [Hum98]

Johtajan tai projektipäällikön ei tule kuitenkaan jakaa kaikkea tietoa alaisensa kanssa. Johtajan on kannettava iso osa epävarmuudesta yksin. Osa tiedosta voi olla esimerkiksi hallinnollista tai se ei kosketa tiimin jäseniä. Voi olla, että tiimin jäsenillä ei ole riittävästi taitoja tai tietoa edes keskustella kaikista asioista. Jos heitä pyytää tekemään päätöksiä tällaisista asioista, he menevät turhaan hämilleen. [Hum98]

On myös reilua kertoa, jos aiotaan keskustella asiasta, josta päätös on jo tehty. Yleisesti ottaen ennen päätöksen tekoa tulee keskustella asiasta niin hyvin ja riittävästi, että päätöksen teon jälkeen ei asiasta enää tarvitse keskustella. Avoin kommunikaatio johtotasolta näyttää johtavan avoimempaan kommunikaatioon myös ryhmän sisällä. Vähäinen tiedonvälitys johtaa vain turhiin huoliin "salatuista asioista" ja lisää juoruilua. [Hum98]

4.2.4 Kehityskeskustelut

On selvää että, johtajan tulee saada henkilöstönsä voimavarat tehokkaasti yrityksen käyttöön. Hänen tulee osata mm. tilannejohtamista eli hänen tulee tuntea alaisensa hyvät ja huonot puolet sekä motivaatiotekijät. Helpoiten nämä asiat selviävät keskustelemalla alaisten kanssa säännöllisin väliajoin, jolloin kyse on *kehityskeskusteluista* eli alaisen ja esimiehen välisistä kaksisuuntaisista luottamuksellisista vuorovaikutustilanteista [Val00].

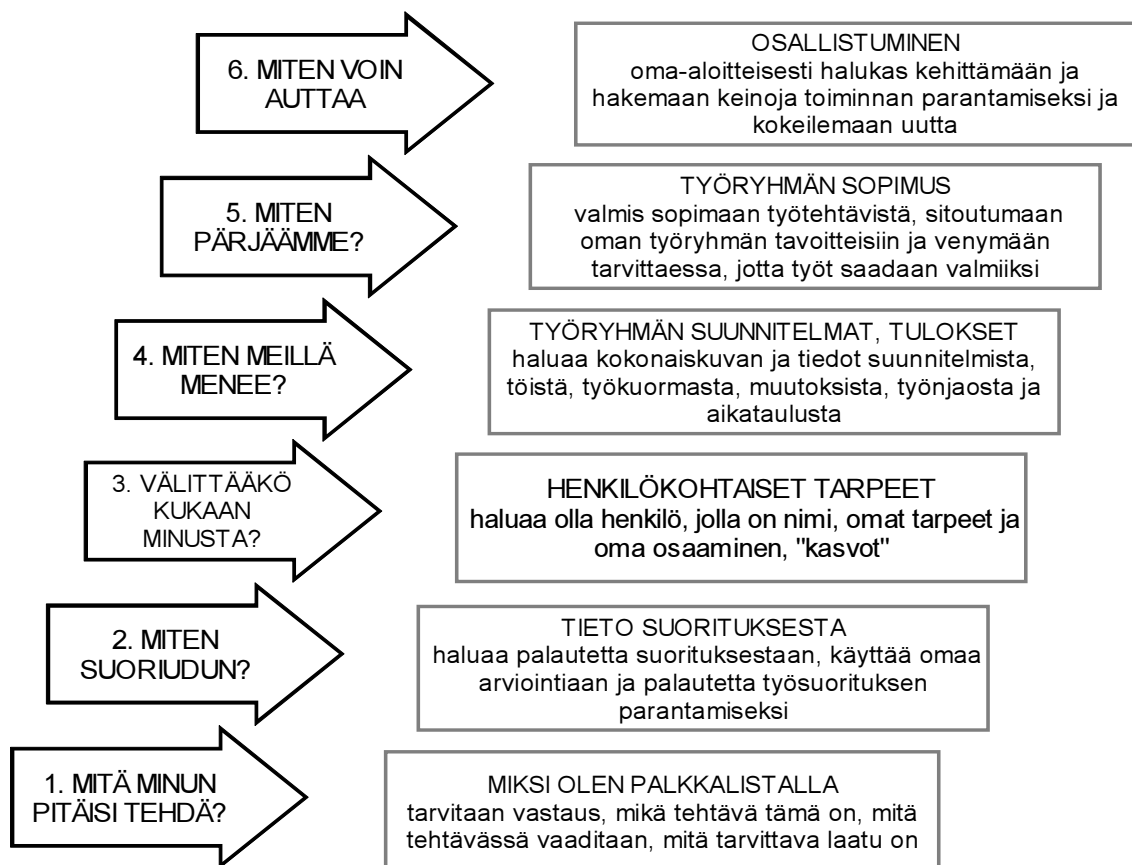
Kehityskeskustelut voidaan määritellä (1) osaksi johtamisjärjestelmää, jolloin ne on myös ohjeistettu ja niiden toteutumista seurataan. Lisäksi kehityskeskustelut voivat olla (2) osa tavoitteiden saavuttamisen arviointia tai (3) osaamiseen ja kehitystarpeisiin keskittymistä, mutta ne voivat ilmentää myös (4) yhteisvastuullista työkuultuuria, jossa vuorovaikutukseen pyritään jatkuvasti eikä siihen käytetä muodollista kehityskeskustelumallia. [Val00]

Kehityskeskustelu tarjoaa alaiselle tietoa ja vaikutusmahdollisuuden. Alainen saa tietää, mitä häneltä odotetaan ja miten hänen työpanoksen sitoutuu kokonaisuuteen. Samalla hän voi kertoa toiveistaan. Kehityskeskusteluista ei ole siis hyötyä, jos alainen ei uskalla kertoa

todellisia mielipiteitään esimerkiksi määräaikaisen työsuhteen vuoksi. Esimies saa kehityskeskustelusta varmuuden siitä, että alaisen työhön ja asioihin on paneuduttu. [Val00]

Kehityskeskustelusta tulee löytyä kolme osiota: tavoitekeskustelu, tuloskeskustelu ja kehittymiskeskustelu. *Tavoitekeskustelussa* käydään läpi yrityksen, yksikön ja prosessien strategiaa ja tavoitteita. *Tuloskeskustelussa* annetaan palautetta ja arvioidaan saavutettuja tuloksia. *Kehittymiskeskustelussa* suunnitellaan tulevaisuutta sekä lyhyellä että pitkällä aikavälillä. [Val00]

Kehityskeskustelu on keino antaa vastauksia, joita alainen tarvitsee. Valpola on koonnut [Val00] kysymykset, joihin alainen tarvitsee vastaukset sitoutuakseen organisaation tavoitteisiin ja antaakseen sille parhaan mahdollisen työpanoksensa. Johtamisen kysymykset on koottu kuvaan 14.



Kuva 14: Johtamisen kysymykset [Val00]

4.3 YKSILÖ

Tehokkuuden pienin yksikkö on yksilö. Tiimit voivat koostua vain yksilöistä. Tiimi on tuottoisin vain, kun sen jokainen yksilö on tuottoisa ja tehokas. Vain tehokkaat yksilöt halutaan mukaan ja näistä yksilöistä halutaan pitää kiinni. Miten yksilöstä saadaan tehokas ja miten hänet saadaan pysymään yrityksessä? Yrityksen kannalta on mieluista, että työntekijä viihtyy työpaikallaan ja tekee ylitöitä. Yksilön kannalta voi pidemmän päälle käydä huonosti, jos elämässä ei ole muuta kuin työ. Ihminen tarvitse elämässään useamman sektorin: työ, koti ja harrastukset, jotka mahdollistavat uusiutumisen ja joissa vuorollaan tulevat *onnistumiskokemukset* tasapainottavat toisen sektorin epäonnistumisia. Loppujen lopuksi on myös yrityksen kannalta pitkällä tähtäimellä edullisempaa, että työntekijällä on työn lisäksi harrastuksia.

Jokainen yksilö on yksilöllinen johtuen esimerkiksi erilaisista koulutus- ja perhetaustoista. Toisaalta koulutustausta voi olla yhdistävä tekijä, joka auttaa kestämaan toisen erilaisuutta [Dun98]. Maunula nimeää [Mau96] työyhteisöstä erikoisesti erottuvat henkilöt ”sisäisiksi yrittäjiksi”. Nämä sisäiset yrittäjät ovat useimmiten omatoimisia, joustavia, luovia, halukkaita tarttumaan uuteen ja kyvykkäitä ottamaan riskejä. Tällaiset henkilöt voidaan kokea työpaikalla ärsyttäviksi, joka asiaan puuttuviksi häiriköiksi tai kekseliäiksi, työyhteisön dynaamisuuden kantaviksi voimiksi.

Se, miten hyvin sisäisten yrittäjien positiiviset voimavarat saadaan työyhteisön käyttöön, riippuu mm. esimiehen taidoista ja mahdollisuuksista antaa näiden henkilöiden toteuttaa itseään. Tärkeää on myös työtovereiden kyky nähdä tällainen eroavaisuus positiivisena ominaisuutena ilman kateutta tai syrjintää, koska ne *hulluimmat ideat* voivat osoittautua juuri parhaimmiksi. Esimerkiksi aivoriivessä huono idea voi toimia hyvänä alkuna tai katalysaattorina. Erilaisuuteen palataan kappaleessa 5.2.3 Erilaisuuden hyödyntäminen.

4.3.1 Motivaatio

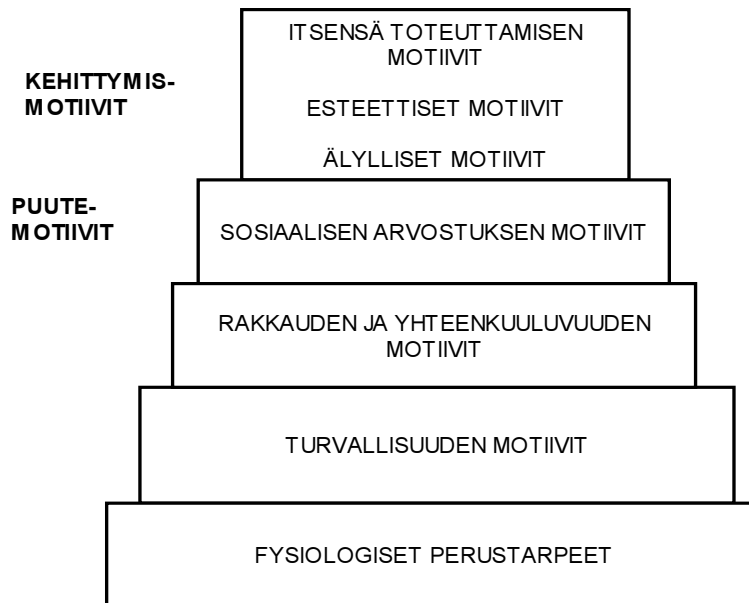
Organisaation tuottavuus ratkeaa suurelta osin sen mukaan, miten motivoituneita siellä työskentelevät yksilöt ovat [Juu92]. Ohjelmistojen kehittäjät motivoituvat eniten: saavutuksista, haastavista tehtävistä, itsensä kehittämisen mahdollisuudesta sekä mielenkiintoisesta työstä (työstä itsestään). Johtamisella on myös suuri vaikutus motivaatiotasoon. Lisäksi tärkeitä tekijöitä ovat: suhteet työtovereihin, uuden luomisen mahdollisuus ja riippumattomuus, vastuu, arvontanto, arvostus sekä huomatuksi tuleminen.

Jonkin verran motivaatioon voi vaikuttaa myös oma kunnianhimo, sopiva työpaine, työolosuhteet, palkka ja etuisuudet sekä urakehityksen mahdollisuus. [Vir01]

Ennen kuin kannattaa käyttää energiaa motivoimiseen on hyvä varmistaa, että muut toimimisen edellytykset ovat kunnossa. Tällaisia ovat esimerkiksi hyvät työskentelyolosuhteet eli riittävä *ergonomia*. Ergonomian tavoitteena on kehittää työoloja vastaamaan ihmisen fyysisiä, psyykkisiä ja sosiaalisia valmiuksia. Samalla pidetään yllä työkykyä ja terveyttä esimerkiksi työpaikkaliikunnan avulla [Erg02]. Lisäksi tarvitaan riittävät työkalut, riittävä tieto ja taito sekä *oikeantasoiset* ja työn sujumuuden kannalta *järkevät* tehtävät. Jokainen yksilö voi motivoitua eri asioiden vaikutuksesta, mutta on olemassa joitakin yleisiä motivoinnin lainalaisuuksia ja keinoja.

Ihminen ei ole kognitiivisestikaan hyvässä terässä ilman motivaatiota. Tämä tarkoittaa sitä, että luovuuteen tarvitaan motivaatiota. Työhönottajan tulee valita työhön henkilö, jolle tehtävät antavat riittävästi haasteita. Tämä on jo tiedostettu ja monet työnhakijat ovat pettyneet, kun heitä ei ole valittu siksi, että he ovat ”liian päteviä kyseiseen tehtävään”. Pidemmällä tähtäimellä on kuitenkin hyväksi niin yritykselle kuin myös työnhakijalle, että jokainen saa omaa osaamistasoaan vastaavan työn [VLK98].

Tunnetuin motivaatioteoria on Maslowin *tarveteoria*. Abraham Maslowin 1960-luvulla kehittämän tarvehierarkian (ks. kuva 15) perusidea on, että jos *puutemotiivit* jäävät toteutumatta, ei ylemmällekin tasolle päästä eli *kehittymismotiivit* jäävät luultavasti toteutumatta. Tämä tarkoittaa esimerkiksi sitä, että älylliset (tutkimisen, tietämisen ja ymmärtämisen tarve), esteettiset (uuden luomisen tarve) tai itsensä toteuttamisen motiivit eivät toteudu, jos alemmalla tasolla olevat fyysiset perustarpeet, kuten nälkä, jano tai unentarve, turvallisuuden motiivit (esim. jatkuva työsuhde ja turvallinen työympäristö), rakkauden ja yhteenkuuluvuuden motiivit (esim. yhteishenki) sekä sosiaalisen arvostuksen motiivit (esim. asema organisaatiossa) eivät ole toteutuneet *riittävällä* tasolla.



Kuva 15: Maslowin tarvehierarkia [VuT95], [Juu92]

Yleisesti voidaan todeta ihmisen olevan varsin tyytyväinen, jos Maslowin tarvehierarkian ylin taso saavutetaan eli jos ihminen pääsee toteuttamaan itseään. Tärkeänä koetaan myös turvallisuuden ja liittynän tarpeiden tyydyttyminen työssä [Juu92]. Jos ihmiseltä vaaditaan työssä kehittämismotiivien toteutumista, mikä on työssä viihtymisenkin kannalta suotavaa, tulee puutemotiivien olla toteutuneet ihmisen elämässä riittävällä tasolla. Maslowin tarvehierarkiaa on kritisoitu sen suoraviivaisen ja ihmisen käyttäytymistä yksinkertaistavan ajattelutavan vuoksi, mutta sitä voi kuitenkin käyttää hyväkseen esimies, joka ei ole selvillä alaistensa tarpeista. [VuT95], [HeB90]

Motivaatio työtä kohtaan syntyy tarveteorian mukaan siis siitä, että henkilö voi tyydyttää itselleen tärkeitä tarpeita työssään. *Odotusarvoteorian* kannattajat ovat sitä mieltä, että motivaatio työssä syntyy siitä, että työ tarjoaa henkilölle hänen arvostamiensa päämäärien toteutumismahdollisuuksia. Henkilö vertailee sitä, mitä hän saa, siihen, mitä hänen tulisi saada. Jos henkilö panostaa työhönsä paljon, sitä enemmän hän odottaa saavansa palkkioita. Tyytyväisyys syntyy, kun palkkiot ovat sitä, mitä henkilö odottikin. Jos palkkiot ovat liian suuria, voi henkilö kokea jopa syyllisyyttä, mutta hän voi myös yrittää tehdä parempaa työtä, jotta hän olisi palkkion arvoinen. [Juu92], [HeB90]

Motivaatio muodostuu myös sosiaalisen vertailun kautta. *Viiteryhmäteorian* mukaan henkilö vertailee itseään muihin saman työpaikan työntekijöihin sekä kokemuksiinsa

aikaisemmissa työpaikoissa. Jos henkilö saa mielestään riittävästi palkkioita suhteessa omaan työpanokseensa ja verrattuna muihin, on hän tyytyväinen työhönsä. Tämänkaltaista vertailuun perustuvaa motivaatioteoriaa voidaan sanoa myös *oikeudenmukaisuusteoriaksi*. [Juu92]

Ihmistä motivoivat tekijät voidaan luokitella: sisäisiin yllykkeisiin (tarpeet ja vietit), sosiaalisiin vertailuihin, tulevaisuuden odotuksiin sekä mielenkiintoisiin tehtäviin ja haastaviin päämääriin [Juu92].

4.3.2 Työtyytyväisyys

Työtyytyväisyydellä ei ole välttämättä suoraa yhteyttä työsuorituksen tehokkuuteen, mutta se vaikuttaa kuitenkin työssä viihtymiseen ja sitä kautta työhön sitoutumiseen. Tämä tarkoittaa sitä, että työtyytyväisyys vähentää työpaikan vaihdossuunnitelmia. Toisaalta tähän vaikuttaa myös avoimien työpaikkojen määrä. Jos avoimia työpaikkoja on runsaasti tarjolla, on helpompaa suunnitella työpaikan vaihtoa. [Juu92]

Työtyytyväisyyttä edistävät esimerkiksi osallistuva johtamistapa ja hyvä ilmapiiri työryhmässä ja organisaatiossa. Työtyytyväisyys kietoutuu kuitenkin yksilötasolla motivaatioon liittyvään *tarveteoriaan*, *odotusarvoteoriaan* ja *viiteryhmäteoriaan* [Juu92]. Pelkistetysti voidaan todeta ihmisen olevan tyytyväinen työhönsä, jos hän saa, mitä haluaa, mitä odottaa saavansa ja lisäksi saa mielestään riittävästi verrattuna muihin. Tässä yhteydessä ajatellaan ihmisen saavan palkan lisäksi myös muita palkkioita. Tällaisia *palkkioita* voivat olla esimerkiksi: haasteellinen ja kehittävä työ, hyvät työolosuhteet, hyvä ilmapiiri, arvostuksen saaminen tehdystä työstä ja vaikutusmahdollisuudet omaan työhön [Toi00].

Työtyytyväisyyden vastakohta on *työtyytymättömyys*. Työtyytymättömyydestä voi seurata työtehon laskua, poissaoloja ja työpaikan vaihdoksia [Juu92] sekä stressiä. *Aktiivisesti* tyytymätön työntekijä voi olla kuitenkin hyvä työntekijä, vaikka työtoverit voivat pitää häntä hankalana, koska hän ajaa eteenpäin muutoksia. *Passiivisesti* tyytymätön taas valittaa ja marisee ja täten levittää ympärilleen masennusta ja vie työn ilon muiltakin. [Typ02]

4.3.3 Työaika, vapaa-aika ja työvire

Työnantajan ja samassa projektissa olevien jäsenten pitää tietää, milloin henkilö on töissä, jotta työnjako, tiedotus ja tarvittava yhteistyö onnistuvat. Tähän ratkaisuna voi olla

jonkinlainen työajanseuranta tai raportointi. Työajanseuranta on tärkeää myös tehokkuuden vuoksi. Ilman seurantaa voidaan tehdä liian vähän töitä mutta myös liikaa. Esimerkiksi univaje ehkäisee luovuutta ja näin alentaa työtehoa. Tutkimuksissa on päädytty siihen, että kokonaistyöaika on tärkein unen pituuteen vaikuttava tekijä [Työ02]. Työnantajan tulisi mahdollistaa olosuhteet, joiden puitteissa yksilön on mahdollista olla hyvässä ”työvireessä”. On kuitenkin yksilön vastuulla, kuinka hän tätä mahdollisuutta käyttää.

Tietotekniikan ammattilaisille vuonna 2001 tehdyssä kyselytutkimuksessa selvisi mm. että henkilöillä, joilla oli täysin vapaa työaika, keskimääräinen viikkotyöaika oli 53 tuntia (normaali viikkotyöaika on 35-40 tuntia). Tämän oletetaan johtuvan työn korkeista sisällöllisistä vaatimuksista. Työaikaan voivat vaikuttaa lisäävästi myös omat vaikutusmahdollisuudet, mutta toisaalta ne voivat myös lyhentää kokonaistyöaikaa. Omat vaikutusmahdollisuudet työaikaan vaikuttavat kokonaistyöaikaa lisäävästi, kun yrityksessä ylitöitä pidetään sallittuna ja ”suotavana”. Kokonaistyöaika voi lyhentyä omien vaikutusmahdollisuuksien seurauksena, jos työyhteisön *työaikanormit* sen sallivat. [Työ02]

Epäsäännöllisistä työpäivistä voi koitua jopa vakavia sairauksia. Vuorotyöntekijöillä on havaittu tyypillisiä stressioireita, kuten ruuansulatuselimistön oireet (näristys ja vatsakivut). Tutkimusten mukaan vuorotyötä tekevillä on päivätyöläisiä enemmän myös erilaisia psykosomaattisia oireita (väsymys, hermostuneisuus ja masennus). Sopiva työaika lähtee siitä, että työntekijällä on mahdollisuus vaikuttaa omaan työaikaansa.

Yhdessä suunnitellut työajat luovat työtyytyväisyyttä ja sitä kautta työtehoa. Ei ole järkevää kuitenkaan hypätä ääripäähän ja antaa työntekijän tehdä töitä oman mielensä mukaan. Välillä on mentävä kotiinkin. Ihmisen tulee tietää, mikä aika on hänen vapaa-aikaansa ja mikä työaikaa. On vaarallista ottaa töitä kotiin. Jos vapaa-aikaa ja työaikaa ei ole erotettu, voi kaikki ihmisen aika muuttua pikku hiljaa työajaksi. Jossain tilanteessa pitkäkestoinen stressi voi puhjeta pahaksi loppuunpalamiseksi, kun vapaa-aikaa, jolloin stressi helpottaa, ei ole. [Toi00]

Töissä jaksamiseen vaikuttaa riittävä vapaa-aika, jolloin täytyy tyydyttää mm. unentarve. Aikaa tulee olla myös perheenjäsenten kanssa seurusteluun ja liikunnan tarpeen tyydyttämiseen. Liikunta on suositeltavaa ajatustyötä tekevälle fyysisen toimintakyvyn ylläpitämiseksi, mutta myös henkinen toimintakyky paranee sopivan liikuntamäärän myötä. Unen laatu paranee samalla, kun ihminen liikunnan kautta rentoutuu paremmin sekä lihaksistoltaan että mieleltään. Liikunta parantaa myös itsetuntoa ja vähentää

masentuneisuutta. Raskasta liikuntaa ei tule harrastaa kuitenkaan juuri ennen nukkumaanmenoa, koska elimistö joutuu toipumaan liikunnasta ennen todellista rentoutumista. Suositeltavinta liikuntaa on hyötyliikunta: työmatkojen kävely tai pyöräily ja portaiden käyttäminen hissin sijaan. [Toi00]

Työvireyteen vaikuttavat terveelliset elämäntavat niin vapaa- kuin työaikanakin. Riittävän liikunnan ja unen ohella tärkeää on myös monipuolinen ruoka, joka sisältää riittävästi vitamiineja ja ravintoaineita. Ruokailtaessa ei saa olla kiire; hotkiminen aiheuttaa vain ruuansulatusongelmia. Liian suuret ruoka-annokset väsyttävät eivätkä piristä. Suklaapatukat ja karkit aiheuttavat verensokerin nousemisen korkealle hetkeksi, jolloin tulee hetkellisesti pirteä olo, mutta hetken päästä on taas nälkä. [Toi00]

4.3.4 Stressi

Tietotyöläisiltä vaaditaan useimmiten normaalia pidempiä työaikoja sekä paljon henkisiä voimavaroja. Lisäksi tietotyö on pääasiassa *ajattelua vaativaa aivotyötä*. Vaikka tietotyöläiset itse arvostavat työnsä haasteellisuutta, ei jatkuvaa kiirettä ja luovuuden vaatimuksia kannata aliarvioida, kun kartoitetaan stressitekijöitä. IT-ammattilaisilla on keskimäärin hyvä itsetunto, kun taas ammatillinen itsetunto oli alentunut monilla vuonna 2001 tehdyn kyselyn mukaan. Alan nopea kehitysvauhti johtaa riittämättömyyden tunteeseen ja pelkoon, ettei pysy kehityksessä mukana. Tämä on kiireen ohella suurimpia stressitekijöitä IT-alalla. [Työ02]

Esimiesten ja työtovereiden voi olla vaikea havaita toisen ihmisen työstressiä siksi, että on erilaisia tapoja oireilla stressiä (nimenomaan pahanlaatuista stressiä). Kaksi ääripäätä ovat *aktiivinen* ja *passiivinen* [Toi94].

Pako tai taisto on aktiivinen tapa. Stressaantunut henkilö pyrkii selviytymään tekemällä jotakin, jopa aggressiivisesti, jotta hänen olisi parempi olla. Hän voi olla hyvinkin hyökkäävä, mistä seuraa, etteivät työtoverit mielellään ole häntä tukemassa. Tarvittaessa hän voi myös paeta paikalta. Tällainen henkilö ”käy kuumana” niin henkisesti kuin fyysisestikin. Hänen pulssinsa voi olla kiihtynyttä, hengityksensä tihentyynyttä ja verenpaineensa koholla, sillä hänen sympaattinen hermostonsa toimii yliaktiivisesti. Jos tämä tila kestää pitkään, vaarassa ovat sydän ja verenkiertoelimistö. [Toi94]

Passiivisesti reagoiva henkilö puolestaan voi olla hyvin myötäilevä, jopa liiankin kiltti. Hän ei jaksaa taistella, vaan hän *lamaantuu* ja *luovuttaa*. Näin reagoi henkilö, joka arvioi,

että mitään ei ole tehtävissä. Tällainen henkilö voi vaipua apatiaan, toivottomuuteen tai masennukseen. Hän odottaa vain parempaa huomista. Hänen pulssinsa voi olla normaalia alhaisempi ja hänen verenpaineensa on laskussa. Hänen parasympaattinen hermostonsa reagoi, eli vaarassa ovat maha, suolisto ja keuhkoputket. Stressiin tällä tavoin reagoivan ei useinkaan havaita olevan avun tarpeessa. [Toi94]

Se, kummalla tavalla henkilö suhtautuu stressiin, on osittain persoonallisuudesta kiinni. Reagointitapa on jo geneeissä, mutta siihen voi tulla muutoksia elämäkokemusten myötä. Miehet oireilevat useimmin aktiivisesti käyttäen järkeistämistä apunaan ja naiset passiivisesti ja tunnepitoisesti. Toisaalta sama henkilö voi oireilla myös molemmilla tavoilla. Oireilutapa voi riippua esimerkiksi tilanteesta. Eri tilanteissa toinen tapa toimiikin paremmin. Aktiivinen tapa tuo tuloksia, jos asioille voidaan tehdä jotakin. Näin voidaan toimia niin kauan kuin on *hallinnan tunnetta*. Kun hallinnantunne häviää ollaan ”vaaravyöhykkeellä”, mikä voi johtaa syviin masennustiloihin, joista selviytyäkseen yksilö voi tarvita pitkiäkin sairauslomaa ja hoitoa. [Toi94]

Ensisijaisena apuna kiireiseen ja haasteelliseen työhön tietotyöläisten keskuudessa koetaan *toimiva yhteishenki* ja *hyvät tiimit*, jotka ovat syntyneet vuosien aikana. Myös työajan ulkopuolella vietetty *yhteinen aika* vaikuttaa siihen, että työtoveria on helpompi lähestyä [Työ02]. Stressinhallintaan on keinoina lisäksi erilaiset rentoutusharjoitukset, joita voidaan tehdä esimerkiksi kesken työn teon omalla tuolilla. Yksi tapa on laittaa silmät kiinni ja ajatella mukavia asioita sekä hengitellä rauhallisesti [Toi94].

4.3.5 Luovuus

Kognitiiviset taidot ohjaavat ihmisiä ongelmanratkaisutehtävissä. Kun vastaan tulee uudenlaisia ongelmia, tarvitaan *luovuutta* [EyK95]. Luova ihminen käyttää aiemmin opittuja ratkaisumalleja, mutta luovasti. Hän assosioi jopa niin voimakkaasti, että voi olla vaikea nähdä asioiden yhteyttä. Luovan ihmisen ideoita voidaan pitääkin ensin ihan hulluina. Luova ihminen näkee ongelmia, joita muut eivät havaitse ja kykenee tuottamaan uusia ideoita, jotka ovat myös omaperäisiä. Luovuus ei synny kiireen keskellä vaan luovaan prosessiin ihminen tarvitsee riittävästi *tilaa* ja *aikaa* [UuP99]. Tämä tarkoittaa sitä, että tämän päivän ”laiskottelu” voi olla pohjana huomisen luovalle prosessille.

Luovuus ei ole sama asia kuin älykkyys. Keskitason älykkyys riittää korkeatasoiseen luovuuteen. Luovuuteen liittyy kuitenkin *arviointikyky* ja *motivaatio*. Jotta luovuudesta

olisi hyötyä, pitää ideoita pystyä arvioimaan. Motivaatio taas syntyy mm. oikeanlaisesta ilmapiiristä (ks. kappale 4.3.1 Motivaatio). Luovuus voidaan nujertaa antamalla kaikki vastaukset valmiina ja näkemällä vain yksi mahdollinen ratkaisu. Jos ihminen oppii sopeutumaan ylhäältä saneltuihin standardeihin, on turha odottaa luovuutta [UuP99].

Maslowin mukaan jokainen voi olla luova, jos siihen annetaan mahdollisuus. Maslowin hierarkiassa luovuus on korkeimmalla tasolla: itsensä toteuttamista. Esimerkiksi perheenäidit joutuvat päivittäin kehittämään luovia ideoita huolehtiessaan perheistään. Luovaksi tiedemieheksi tai taiteilijaksi ei kuitenkaan tulla hetkessä, vaan se vaatii syvällisiä tietoja ja taitoja sekä paljon työtä [UuP99].

5 PROJEKTITYÖSKENTELY

Ohjelmistot rakennetaan projekteina. Ohjelmiston elinkaaren voi sijoittua useampiakin projekteja riippuen organisaatiosta. Tuotteen tekemisen lisäksi voi rinnalla olla muun muassa ideointiprojekti, suunnitteluprojekti tai ylläpitoprojekti, jotka voivat myös olla samaa projektia. On myös mahdollista, että tuotteen eri komponentit koostetaan eri osaprojekteissa. Tällöin on tärkeää hahmottaa eri projektien väliset suhteet.

Projekti on *ainutkertainen* ajallisesti *rajattu* työsuoritus, jolla on rajalliset *resurssit* ja jolle on asetettu *tulostavoitteet* ja projektin ajan voimassa oleva *organisaatio* [LoG96].

Projektista erotetaan useimmiten neljä vaihetta:

- 1 ajatus, hedelmöityminen (conception)
- 2 kehitys (development)
- 3 toiminta (realisation) ja
- 4 päättäminen (termination) [Log96].

Hedelmöitymisvaiheessa idea tuotteesta esitellään organisaatiossa ja hyväksytään projektiksi eli asetetaan projektin tavoitteet. Määritellään tarkasti tuotteen tavoite ja suunnittelutavat. *Kehitysvaiheessa* täytyy olla määritelty ryhmä projektin toteuttajista ja projektipäällikkö. Tässä vaiheessa ollaan tyypillisesti paljon yhteydessä asiakkaaseen. *Toimintavaiheessa* projekti on täydessä vauhdissa ja asiakasta informoidaan projektin kulusta, tuloksista, rahankäytöstä ja kustannuksista. Lisäksi kirjataan kohdatut ongelmat ja näiden ratkaisut. *Päättymisvaiheessa* on tärkeää dokumentoida projektiin liittyen onnistumiset ja epäonnistumiset tulevia projekteja varten. [LoG96]

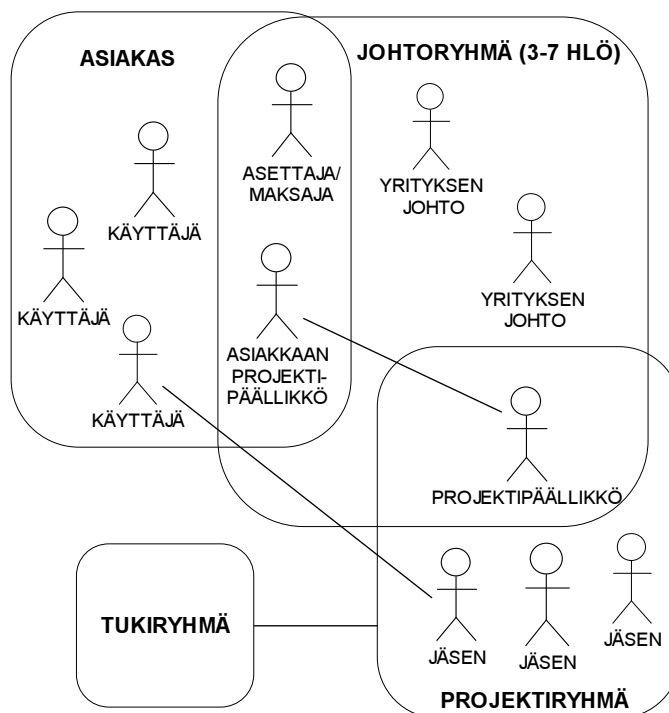
Projektin elinkaaren tueksi voidaan ottaa jokin ohjelmistoprosessin vaihejakomalleista, joille on useimmiten mahdollisuus löytää myös *CASE*-välineen (*Computer Assisted Software Engineering*) tuki. RUP-mallin vaiheet (inception, elaboration, construction ja transition) ovat osittain rinnastettavissa edellä esitettyihin vaiheisiin. RUP tarjoaa tukea myös projektinhallintaan [Roy98].

5.1 PROJEKTINHALLINTA

Projektinhallinta on esimerkiksi laadunvarmistusta, tuotteenhallintaa, dokumentointia ja vaatimustenhallintaa läpi projektin *suunnittelun, käynnistämisen, toteuttamisen ja päättämisen* [HaM01]. Projektinhallinnasta on vastuussa jokainen projektiin osallistuva oman roolinsa edellyttämällä tavalla. Jotta projekti pysyisi hallinnassa tarvitaan päteviä ja kokeneita esimiehiä. Heidän avukseen on kehitetty myöskin projektinhallintaohjelmistoja. Lisäksi projektinhallintaan liittyy kiinteästi riskienhallinta.

5.1.1 Projektiorganisaatio

Jotta projektinhallinta olisi mahdollista, on tärkeää määritellä koko projektiorganisaatio ja vastuut (ks. kuva 16).



Kuva 16: Projektiorganisaation ryhmät ja roolit [Poh02]

Projektiorganisaation eri ryhmillä erilaisia tehtäviä. *Projektiryhmä* toteuttaa annetun tehtävän. *Johtoryhmän* tehtävänä on projektin ulkoinen ohjaus, suunnitelmien ja välitulosten hyväksyminen sekä päätösten tekeminen. Johtoryhmän täytyy olla

päätösvaltainen, siksi siihen kuuluu asiakasprojekteissa *asiakkaan edustaja*. *Projektiryhmä* saa tiedot yleensä projektipäällikön välityksellä. Johtoryhmä ei puutu käytännön asioihin. Projektiryhmä on yhteydessä asiakkaaseen ja tuleviin *käyttäjiin*, jotta vaatimukset selviävät. Asiakkaan ja projektiryhmän projektipäälliköt edustavat alaisiaan samalla kun ovat yhteydessä toisiinsa ja hoitavat tiedotusta. Lisäksi ryhmä voi tarvita *asiantuntijoiden* apua esimerkiksi käytettävyyden parantamiseksi tai tietoliikenne-ratkaisujen selvittämiseksi. Tällaista erityisasiantuntijoiden ryhmää voidaan nimittää *tukiryhmäksi*.

Projektipäällikkö on eräänlainen johtaja tai tiiminvetäjä. Projektipäälliköllä on tärkeä rooli tiedottajana ja vastuunkantajana. Hän toimii tukena ryhmäläisilleen ja jakaa tehtäviä näille. Projektipäällikkö suunnittelee, mieluiten ryhmänsä kanssa yhdessä, sen, miten annettu tehtävä käytännön tasolla toteutetaan. [Pel96]

Kuvassa 16 on kuvattu yksinkertainen projekti, johon on osallisena vain yksi tai kaksi yritystä. Nykyisin projektit usein myös verkottuvat, jolloin mukana voi olla useita, jopa kymmeniä, yrityksiä.

5.1.2 Suunnittelu ja käynnistäminen

Projektin suunnittelu on aiheeseen tutustumista, projektiorganisaation perustamista ja tarvittavien välineiden ja tekniikoiden tunnistamista. Suunnitteluvaiheessa tehdään *projektisuunnitelma*, jossa määritellään muun muassa [Poh02]:

- projektin tavoitteet,
- projektiin osallistuvat henkilöt ja heidän vastuunsa,
- vaiheistus ja tehtävien riippuvuudet,
- aikataulut,
- talousarvio,
- henkilöressurssien käyttö ja työnjako,
- työtilat ja laitteet,
- raportointi- ja dokumentointisuunnitelma,
- projektin valvonta ja ohjaus,
- projektisuunnitelman ylläpito,
- riskienhallinta ja
- laadunvarmistus.

Ennen kuin projekti käynnistetään tulee miettiä paljon asioita. Voi olla myös, että kun talousarvio valmistuu, koko projekti perutaan. Suunnittelussa on tärkeää huomioida *tekniset* ja *inhimilliset* onnistumismahdollisuudet. Projektisuunnitelman tekeminen ja julkistaminen osapuolille antaa tietoa ja varmuutta siitä, mihin ollaan ryhtymässä. Projektin toteuttajat ovat sitä motivoituneempia, mitä enemmän he voivat vaikuttaa suunnitteluun [Som98].

Suunnittelussa voi olla vaikeaa ottaa huomioon inhimillisiä tekijöitä. Kuitenkin on kehitetty esimerkiksi *COCOMO-malli (CONstructive COst MOdel)*, jossa pyritään huomioimaan toteuttajien taitotasoa. COCOMO-mallin tarkoitus on arvioida tulevia kustannuksia. *Kustannuskertoimiin* vaikuttavat muun muassa suunnittelijoiden kyvykkyys, ohjelmointikielen tuntemus ja sovellusalueen tuntemus. Lisäksi kiinnitetään huomiota monenlaisiin teknisempiin vaatimuksiin, kuten luotettavuusvaatimuksiin, monimutkaisuuteen ja suoritusaikavaatimukseen [Roy98]. Arvioitujen kustannusten vaihteluväli on melkoinen, sillä parhaimpien eli pienempien kustannuskertoimien ollessa kyseessä *kustannustyömääräarvio* voidaan jakaa noin kymmenellä, kun se suurimmillaan on kerrottava 70:llä [HaM01].

Projektin suunnitteluvaiheen jälkeen selkeä aloitus on tärkeä, koska silloin projektin organisaatioon kuuluvat ihmiset tietävät olevansa mukana projektissa. *Käynnistämistilaisuudessa* ovat mukana kaikki projektiin osallistuvat henkilöt. Käynnistäminen voi olla esimerkiksi seminaari tai kokous kahvipöydän ympärillä pienimuotoisemmissa projekteissa. Tilaisuuden tulee olla kuitenkin riittävän virallinen, jottei kenellekään jää epäselväksi kokouksen tarkoitus. Toisaalta projektista täytyy olla myös niin hyvät dokumentaatiot olemassa, että myöhemmin mukaan tulevat henkilöt pystyvät omaksumaan projektin tavoitteet ja käytännöt. [HaM01]

5.1.3 Riskien hallinta

Riskien hallinta suunnitellaan projektin alussa. Projektin aikana riskien hallinta – suunnitelmaa noudatetaan ja muokataan tarvittaessa. Lopussa riskien toteutumista arvioidaan. Riskien hallintaa tukemaan voisi valita spiraalimallin. Tätä vaihejakomallia voisi käyttää esimerkiksi kriittisissä projekteissa, joiden epäonnistumisesta seuraisi pahoja ongelmia. [Som95]

Projektisuunnitelmaan liitetään useimmiten *riskilista* mahdollisista riskeistä, joita projekti voi kohdata. Tyypillisiä riskejä muodostavat esimerkiksi uudet teknologiat, epärealistiset suunnitelmat ja vaatimusten muuttuminen. Riskilistaan tulee kirjata myös, mitä tehdään kun riski toteutuu. Ajatus siitä, että pahimpaankin on varauduttu luo turvallisuutta. Riskien määrittämiseen liittyy riskien *minimoiminen* ja *luokittelu*. Osa riskien todennäköisestä toteutumisesta voidaan minimoida jo alussa. [Pel96]

Toiset riskit taas voivat olla tämän projektin kannalta vähäpätöisempiä kuin toiset. Projektin kannalta merkitykselliset riskit tarvitsevat *torjuntasuunnitelman*. Tyypillisesti riskeihin voidaan varautua hyvällä määrittelyllä, henkilöstön kouluttamisella ja sitouttamisella sekä riittäväillä varatoimenpiteillä, kuten esimerkiksi turvatoimet, varmuuskopioinnit, sääntöjen noudattaminen ja vakuutukset. [Pel96]

Aikataulun pettäessä voidaan tärkein osa toiminnallisuudesta ottaa ensin käyttöön esimerkiksi keskeisimmät käyttötapaukset. Siksi on tärkeää määritellä ohjelmiston toimintojen tärkeys jo vaatimuksia kerätessä. [HaM01]

5.1.4 Toteutus ja päättäminen

Projektin toteutusvaiheessa tulee valvoa *välitavoitteiden saavuttamista* sekä kustannus- ja aikataulusuunnitelmien toteutumista. Jos tavoitteisiin ei päästä voi olla tarpeen *muuttaa projektisuunnitelmaa*. Tällaisesta toimenpiteestä päättää johtoryhmä.

Aikataulujen tekemisessä ylioptimistisuus on yleistä ja sitä tulee välttää. Osatehtävien välille tulee jättää riittävästi pelivaraa. Kun tehtävien riippuvuudet toisistaan määritellään hyvin, voidaan kiireen yllättäessä keskittyä kriittisimpiin tehtäviin. Tosiasia on, että suuri osa projekteista myöhästyy tai ei valmistu koskaan [HaM01]. Koko projektiorganisaatiota voi masentaa odotusten saavuttamattomuuden tuoma pettymys. Joskus voi olla tärkeää keskeyttää projekti. Jos projekti on selvästi epäonnistumassa, luovuttamisesta seuraa helpotus, vaikka myös pettymyksen tunteita voi esiintyä.

Projektin päättää virallisesti johtoryhmä. *Päättäminen* on projektiin osallistuville tärkeää, jotta heille ei kerry ”ikiprojekteja”. Jos projekteja ei päätetä, voi yhdelle henkilölle kertyä paljon ylimääräistä ”näkömätöntä” työtä. Siksi olisikin hyvä, että ohjelmistojen ylläpidosta tehtäisiin oma projektinsa eikä sitä jätettäisi varsinaisten ohjelmiston tekijöiden harteille. Myös keskeytetty projekti täytyy päättää virallisesti, ettei projektin jäsenillä jää epävarma olo projektin jatkumisesta. [Poh02]

5.1.5 Automatisointi

Projektinhallintaa on alettu myös automatisoida. Työkaluja on kehitetty projektien aikataulujen ja resurssien hallintaan. Yksi projektin suunnittelu- ja seurantatyökalu on Microsoft Project -ohjelmisto. Ohjelmisto mahdollistaa ajan, kustannusten ja henkilöstöresurssien yhtäaikaisen suunnittelun ja suunnitelmien toteutumisen seurannan. Ohjelmiston avulla voidaan muun muassa määritellä kriittiset päivämäärät sekä rajoittaa kustannuksia. Ohjelmisto tarjoaa visuaalisia apuvälineitä suunnitteluun ja seurantaan. [ChJ00]

Projektipäälliköille voivat olla jo ennestään tuttuja toimintaverkkoajatus [LoG96] ja Gantt-kaaviot, jolloin ohjelmiston käyttöön siirtyminen on vaivattomampaa [ChJ00]. Projektinhallinta-ohjelmiston käyttö ei kuitenkaan takaa projektinhallinnan onnistumista, ja toisaalta ilman ohjelmistoakin se voi onnistua.

5.2 TIIMIT

Projekteissa tarvittavat ryhmät voivat olla myös tiimejä. Tiimi on *ryhmä* ihmisiä, jotka ovat *yhteisvastuullisia* kokonaisesta työprosessista tai alueesta. Tarvitaan siis ensin ryhmä. "Ryhmä on kahden tai useamman keskenään jatkuvasti vuorovaikutuksessa olevan henkilön muodostama yhteisö, jolla on yhteinen päämäärä tai yhteiset tavoitteet. Lisäksi ryhmän jäsenten täytyy tuntee, että heistä muodostuu ryhmä ennen kuin jotakin ihmisjoukkoa voidaan kutsua ryhmäksi [Juu92]."

Tiimiorganisaatioita on alettu suosimaan, koska ryhmässä työskentelyn on havaittu motivoivan ja tukevan oppimista. Vaikka perinteiset linjaorganisaatiot mahdollistavat perinteisten työryhmien olemassaolon, linjaorganisaatioista on siirrytty hiljalleen tiimityöhön, koska tiimityöskentelystä on saatu huomattavia etuja. Tiimityöskentelyssä lähtökohtana on tiimipohjainen organisaatio. Työn tuottavuus on parantunut, koska työprosessit ovat järkeistyneet, kun päällekkäisiä työtehtäviä on karsiutunut ja koska työmotivaatio on parantunut. Motivaation nousun ensisijaisia vaikuttajia ovat *mahdollisuus vaikuttaa omaan työhön (=osallistaminen)* sekä *sosiaalisen tuen mahdollisuus*. [VLK98]

Tiimin muodostaminen on haasteellinen tehtävä. Projektipäälliköllä on useimmiten vastuullaan valita tiimiin henkilöt, jotka kykenevät yhteistyöhön ja suoriutuvat annetusta työtehtävästä yhdessä. Tiiminvetäjän tulee tietää, mitä kukakin osaa ja hänellä tulee olla

taitoja tunnistaa ihmisten välisiä suhteita. Lisäksi hänen tulee toimia tiimin yhdistäjänä ja motivoijana.

5.2.1 Ryhmän kehitysvaiheet

Jotta ryhmä toimisi tehokkaasti, pyritään ryhmän *kiinteyteen*. Pienistä ryhmistä tulee kiinteitä todennäköisemmin kuin suurista ryhmistä. Jos ryhmän koko on yli 10 henkilöä, muodostuu ryhmän sisälle helposti uusia ryhmiä. Ryhmästä tulee todennäköisemmin kiinteä, jos sen jäsenet ovat toisistaan riippuvaisia yhteisten päämäärien saavuttamiseksi. Myös ryhmän menestyminen lisää kiinteyttä, koska se saa osakseen arvostusta. [Juu92]

Ryhmän kehitysvaiheet vaikuttavat ryhmän *kypsytyteen*. Kypsän ryhmän jäsenet hyväksyvät toistensa vahvuudet ilman kateutta. Heidän välillään voi olla ristiriitoja, mutta ne keskittyvät lähinnä päämääriin eivätkä ryhmän jäsenten välisiin rooleihin tai suhteisiin. Ryhmän kypsyys voi kehittyä orientoitumisen, ristiriidan, kiinteyden, harhakuvitelman ja ”kuplan puhkeamisen” kautta hyväksymiseen. [Juu92]

Orientoitumisvaiheessa ryhmän jäsenet kyseenalaistavat tehtäviä ja koettelivat rajoista. Rajojen koettelu voi johtaa *ristiriitoihin* ryhmän sisällä. Näiden ristiriitojen selvittäminen johtaa ryhmän *kiinteyteen* ja ryhmän rakenteen selvittämiseen. Tästä syntyy ”*onnellisen perheen*” *idylli (happy family -tunne)*. Tässä *harhakuvitelman* vaiheessa on tyypillistä kaikkivoipaisuuden tunne, jolloin kuvitellaan ettei mitään ristiriitoja enää ole. Kun harhakuvitelma ”onnellisesta perheestä” puhkeaa, voi ryhmä hajaantua osaryhmiin, jolloin kiinteyden häviää tai ryhmä voi kehittyä hyväksymisvaiheeseen. *Hyväksymisen* jälkeen ryhmän jäsenten ei tarvitse enää näytellä, vaan he pystyvät näkemään toistensa erilaisuudet ja arvostamaan niitä. Tässä vaiheessa ryhmä toimii tehokkaasti ja myönteisessä ilmapiirissä. [Juu92]

Tiimi kokee oman ryhmänsä kiinteyden voimakkaampana, jos ryhmää uhkaa jokin ulkoinen tekijä, esimerkiksi toinen ryhmä. *Kilpailutilanne toisen ryhmän kanssa* voi johtaa konfliktiin, mikä voi katkaista jopa aiemmin solmittuja ystävyyssuhteita. Erilaisia konfliktitilanteita, klikkejä ja epävirallisia liittoutumia syntyy eniten epävarmoissa muutostilanteissa. [Ter94] Selvittämättömät konfliktit haittaavat työn sujumista.

Ryhmäajattelu voi myös rajoittaa tiimin voimavarojen käyttöä. Jos on liian hyvä ryhmähenki saattaa olla, että keskitytään liian harvoihin vaihtoehtoihin tai pysytään kerran valituissa kannoissa tai toimintatavoissa, eikä kyseenalaisteta omaa toimintaa, jolloin

luovuus sekä ryhmän itsekritiikki häviää. Ilman yhteishenkeäkin ryhmän ja sosiaalisten odotusten vaikutuksesta voidaan olla *yhtä mieltä ryhmän kanssa*, vaikka henkilökohtaisesti tiedettäisiin asioiden olevan eri tavalla. [Pir95]

5.2.2 Ryhmä vai tiimi?

Puhutaan useimmiten *tiimistä*, kun ryhmä perustetaan *kehittämään* jotakin uutta tai ratkaisemaan jokin *rajattu tehtävä* [Ter94]. Tiimille ominaista on yhteisvastuullisuus ja itseohjautuvuus. Yhteisvastuullisuus lisää myös yksilön vastuuta. Yksilön työsuoritukset tulee nähdä osana kokonaisuutta. Itseohjautuvuus tarkoittaa ryhmän aktiivista suuntautumista kohti päämääriä ilman ylemmältä taholta tulevia kehoituksia. [Pir95]

Ryhmän jäsenten *erilaisuuden hyödyntäminen* tekee tiimistä kehittyneemmän version perinteisistä työryhmistä [Pir95]. Tiimi on lisäksi *tavoitekeskeinen* ryhmä. Tiimi ymmärretään yleisesti myös *myönteissävytteisenä ryhmänä*, koska se tarjoaa sosiaalisen tuen, hyväksynnän ja sitoutumisen mahdollisuuden. Tiimi tarjoaa mahdollisuuden kehitykseen, helpottaa oppimista, tehostaa tiedonkulkua sekä mahdollistaa myös liittymistarpeiden tyydyttämiseen. [Ter94]

5.2.3 Erilaisuuden hyödyntäminen

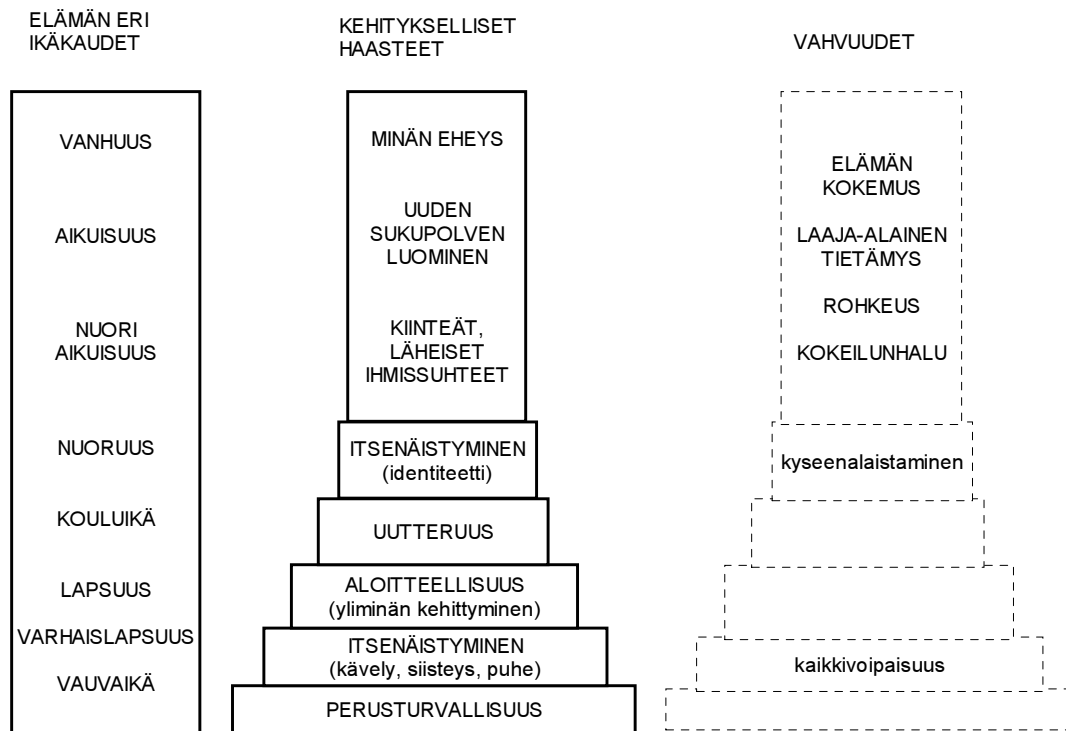
Kaikki ihmiset eivät halua olla projektipäälliköitä, mutta joiltakin se sujuu luonnostaan ja toisilta pienen harjoittelun jälkeen. Toiset ihmiset haluavat puurtaa enimmäkseen yksin, kun toiset taas haluavat aina tehdä yhteistyötä. Ohjelmistotuotannossa tarvitaan monenlaisia ihmisiä. Prosessin läpiviemiseksi tarvitaan erilaisia ammattilaisia, kuten suunnittelijoita, koodaajia, testaajia tai kouluttajia, mutta lisäksi tarvitaan erilaisilla luonteenpiirteillä varustettuja henkilöitä kuten: ”joukkuepelaajia”, mietiskelijöitä, harkitsijoita, skeptikkoja, toimeenpanijoita, rohkaisijoita, tasapainottajia, kompromissintekijöitä, ”sillanrakentajia” ja havainnoijia [Juu92].

Erilaisuus muodostuu *yksilöllisistä* eroista. Kaksosetkaan eivät ole samanlaisia eivätkä käyttäydy samalla tavalla, vaikka heillä voi olla paljon yhteisiä selittäviä tekijöitä. Erilaisuutta selittäviä tekijöitä on todella runsaasti: temperamentti, ikä, sukupuoli, kasvatus, koulutus, perhetausta, sukupolvi, biologiset ja kulttuuritekijät, elämäkokemukset, asioiden merkityksellisyys itselle, motivaatio käsiteltävää asiaa kohtaan, elämäkatsomus ja arvot.

Yksinkertaistaen ihmisiä voidaan luokitella *temperamenttityylin* mukaan. Toiset ihmiset ovat sosiaalisempia ulospäin suuntautuneita *ekstrovertejä*. Toiset taas sisäänpäin suuntautuneempia *introvertejä*. Ekstrovertejä voidaan kuvata sanoilla: sosiaalinen, puhelias, levoton, huoleton, impulsiivinen, aktiivinen, optimistinen, herkkä, aggressiivinen ja konstruktiiivinen. Introvertejä voidaan kuvata sanoilla: hiljainen, mietiskelijä, maltillinen, tasainen, luotettava, ahdistunut, varautunut, passiivinen, mukautuva ja harkitseva. Ekstrovertillä on johtajantaitoa ja introvertti sopii ehkä paremmin alaisen rooliin. [Dun98]

Ekstrovertti tai introvertti temperamenttityyli ei sulje kuitenkaan toista tyyliä pois, eikä aina ole tarkoituksen mukaistakaan luokitella ihmisiä tällä tavoin. Tietynlaiset luokittelut voivat kuitenkin auttaa ymmärtämään ihmisten välille syntyviä *henkilökemioita*. Ihmisten erilaisuudelle on yleensä järkeenkäyvät selitykset. [Dun98]

Yksi erilaisuuteen vaikuttava tekijä on *ikä*, sillä ihmiset ovat eri ikäisinä "selvinneet" tietyistä kehitystehtävistä. Eriksonin teorian mukaan ihminen joutuu selvittämään tietyt kriisit elämässään. Erik H. Erikson (s. 1902) oli psykoanalyttikko, jonka oppeja opetetaan edelleen. Riippuen siitä, miten ihminen onnistuu vastaamaan elämänkaaren vaiheisiin liittyviin kehityshaasteisiin ja mahdollisiin kriiseihin, hän kykenee jatkossa erilaisiin asioihin (ks. kuva 17). Selvittämättömän kehitystehtävän voi käydä läpi myös myöhemmin. Esimerkiksi jos aikuisuudessa huomaa, että itsenäistyminen on jäänyt nuoruudessa kesken, sitä voi vielä "paikkailla". [VuT92]



Kuva 17: Elämänskaaren kehityshaasteet Eriksonin mukaan mukailtuna lähteestä [VuT92]

Elämänskaaren eri vaiheisiin kuuluu tyypillisesti myös erilaiset *vahvuudet*. Nämä on esitelty kuvassa 17. Lapsuuteen kuuluu *kaikkivoipaisuuden* kuvitelma, joka on hyödyllinen. Ilman sitä lapsi ei esimerkiksi oppisi kävelemään, koska se tuntuisi liian vaikealta. Kaikkivoipaisuuden tunne voi esiintyä myös työelämässä, jolloin se ei ole pelkästään myönteinen. Kaikkivoipaisuus tunnetaan myös nimellä *narsismi* [Juu95].

Nuoruuteen liittyy usein erilaisten asioiden *kyseenalaistaminen*, joka liittyy oman identiteetin etsintään. Kun nuori miettii esimerkiksi "onko tuo tyyli minulle sopiva", hän rakentaa identiteettiään, esimerkiksi ammatillista identiteettiä. Identiteetin etsintä kuuluu myös ryhmän muotoutumiseen. Ryhmälle kehittyy yhteinen kollektiivinen totuus siitä, "mitä me olemme" eli millainen on ryhmän identiteetti.

Jatkuvaa kyseenalaistamista ei saa unohtaa myöskään työelämässä, mutta on hyvä pitää joitakin perustavaa laatua olevia asioita selvyyksinä eikä kyseenalaistaa jatkuvasti esimerkiksi yrityksen perusarvoja. Nuori aikuinen on *kokeilunhaluinen* sekä valmis uusiin ideoihin. Iän myötä asioita nähdään *laaja-alaisemmin*. Nuoruuden hurjatkin ideat voivat onnistua erinomaisesti, jos ne nähdään voimavarana, jota kokeneemmat henkilöt voivat

ohjata oikeaan suuntaan. Nuori saattaa innoissaan ehdottaa asiaa, jota vanhemmat työntekijät ovat olleet jo muutamia kertoja yrittämässä. Tilanteesta voi tulla konflikti, jos osapuolet eivät pysty ymmärtämään toisiaan.

Kaikkein pisimpään työelämässä olleet tuovat yritykseen *elämäkokemusta*. He tietävät asioista eniten, ja siksi he osaavat suhtautua uusiin ideoihin riittävällä kriittisyydellä, mutta he voivat olla myös urautuneita omiin toimintamalleihinsa. Kokeneilla työntekijöillä on yleensä myös kehittyneemmät ongelmanratkaisutaidot, joista nuoremmat voivat ottaa oppia. Nämä seikat huomioiden yhteistyöllä päästään parhaimpiin tuloksiin. Kaikenlainen *rasismi* on siis vain haitallista.

5.2.4 Erilaisuuteen ja muutokseen suhtautuminen

Myös *erilaisuuteen* suhtaudutaan eri tavoin. Ääripäitä voisi edustaa lausahdukset: "erilaisuus on vaarallista" ja "erilaisuus on kiinnostavaa", joiden välimaastossa voisi olla sanonta: "erilaisuus on huvittavaa" [Hof93]. Erilaisuuteen suhtautuminen on sidoksissa *epävarmuuden välttämiseksi* nimettyyn tekijään. Ihmiset, jotka välttävät voimakkaasti epävarmuutta uskovat erilaisuuden olevan vaarallista. Tälle uskomukselle on loppujen lopuksi vain vähän tehtävissä, sillä se on jopa kulttuurisidonnaista. Tutkimuksessa, jossa kysyttiin mm. työn aiheuttaman stressin määrää, yrityksen sääntöjen rikkomisaikeita/mahdollisuutta ja työpaikassa pysymisaikeita, saatiin eri maiden välillä huomattavia eroja [Hof93].

Yleisesti epävarmuutta voimakkaasti välttävät tuntevat itsensä usein ahdistuneeksi ja täten myös stressaantuneeksi eivätkä mielellään riko sääntöjä, koska siitä voi seurata jotain ennalta arvaamatonta. "Ahdistus on epämääräistä levottomuutta tai huolestuneisuutta, ei siis pelkoa, jolla on yleensä kohde." Epävarmuutta välttävät henkilöt eivät myöskään suunnittele työpaikan vaihdosta, koska se tietäisi "hyppyä tuntemattomaan" [Hof93]. Epävarmuuden välttämisindeksi voi vaihdella eri maiden välillä. Tämä tarkoittaa sitä, että esimerkiksi saksalainen täsmällisyyden tavoittelu voi huvittaa brittiä ja saksalaisen mielestä britit suhtautuvat asioihin liian löyhästi [Hof93].

Erilaisuutemme johtuu siis osittain yhteiskunnallisista tekijöistä tai kasvatuksesta. Näihin emme voi vaikuttaa. Tällaiset tekijät (kuten muutkin persoonalliset piirteet) on mahdollista ymmärtää järjen kautta, vaikka ne tuntuisivatkin kummallisilta. Ei ole olemassa oikeaa tai väärää tapaa tehdä asioita - vain erilaisia. Erilaiset tavat tulee tietenkin olla yleisesti

hyväksyttäviä muun muassa lain tai yrityksen liiketoiminnan puitteissa eli normien sallimissa rajoissa.

6 VUOROVAIKUTUSTAITOT

Ohjelmistotuotetta ei voi tehdä tyhjiössä. Vuorovaikutustaitojen tarve on ilmeinen. Vaatimukset tuotteelle saadaan vuorovaikutuksessa asiakkaan kanssa. Asiakkaan liiketoimintaidean ja –prosessien ymmärtäminen on tärkeää. Tiimissä toimivat asiantuntijat kommunikoivat keskenään, jotta esimerkiksi tietoliikenne, laitteistot ja ohjelmistot voivat toimia yhdessä. Usean henkilön yhteistyötä tarvitaan esimerkiksi silloin, kun [Haw01]:

- kehitetään uusia ideoita (ideariihi),
- yritetään ymmärtää, mistä on kysymys,
- analysoidaan ja tarkennetaan vaatimuksia,
- ideoita testataan,
- päätetään, mitä tehdään ja
- kun osista kootaan kokonaisuuksia.

Itseluottamus ja *itsearvostus* ovat pohjana moniin vuorovaikutustaitoihin, kuten kritiikin vastaanottamiseen. Itseluottamus on luottamusta omaan itseen sekä uskoa omiin kykyihin ja mahdollisuuksiin. Ajattelutapana: "Mitä en vielä osaa, sen voin oppia." Itsearvostus on sitä, että arvostaa itseään, elämäänsä ja esimerkiksi työpanostaan, vaikkei kaikista asioista pitäisikään [Hak95].

On hyödyllistä muistaa, että *myönteisyys* synnyttää myönteisyyttä ja *ratkaisukeskeisyys* on ongelmakeskeisyyden vastakohta. Ratkaisukeskeisyys tarkoittaa sitä, että keskitytään ratkaisumahdollisuuksiin eikä siihen, mikä kaikki voi mennä pieleen. Ohjelmistotuotannossa puhutaankin *ongelmanratkaisusyklistä (problem-solving cycle)*. Tämä tarkoittaa tiettyjen vaiheiden kokonaisuutta, jonka avulla ratkaistaan, miten vaatimusmäärittelyistä johdetaan määrittelyjen mukainen ohjelmisto [Haw01]. Projektin hallinnan kannalta on tosin tärkeää nähdä myös etukäteen, mikä kaikki voi mennä pieleen, jotta pystytään mahdollisimman hyvään riskien hallintaa (ks. Kappale 5.1.3 Riskien hallinta).

Edellytys ryhmälle on, että sen sisällä on vuorovaikutusta. Ryhmässä tulee kertoa asioista. Henkilökohtaisten asioiden kertominen tuo yhteishenkeä, muttei ole välttämätöntä. Tässä voi olla hankalaa vetää raja avoimuudelle. Voi olla hyvä pitää intiimit asiat työkavereilta

salassa, jottei tule turhaa kateutta tai sääliä. Toisaalta koko ryhmän ei tarvitse tietää toisistaan samoja asioita ja voi olla helpottavaa, että työpaikalla on edes joku, jolle voi kertoa todellisista tuntemuksistaan niin työtä kuin henkilökohtaisiakin asioita kohtaan. Työhön liittyvät asiat tulisi aina *kertoa*. Jos ei uskalleta puhua asioista, toiminta menettää tehonsa. Tulisi tuntea sen verran vastuullisuutta, että kertoo esimerkiksi projektin todellisen tilanteen sitä kysyttäessä tai kysymättäkin, jos tilanne sitä vaatii eli esimerkiksi, jos projekti on umpikujassa ja pahasti myöhässä [Tie02].

Vuorovaikutustaitoihin kuuluu myös *luottamus*. Se on monenlaisia pikku asioita. Se on esimerkiksi apua tarvitsevan tukemista ja sitä, että voi luottaa saavansa apua. Luottamuksen osoittamista työtoverille on esimerkiksi, että antaa hänelle vastuuta jonkin tehtävän suorittamiseksi eikä jatkuvasti valvo tai tarkista perässä, tuliko tehtävä varmasti hoidetuksi ja juuri niin kuin itse sen tekisi. Luottamus saavutetaan aina hitaasti, mutta sen menettämien tapahtuu useimmiten nopeasti [Hak95].

6.1 VIESTINTÄ

Vuorovaikutus on ennen kaikkea *keskustelua*, mikä tarkoittaa, että kaikki saavat osallistua. Keskustelutilanteessa tulee osata esittää ”oikeat kysymykset”. Esimerkiksi ohjelmiston määrittelijän, joka haastattelee asiakasta on tärkeä osata muotoilla kysymyksensä riittävän tarkasti. Asiakkaan tulee ymmärtää kysymykset ja määrittelijän on pystyttävä tarkentamaan analyysiään vastausten perusteella. Haastattelijan on myös muistettava esittää asiansa niin, että hän on täällä auttamassa asiakasta tämän ongelmien ratkaisemisessa eikä hän suinkaan ole esimerkiksi etsimässä virheitä tai arvostelemassa asiakkaan toimintamallia [Haw01].

Asioiden *kysyminen* osoittaa, että kysyjä on kiinnostunut asiasta ja ihmisistä. Kannattaa siis miettiä, kuinka kysymyksensä asettaa. Motivoivat kysymykset aloitetaan Hagemanin mukaan kuka-, mitä-, miten-, mikä- ja milloin-sanoilla. *Miksi*-sanaa kannattaa välttää, sillä se voi sisältää ”syytöksen”. Esimerkiksi lause: ”Miksi teet työsi tässä järjestyksessä?” voi sisältää merkityksiä: ”Eikö kannattaisi tehdä toisin” tai ”Etkö osaa tehdä järkevämmiin?” [Hag91].

Keskustelun toinen tärkeä puoli puhumisen lisäksi on *kuunteleminen*. Hyvin muotoilluista kysymyksistäkään ei ole hyötyä, jos ei keskity myös kuuntelemaan. Kysyjä voi kysyä

myös aivan vääriä kysymyksiä, jos hän kuuntelemisen sijaan keskittyy vain seuraavaan kysymykseen [Hag91].

Kuuntelemisen taitoaan voi arvioida kysymällä itseltään [Hak95]:

- ”Onko minun vaikea keskittyä siihen, mitä toinen sanoo?
- Olenko kohtelias kuunnellessani toista?
- Löydänpö toisen sanomasta pääasian?
- Osaanko olla tekemättä asiaankuulumattomia huomautuksia?
- Saanko selville, mitä toinen tarkoittaa?
- Muistanko, mitä toinen on sanonut?
- Olenko kiinnostunut, osoitanko kiinnostuneisuutta?”

On hyvä muistaa vuorovaikutuksesta, että *hiljaisuus* on ääneen lausumattomia ajatuksia kenties tunteitakin, joten hiljaisuudella voi olla suurikin vaikutus ja sitäkin pitäisi osata "kuunnella". Vaikeneminen onkin suomalaisille ominaista. Suomalaiset harkitsevat tarkoin sanojaan; hiljaisuus ei siis suoraan tarkoita, ettei vastapuolella ole mitään sanottavaa.

Huumori voi ratkaista vaikeissa vuorovaikutustilanteissa. Aito, ketään loukkaamaton huumori on ryhmälle voimanlähde, jonka avulla voi laukaista jännitteitä. Ristiriitoja saa toki olla, mutta silloin tulisi keskittyä niiden ratkaisemiseen eikä syypäiden etsimiseen. Kannattaa ottaa vastuuta itselle, sillä toista ei yleensä voi muuttaa. Tiimin toiminta helpottuu, kun arvostaa toista ihmistä itsensä lisäksi.

Jos haluaa parantaa yhteisön, organisaation tai tiimin, ilmapiiriä, voi yrittää sanoa jotain positiivista ihmisestä, jota moititaan. Se voi auttaa toisia ihmisiä näkemään tämän ihmisen hyvät puolet. Yleensä on itsestä kiinni, miten näkee toisen ihmisen. Ennakkoluuloista pitää päästää irti ja antaa jokaiselle mahdollisuus olla hyvä työtoveri [Hak95].

Viesti vääristyy usein kulkiessaan lähettäjältä vastaanottajalle. Näin tapahtuu, koska *lähettäjän* täytyy ensin muotoilla *viesti* lähetettävään muotoon ja *vastaanottajan* täytyy tulkita saapunut viesti. Ensimmäinen vääristymä voi tapahtua viestiä muotoiltaessa. Tähän ongelmaan toki auttaa, jos viestin lähettäjä on taitava viestijä. Seuraavaksi vastaanottajan tulee ”kuulla” viesti oikein sekä ymmärtää sen sisältö. Lähettäjän ja vastaanottajan ajatukset eivät kuitenkaan voi olla täysin yhtenevät, jolloin väärinymmärrys on varsin inhimillistä. Eroavaisuuksia voi olla muun muassa henkilöiden kasvatuksessa,

koulutuksessa, maailmankuvissa, kognitiivisissa taidoissa tai sen hetkisissä orientaatioissa. Tilannetta voi helpottaa, jos vastaanottaja on hyvä kuuntelija ja hänellä on mahdollisuus esittää tarkentavia kysymyksiä. Viestin perillemeno voi vaikeuttaa myös rauhaton ympäristö tai jonkun viestijän turvattomuuden tunne. [Juu92]

6.1.1 Sanaton viestintä

Sanaton viestintä on osa kommunikointia ja vuorovaikutusta. Ihminen luottaa enemmän *sanattomaan viestintään* eli äänensävyihin, asentoihin, eleisiin ja ilmeisiin kuin sanoihin. Jos sanat ja eleet ovat tasapainossa ei ole väärinymmärryksen pelkoa, mutta jos eleet kertovat jotain muuta kuin sanat, niin vastaanottava osapuoli uskoo toisen valehtelevan. Eleet ja ilmeet tulevat ihmiseltä vaistomaisesti ja siksi niillä on vaikeampi huijata kuin sanoilla.

Se, että ihmiset luottavat enemmän eleisiin kuin sanoihin, mahdollistaa myös vitsien ymmärtämisen. On hyvä muistaa, että useimmiten vitsit tarvitsevat vuorovaikutusmuodon, missä väärinymmärryksen vaaraa ei ole. Esimerkiksi vitsien lähettäminen sähköpostitse on taituruutta vaativa laji. Eleitä, joiden avulla osataan tulkita viestejä, opitaan iän myötä lukemaan. Lapset uskovatkin sanoja ennemmin kuin eleitä, joten he ottavat vitsitkin konkreettisesti. Jokaisen ”vitsinkertojan” kannattaa muistaa, että kaikki aikuisetkaan eivät ole aina ”samalla aaltopituudella” [ToK98].

6.1.2 Manipulaatio

Sanaton viestintä on "hyvä" keino ihmisten *manipulointiin*. Taitavat manipuloijat hallitsevat kehon kielen. Yksinkertainen esimerkki on kokous, jossa istutaan pöydän ympärillä. Jos haluaa vaikuttaa kiinnostuneelta, vaikkei oikeasti olisikaan, voi nojautua eteenpäin, katsoa puhujaan ja kuunnella. Jos taas haluaa vaikuttaa ylimieliseltä, voi nojautua taakse päin tuolissa, laittaa kädet puuskaan ja katsella ulos ikkunasta [RuS98].

Esimiehellä voi olla vaikeuksia saada alainen yhteistyöhön. Tällöin tehtävänä on osoittaa, että olemme menossa *samaan suuntaan* [ToK98]. Tämä manipulaatio toimii myös alaisen tekemänä. Ikäviä manipulointitapoja ovat huijaus ja asioiden salaaminen. Yleensä nämä paljastuvat myöhemmin ja saavat aikaan luottamuksen menetyksen. Sekä alainen että esimies voi osoittaa toiselle, että olemme loppujen lopuksi menossa samaan suuntaan ilman ikävää manipulointia, jos vain tahtoa, taitoa ja aikaa riittää. Toisen kuunteleminen ja

toisen mielipiteen arvostaminen voi konkretisoitua seuraavanlaisiin lauserakenteiseen: "...mikäli ymmärsin oikein, tarkoituksesi oli...", "arvostan näkemystäsi ja olen varma, että sinäkin ymmärrät minun..." tai "olet oikeassa siinä, että... mutta...". Yleensä ihminen hyväksyy ikävätkin asiat, jos ne perustellaan oikein ja totuudenmukaisesti. [Hag91], [ToK98]

NLP (Neuro Linguistic Programming) [ToK98] on eräs keino ”mielen ohjelmointiin”. NLP:ssä puhutaan *samauttamisesta*, joka voi olla myös manipuloinnin keino. Tämä tarkoittaa esimerkiksi edellä esiteltyjen lauserakenteiden käyttämistä. Muita sanattoman viestinnän keinoja ovat mm. katsekontakti, äänenvoimakkuus ja kehon kieli. Samauttamisen avulla voidaan vuorovaikutustilanteesta tehdä myös miellyttävämpi. Esimerkiksi samalle puolelle pöytää istuminen vastakkainasettelun sijaan voi toimia hyvin neuvottelutilanteessa. Samauttamista on myös esimerkiksi samanlaisen äänenvoimakkuuden löytäminen neuvottelijoiden kesken. Joskus on syytä samauttaa jopa sanojen tasolla. Jos allapäin oleva henkilö sanoo olevansa ”murtunut”, ei pidä sanoa ”ymmärrän surusi” tai ”eihän sinulla ole mitään aihetta” kuin tietäisi, mistä on kyse, vaan kysyä enemminkin ”mikä sai sinut murtumaan?” Kyse ei nimittäin välttämättä ole surusta, vaan esimerkiksi masennuksesta. Tärkeintä on ottaa huomioon, miten toinen kokee asiansa [CoD98]. Tunteisiin palataan kappaleessa 6.3 Tunneäly.

Vuorovaikutustaitojen *kehittämiseen* NLP tarjoaa myös erilaisten näkökulmien tiedostamisen. Tämä on keino itsenä manipulointiin. Eri näkökulmien tiedostamiseksi voi ajatella ensin jotakin tilannetta, jossa keskustelee jonkun henkilön kanssa ja miettiä sen jälkeen, mistä suunnasta katsoi tätä tilannetta. Oliko se minä-, toisen vai tarkkailijan näkökulma. Tilannetta voi käydä uudelleen läpi pyrkien käymään noissa kaikissa näkökulmissa. Minä-näkökulmasta löytää omat tavoitteensa eli pystyy paremmin perustelemaan oman kantansa, toisen näkökulmasta ymmärtää paremmin toisen tarkoitusperiä ja tarkkailijana voi keskittyä neutraalisti tapahtumien kulkuun. [ToK98]

6.1.3 Palaute

Kun työskennellään organisaatiossa, tiimeissä ja projekteissa, on sekä yksilölle että ryhmälle tärkeää *saada palautetta* työstään. Vain palautetta antamalla ja saamalla toiminta voi kehittyä haluttuun suuntaan. Esimerkiksi esimiehen tulee antaa palautetta alaisilleen, alaisen esimiehelleen, tiiminjäsenen toiselle tiiminjäsenelle ja projektiryhmän

asiantuntijatiimille. Lisäksi ohjelmistotuotantoon liittyy paljon erilaisia tarkastustilaisuuksia, katselmuksia sekä kehitystilaisuuksia, joissa on tärkeää osata antaa palautetta ja ottaa sitä vastaan niin, että tilaisuuksista on hyötyä.

Olisi uskallettava esittää kriittiset ajatuksensa oikeassa paikassa, koska selän takana puhuminen ei auta asioiden kehittämisessä. Asiat on hoidettava suoraan ja välikäsiä tulisi välttää. Palautteen tulisi koitua *vastaanottajan eduksi* ja ohjaukseksi. Tärkeää palautteen antamisessa on välittömyys. Palaute pitäisi antaa mahdollisimman pian suorituksen jälkeen. Palaute täytyy aina myös perustella; pelkät moitteet tai kehu ei hyödytä mitään. Palautteessa pitäisi aina olla myös positiivinen puoli ja keskiarvoisesti positiivista palautetta tulisi antaa enemmän kuin negatiivista [Hak95]. Positiivista puolta voi korostaa jopa lauseeseen valittavilla sanoilla. Esimerkiksi pronomini *tämä* on läheisempi kuin *tuo*. Lisäksi viimeksi sanottu jää paremmin mieleen. Siksi palautelause tulisi muotoilla esimerkiksi näin: "Tuo kohta ei vielä ole valmis, mutta tämä tässä on oikein hyvin tehty." (Vrt. "Tuo on oikein, mutta tämä tässä on todella pielessä.") [ToK98].

Myös positiivista palautetta täytyy osata antaa. Kiitä, kun siihen on aihetta. Ota kiitos vastaan, äläkä vähättele saavutuksiasi. Kannustettu ja rohkaistu kannustaa mielellään toisia ja siten edistää yhteishengen kehittymistä ryhmässä. Yhteishenki on tärkeää haasteiden ja vaikeiden asioiden edessä. Toista ihmistä tulee arvostaa, mikä ei tarkoita kuitenkaan jatkuvaa kehumista. Kun pitää toista vertaisenaan ja ottaa hänen ajatuksensa tosissaan, se on arvostusta [Hak95].

Tulee osata ottaa vastaan kiitos ja myös kritiikki. Siitä voi oppia jotakin. Palautteen *vastaanottamisessa* taidokkuutta on kyky nähdä palautteen merkitys itselle ja hyödyntää se.

Isoherranen antaa [Iso91] seuraavanlaisia palautteenanto-ohjeita, lähteestä [Hak95] muotoiltuna:

”Kun annat palautetta:

- Kuvaa omia tunteitasi ja ajatuksiasi, älä arvioi toista.
- Anna palaute tämänhetkisistä työn tuloksista; älä yleistä tai muistele menneitä.
- Anna palaute näkyvästä työn tuloksesta, älä henkilöstä.
- Kerro mitä tapahtui, älä arvioi toista hyvä-paha -ulottuvuudella
- Anna tietoa ja ideoita; älä neuvo.

- Keskity työn tuloksiin, joita voi muuttaa; älä kuvaa toisen luonnetta.
- Ajoita palaute oikein; mieluummin heti tapahtuman jälkeen.
- Ole aito ja vilpitön niin myönteisessä kuin kielteisessäkin palautteessa. ”

*Suunnittelumallien (patterns) kehittämässä voidaan käyttää katselmustilaisuuksia, joissa palautteen antaminen on melko tarkasti määritelty. Suunnittelumallin tekijä istuu sivussa ja palautteenantajat ympärässä. Tilaisuus alkaa yleiskatsauksella kyseessä olevaan suunnittelumalliin. Palautteenantajat aloittavat keskustelun suunnittelumallin *positiivisista ominaisuuksista*, jonka jälkeen he voivat vuorollaan esittää parannusehdotuksia, ei siis virheitä tai puutteita vaan *konkreettisia ja rakentavia parannusehdotuksia*. Tämän jälkeen suunnittelumallin tekijä tulee kehän keskelle, jotta hän voi kysellä tarkennuksia parannusehdotuksiin. Tilaisuus päättyy jonkun palautteenantajan *positiivisen yhteenvedon*. [Cop99]*

Palautetta on vaikea ottaa vastaan, jos siinä ei ole mitään positiivista. Edellä esitetyllä palautteenantokaavalla päästään palautteessa kokonaisuudessaan positiiviselle puolelle, jolloin palautteen saaja hyötyy eniten ja on motivoitunut jatkamaan työtään.

6.2 TYÖVÄLINEIDEN VAIKUTUS

Ohjelmistotuotannossa tehdään paljon töitä tietokoneella ja sitä kautta voi tulla tietynlaisia odotuksia myös suhteessa ihmisiin. Ohjelmistotyön luonnetta voisi kuvata kiireiseksi, uutta luovaksi ja vaativaksi. Ihmiset eivät ole kuitenkaan koneita; kukaan ei voi odottaa toiselta suoritusta tietyssä suoritusajassa eikä kukaan voi vaatia aina saavansa haluamansa vastauksen.

Koneiden välityksellä työskentely eli esimerkiksi sähköpostin välityksellä kommunikoinen tuo mukanaan sanattomien viestien puutteen, mistä johtuen väärinymmärryksen mahdollisuus lisääntyy. Vitsiksi tarkoitettu huomautus voi loukata vastaanottajaa, koska hän ei näe lähettäjän ilmettä. Interaktiivisuuden puute voi tehdä viestistä ikävämmän kuin kasvokkain, koska viestiä ei voi mukauttaa toisen reaktioiden mukaisesti. Siksi palaute voi olla hyvinkin erilaista kuin kasvokkain.

Toisaalta on hyvin tärkeää osata kommunikoida myös kirjallisesti ja erilasten sähköisten viestimien kautta. Tiimityötä ja projektinhallintaa tukevat tänä päivänä ohjelmistot, joita

pitää osata käyttää ja ”lukea”. Tiimin jäsenet voivat työskennellä maantieteellisesti kaukana toisistaan, jolloin he kommunikoivat puhelimitse, videoneuvotteluin, sähköpostin välityksellä ja erilaisten intranet- ja internetverkkojen välityksellä.

Ohjelmistotuotannossa on tärkeää kirjata asioita ylös, dokumentoida. Dokumentteja käytetään kommunikoinnin välineenä kehitys- ja ylläpitotyössä eri asiantuntijoiden välillä tuotteen ymmärtämiseksi. Vaatimusmäärittelydokumentit ovat mm. testauksen tukena. Testausraportit puolestaan todistavat ohjelmiston riittävän virheettömyyden ja mahdollistavat regressiotestauksen. Dokumentteja tuotetaan samalla kuin tuotettakin. Uudet apuvälineet luovat osittaista dokumentaatiota samalla, kun suunnittelu etenee. Tämä onkin *CASE-välineiden (Computer Assisted Software Engineering)* käytön suurimpia hyötyjä [Haw01].

6.3 TUNNEÄLY

Sosiaalisten taitojen kenttään liitetään nykyisin käsite *tunneäly (Emotional Intelligence, EI)*. Kokonaisuus ratkaisee työelämässä, ja elämässä yleensä, pärjäämisen. Pelkkä tietotaito ja älykkyys työelämässä ei näytä enää riittävän. Tunneäly on vuorovaikutustaitoja parhaimmillaan ja hieman syvällisemminkin. Se ei ole pelkästään esimerkiksi kuuntelemisen taitoa vaan myös ymmärtämisen taitoa. [Car02]

Vasta 1990-luvulla on alettu hyväksyä ajatus tunneälystä tarpeellisena taitona älykkyysosamäärän lisäksi. Ajatus tunteiden järkeistämisestä ei ole kuitenkaan uusi. Jo 1920-luvulla psykologi E. L. Thorndike esitti, että *sosiaalinen älykkyys* eli kyky ymmärtää muita, olisi yksi osa ihmisen älykkyysosamäärää [Gol95]. Edelleen tarvitaan toki myös riittävä älykkyys, jotta pärjää työelämässä [Car99]. Ihmiset ovat kuitenkin yleisesti älykkäitä ja teknisesti taitavia omalla alallaan, vaikka tunneäly voi puuttua. Siksi tunneällyn painotusta tarvitaan työelämässä.

Tunneällyn käsitteen loivat Jack Mayer ja Peter Salovey 1990. Tieteellisissä julkaisuissaan he määrittelevät tunneällyn seuraavasti: "*Emotional intelligence is the ability to perceive emotions, to access and generate emotions so as to assist thought, to understand emotions and emotional knowledge, and to reflectively regulate emotions so as to promote emotional and intellectual growth.*" [Car02]

Mayerin ja Saloveyn määritelmässä on neljä tasoa [Car02]:

1. *tunteiden tunnistamisen (identify),*
2. *tunteiden ilmaisemisen (use),*
3. *tunteiden ymmärtämisen (understand) ja*
4. *tunteiden säätelyn (manage).*

Tunneällyn käsitteen on tehnyt tunnetuksi Daniel Goleman kirjoillaan ”*Emotional Intelligence*” [Gol95] ja ”*Working with Emotional Intelligence*” [Gol98]. Goleman määrittelee tunneällyn: *Omien tunteiden tunnistamiseksi, tunteiden hallitsemiseksi, motivaation löytämiseksi, muiden tunteiden havaitsemiseksi ja ihmissuhteiden hoitamiseksi* [Car02], [Gol95]. Mayerin & Saloveyn ja Golemanin määritelmät eivät ole yhtäläiset, mutteivät myöskään kovin erilaiset. Mayerin ja Saloveyn tutkimukset ovat olleet Golemanin kirjojen lähteinä monien muiden lähteiden lisäksi.

Tunneäly vaikuttaa työyhteisössä *ryhmien* kiinteyteen sekä yhteisten päämäärien löytämiseen ja niihin sitoutumiseen. Lisäksi tunneäly auttaa tehtävissä suoriutumiseen ja arviointiin sekä epävarmuuden sietoon [Gol98]. Tunneäly voi auttaa tiimiä luovuuteen, koska yhteistyö sujuu ja näin saadaan useita näkökulmia ja voidaan nähdä uusia ratkaisuja. Tunneäly auttaa rakentamaan yhteisymmärrystä, uskottavuutta, luottamusta ja empatiaa [Car99]. *Johtajille* tunneällyn käyttö tarkoittaa joustavien suunnitelmien tekemistä, motivoinnin taitoja sekä päätösten tekemistä tunteet huomioiden. [Car99]

6.3.1 Tunteet

Tunteiden tunnistaminen on ensimmäinen askel kohti tunneälykkyyttä [Car99]. Työpaikallakin ihmisellä on tunteet eikä kukaan voi niitä poistaa sanomalla, ettei ole [Juu95]. Kukaan ei jaksa aina hymyillä ja olla hyväntuulinen. Ihmisen tunteiden kirjo on moninainen. Jotta voimme tunnistaa tunteita, niillä täytyy olla nimet.

Perustunteet ovat: suru, viha, pelko, nautinto, rakkaus, hämmästys, inho ja häpeä. Lisäksi voidaan erottaa muun muassa ilo, onnellisuus, pettymys, ahdistus, turhautuneisuus, masentuneisuus, kateus, syyllisyys, positiivinen ja negatiivinen jännitys sekä pitkä lista erilaisia *vivahteita ja yhdistelmiä tunteista* esimerkiksi mustasukkaisuus, joka voi sisältää vihaa, surua ja pelkoa. [Gol95]

Joidenkin tutkijoiden mielestä kaikki tunteet voidaan kategorisoida perustunteiden alle. Esimerkiksi rakkauden alle voidaan sijoittaa: hyväksyminen, ystävällisyys, läheisyys, luottamus, kiltteys, omistautuminen, arvostaminen, ihailu, jumalointi, uppoutuminen ja antautuminen. [Gol95]

”Tunteet ovat elämyksellisiä mielenliikkeitä ja niihin liittyviä ajatuksia, psykologisia ja biologisia tiloja sekä yllykkeitä toimintaan” [Gol95]. Tunteiden äärlaidoilla on *mielialat*. Ne ovat vaimeampia kuin tunteet, mutta voivat kestää pidempään. Esimerkiksi ärtyisyys voi olla pitkään kestävä mieliala, mutta vihan vallassa ei kukaan jaksakaan olla vuorokausia. Lisäksi tunteet kohdistuvat yleensä johonkin, kun taas mielialat ovat kohdistumattomia. Mielialoihin voivat vaikuttaa myös *luonnetyyppit* siten, että toinen voi olla luonteeltaan esimerkiksi surumielinen ja toinen iloinen. [Gol95]

Caruso nostaa esiin [Car99] *levottomuuden (worry)*, *pelon (fear)* ja *ahdistuksen (anxiety)* tärkeinä tunnistettavina tunteina töissä. On tärkeää tunnistaa, mistä mikäkin tunnetila johtuu, koska tunteet antavat varmuutta, jota tarvitaan tehokkaaseen työskentelyyn. Jos on levoton tai vastaavasti onnellinen, toimivat nämä tunteet signaaleina siitä, että asiat menevät huonosti tai hyvin [Car99]. Tunteille on yleensä hyvä syy. Esimerkiksi jos tuote ei menekään kaupaksi tai projekti epäonnistuu, on ymmärrettävää tuntee pettymystä ja surua.

Juuti puolestaan nostaa esille [Juu95] tärkeinä tunnistettavina tunteina: turvallisuuden, ahdistuksen, kaikkivoipaisuuden, kateuden, ahneuden, kiitollisuuden, surun, masennuksen ja rakkauden.

Turvallisuus on määritelty perustarpeeksi ja suureksi motivaation lähteeksi. Turvallisuudentunteen vastakohtana voidaan kokea esimerkiksi hylkääminen. Tämä voi aiheutua esimerkiksi siitä, että ihmiset eivät kykene olemaan aitoja, jolloin heidän välilleen ei synny luottamusta. Turvattomuutta voi aiheuttaa myös se, kun tulevaisuudesta ei ole tietoa eli esimerkiksi pätkätyö tai tietämättömyys tulevista työtehtävistä. *Ahdistusta* syntyy usein muutostilanteissa. Tätä ahdistusta voidaan lieventää mm. riittäväällä tiedottamisella ja varaamalla riittävästi aikaa muutoksiin. [Juu95]

Kaikkivoipaisuus on kuvitelma siitä, että ollaan parempia kuin muut. Yksilö, ryhmä tai organisaatio voi kokea näin. Kaikkivoipaisuuden tunne voi olla tervettä, jos se ei estä kehittämistarpeiden huomaamista. Kun kaikkivoipaisuuden harha ohjaa yksilöitä, he eivät yleensä kuitenkaan näe omia heikkouksiaan eivätkä ota vastaan kehitysehdotuksia, jolloin he eivät voi myöskään kehittyä. [Juu95]

Kateus ilmentää alemmuuden- ja mitättömydentunnetta. Kadehtia voi työtoverin uraa tai toisen projektin menestystä. Kateellinen pyrkii lisäämään omaa arvoaan tai vähentämään toisen arvoa, esimerkiksi käyttäytymällä kaksinaamaisesti ja puhumalla pahaa toisista, päästäkseen samalla tasolle kateuden kohteen kanssa. Tämä ei kuitenkaan auta kuin hetken, koska perimmäinen syy kateuteen ei ole se, että toinen olisi jotenkin parempi vaan se, että kateellinen kokee itsensä riittämättömäksi. [Juu95]

Kateus voi toimia voimana kunnian tai aseman tavoitteluun, mutta pahimmillaan se voi ilmetä myös ahneutena. *Ahneus* syntyy pohjimmiltaan menettämisen pelosta. Ahne ihminen haalii itselleen ja ympärilleen asioita, joita hän ei edes tarvitse. [Juu95]

Kateuden vastakohta on *kiitollisuus*. Jos ihminen on onnistunut luomaan kiitollisen ja positiivisen suhteen toiseen henkilöön, kestää tämä suhde myös ajoittaiset kateuden ja ahneuden tunteet. Kateus voi olla pohjimmiltaan vihaa ja kiitollisuus rakkautta. [Juu95]

Rakkaus mahdollistaa toiminnan, joka ei ole pakon sanelemaa. Rakastaa voi muun muassa perhettään, työtään, työtovereitaan tai onnistumisen tunnetta. Rakkaus ilmenee ihmisten suhteissa kanssaihmiisiinsä ja töihinsä. Rakkaus mahdollistaa ymmärtävän ja empaattisen ilmapiirin. [Juu95]

6.3.2 Tunteiden tunnistaminen ja ilmaiseminen

Tunteiden tunnistaminen on *omien tunteiden* ”näkemistä” ja kuulemista, mutta myös *toisten tunteiden* tunnistamista. *Tunteiden ilmaiseminen ja käyttö* on seuraava askel tunneälyyn [Car99]. Tämä tarkoittaa mahdollisuutta esimerkiksi luovuuteen tuottamalla itselleen tietynlaisia tunnetiloja. Toisten ihmisten tunteiden kokeminen ilmenee empatian taitona.

Empatia on toisen ihmisen eläytyvää ymmärtämistä. Tämä tarkoittaa, että empaattinen ihminen jakaa toisen tunteet sallimalla itsensä tuntea jossain määrin samoja tunteita. Toista on vaikea ymmärtää kiireessä, sillä empatiaan tarvitaan aikaa ja rauhallisuutta. Empaattista ihmistä pidetään yleensä sosiaalisesti lahjakkaana. Empatia on hyökkäävän käytöksen vastakohta. Sitä voidaan opettaa kotona ja koulussa jo lapsille, muun muassa näyttämällä esimerkkiä ja toimimalla empaattisesti sosiaalisissa tilanteissa. [Kal97]

Joiltakin ihmisiltä puuttuu kyky tunnistaa edes omia tunteitaan. Tällaisesta kyvyttömyydestä käytetään nimitystä *aleksitymia*. Muita aleksitymian tuntomerkkejä ovat kyvyttömyys kuvailla omia tunteita sekä köyhä mielikuvitus. Aleksityymikko ei kykene

empatiaan ja hänen on vaikea ymmärtää toisten ihmisten tunteenpurkauksia. Hän suuntaa ajatuksensa konkreettisiin, ulkoisiin asioihin. Aleksityymikko ei kykene tunnistamaan kehonsa tuntemuksia eikä ilmaisemaan tunteitaan sanoin. Hän ei myöskään osaa lukea sanattomia viestejä kuten muut, mistä johtuen hänen voi olla vaikea pärjätä sekä ihmissuhteissaan että ihmiskeskeisessä työelämässä.

Tunteiden ymmärtäminen on kolmas askel tunneälyä. Se tarkoittaa työelämässä muun muassa sitä, että tietää, mikä motivoi ihmisiä, ymmärtää toisten ihmisten näkökulmia sekä ymmärtää ja pystyy käsittelemään tiimin keskinäisiä vuorovaikutussuhteita. [Car99]

6.3.3 Tunteiden säätely

Tunteiden tunnistaminen, käyttäminen ja ymmärtäminen mahdollistaa *tunteiden säätelyn*. Tunteiden säätely on sitä, että pysyy tietoisena omista tunnetiloistaan ja käyttää tätä tietoa hyväkseen ongelmien ratkaisemisessa. Esimerkiksi surun tai vihan syy tulee selvittää ja ratkaista ongelma. Aina ei tietenkään voi poistaa surun tai vihan aiheuttajaa, mutta on hyvä edes tiedostaa, mistä tunteet johtuvat. Ilon tunteen aiheuttajia kannattaa myös tunnistaa, jotta niitä voi käyttää uudelleen. [Car99]

Omia tunteita ja tunteiden säätelymekanismeja kannattaa yrittää tiedostaa. Ihmisellä on taipumusta suojata itseään kestävämmiltä asioilta. Ihminen voi esimerkiksi kieltää tapahtuneen kokonaan samoin kuin hän voi sysätä tunteensa tiedostamattomaan. Se on selviytymisstrategia, jonka avulla selvittää pahoistakin tilanteista, kuten omat epäonnistumiset. Strategia voi olla esimerkiksi erilaiset selittelyt, kuten ”vika ei ole minussa”, jolloin epäonnistumisen kestää paremmin. Epäonnistumista kestää paremmin myös, jos asiaan ei ole omasta mielestään edes panostanut. Tällaiset strategiat vaativat kuitenkin aiheeseen uudelleen paalaamisen ja totuuden käsittelyn, jotta ihminen pysyy henkisesti terveenä.

Tunteista on hyötyä kognitiiviselle toiminnalle. Myönteisten tunteiden on havaittu lisäävän *luovaa ongelmanratkaisua* sekä parantavan *muistia, keskittymiskykyä* ja *oppimista*. Tunteet auttavat havainnoimaan ja arvioimaan asioita sekä tekemään päätöksiä. Jos tunnelatausta ei ole, silloin asia on merkityksetön eikä sitä havainnoida tehokkaasti [EyK95]. Tunteet toimivat *motivoijina* ja toiminnan *kohdentajina*. Lisäksi tunteiden ilmaiseminen parantaa *terveyttä* [Kok01].

6.3.4 Konfliktit

Aina ei voi olla kivaa. Hyvä työllisyystilanne voi vaikuttaa siten, että edes pieniä vastoinkäymisiä tai konflikteja ei selvitetä, vaan vaihdetaan työpaikkaa ongelmien ilmetessä. Toisaalta pelko työpaikan menettämisestä voi johtaa konfliktien selvittämättä jättämiseen. Sen asian tiedostaminen, että tunteet ovat konflikteissa päätekijöinä, voi olla hyödyllistä, ehkä yksi osa tunneälyä.

Konflikteja syntyy esimerkiksi intressiristiriidoista, joille ei voi mitään. Esimies voi esimerkiksi joutua valitsemaan kahdesta yhtä pätevästä henkilöstä toisen. Tällainen tilanne on vaikea. Päätös on perusteltava molemmille osapuolille, jottei tilanne johda selvittämättömään konfliktiin. Selvittämättömät ristiriitatilanteet vaikeuttavat ihmisten välistä yhteistyötä.

Ryhmän sisällä on konflikteja, kun ryhmä ei ole vielä löytänyt yhteistä identiteettiään. Myös ryhmien välille syntyy konflikteja. Konfliktit ovat kuitenkin osa muutosprosessia. Konfliktin rakentava käsittely johtaa uudistumiseen. Kaikkia osapuolia tulee kuunnella ja arvostaa. Pelkkä kuunteleminen ei kuitenkaan aina riitä, vaan voidaan tarvita konkreettisia muutoksia.

Konfliktin selvittämistä auttaa, jos ymmärretään, että konfliktissa pinnalla on tunteet. Ristiriitatilanteita syntyy paljon myös siksi, että ihmiset eivät aina tiedosta omia tunteitaan, vaikka toiset kuitenkin näkevät ne. [Tur90]

7 MITEN INHIMILLISET TEKIJÄT VOIDAAN OTTAA HUOMIOON?

Inhimilliset tekijät ovat tärkeitä vaikuttajia yrityksissä ja liiketoiminnan onnistumisessa. Ne voidaan huomioida vasta kun niistä tullaan tietoisiksi. Tietoisuuteen liittyy läheisesti läpinäkyvyys. Jos yrityksessä *mitataan* jotakin, esimerkiksi työilmapiiriä, mitattava asia tulee konkreettiseksi. On sanottu myös, että asiat joita mitataan tulevat tehdyiksi [Mar00]. Tämä pätee myös työntekijän työn määrittelyssä. Lähtökohta tulee olla se, että työntekijä tietää, mitä hänen työtehtäviinsä kuuluu ja miten hänen työtehtäviensä tuloksia arvioidaan.

7.1 YRITYSSTRATEGIA

Ihmisen *tarpeiden* tiedostaminen on yksi tapa parantaa työn tehokkuutta ja siihen sitoutumista [HeB91]. Tähän liittyy mm. kehityskeskustelut, tilannejohtaminen ja henkilöstön johtaminen sisäisinä asiakkaina. Huomionarvoista on se, että kehityskeskustelua voi ehdottaa myös alainen [Val00].

Ihminen on *kokonaisuus*. Työelämän laatu vaikuttaa elämänlaatuun. Toïssä onnistumiset vievät positiivista energiaa muillekin elämänalueille. Sen sijaan perheen mukavat tapahtumat eivät johda onnellisuuteen työssä. Toisaalta kuitenkin riittävä vapaa-aika mahdollistaa onnistumisten ja epäonnistumisten tasapainon saavuttamisen yksilötasolla.

7.1.1 Työelämän laatu

Yritysjohdo voi miettiä, painottaako *taloutta* vai *henkilöstön hyvinvointia*. Jos työpaikan taloudellista tilannetta kohotetaan vähentämällä henkilöstökuluja, kärsii henkilöstö. Jos palkkataso pidetään korkealla, työntekijöihin panostetaan ja heidän annetaan toteuttaa omia ideoitaan ilman näkyviä taloudellisia hyötyjä, johtaa tämä strategia taloudellisen tilanteen heikkenemiseen. Tästä puolestaan voi seurata henkilöstön mahdollisia irtisanomisia ja lomautuksia. Yrityksen strategia voi olla myös vähäinen panostus molempiin, mikä tarkoittaa siis vähäisiä tuloja ja alhaisia palkkakustannuksia yritykselle. Tällaisia yrityksiä on tosin Suomessa enää vähän. Korkein työelämän laatu näyttää siis olevan yrityksissä, jotka menestyvät samalla, kun työntekijöiden hyvinvointiin panostetaan. Toisaalta voidaan ajatella, että henkilöstön hyvinvointi tai talous eivät kumpikaan voi ”voittaa”, vaan

joudutaan antamaan myönnytyksiä ja tekemään kompromisseja, jolloin menestystekijäksi nouseekin joustavuus. [Työ02]

1990-luvulla menestyneitä työpaikkoja tutkittaessa [Työ02], niistä löydettiin joitakin yhteisiä piirteitä. Yksi piirre oli *funktionaalinen joustavuus*, joka tarkoittaa henkilöstön *mahdollisuuksia vaikuttaa omaan työhönsä ja kehittyä siinä*, jolloin yrityskin pystyy mukautumaan nopeasti muuttuviin ulkoisiin olosuhteisiin. Toinen menestykseen liittyvä yhteinen piirre on *halu ja kyky etsiä aktiivisesti uusia ratkaisumahdollisuuksia*. Tämä tarkoittaa työntekijän kannalta paljon vastuuta ja korkeaa vaatimustasoa. Yksilötasolla voidaan ollakin valmiita tähän, vaikka tämä koetaankin myös stressitekijänä. (ks. kappale 4.3.4 Stressi)

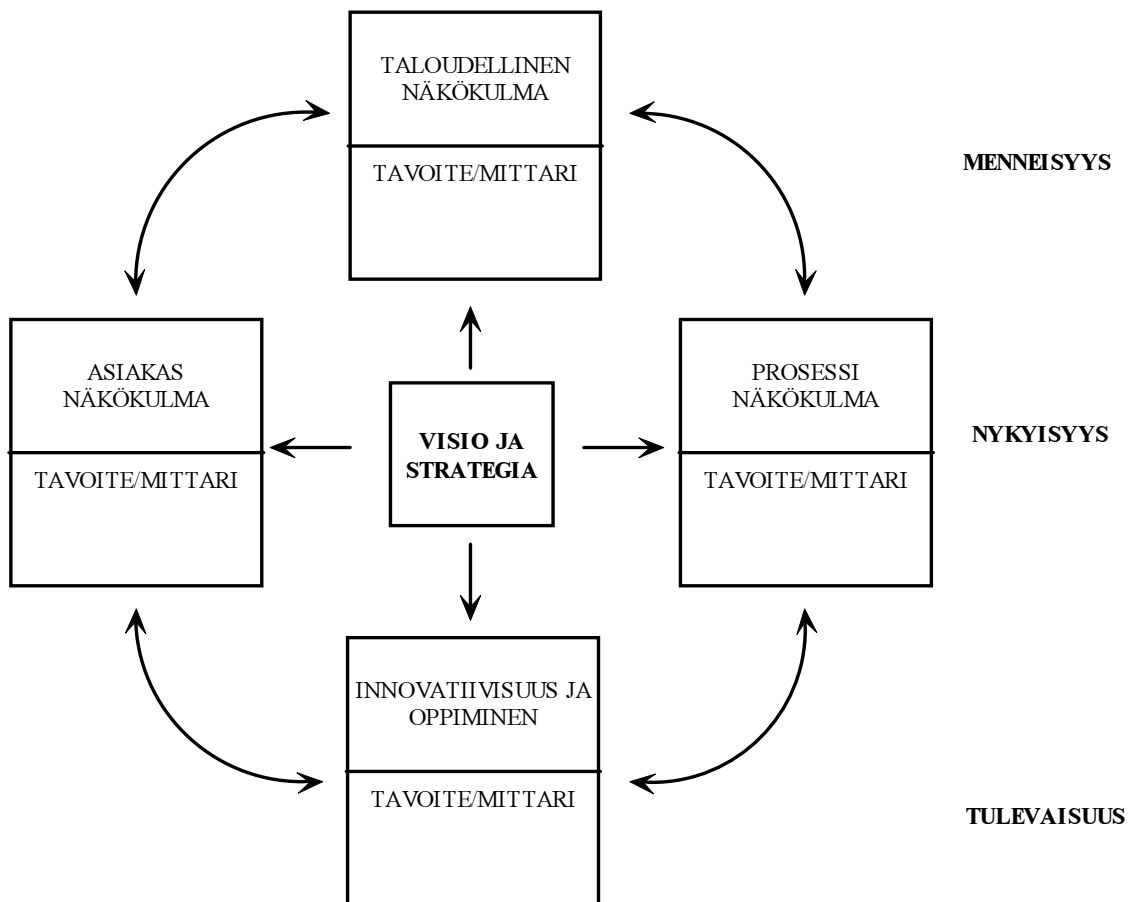
Vastuullisten työpaikkojen hyvinä puolina koetaan johdon tuki, kannustuksen antaminen avoimuus ja palkitseminen. Kanssakäymistä on enemmän kuin sellaisissa työpaikoissa, joissa vastuuta ja vaikutusmahdollisuuksia on vähän, mistä seuraa enemmän myös konflikteja. Johto pystyy kuitenkin käsittelemään niitä ja on muutenkin enemmän vastuussa alaisistaan kuin ei-vastuullisissa työpaikoissa. [Työ02]

Työelämän laadun parantamiseksi ei ole suoraviivaista mallia, koska muutoksen on tapahduttava jokapäiväisten käytäntöjen tasolla eikä vain yrityksen ”säännöissä”. Ohjenuorana voi kuitenkin pitää inhimillisten voimavarojen aikaisempaa järkevämpää käyttöä, jonka tarkoituksena on saavuttaa jatkuvaa menestyksellistä ja joustavaa liiketoimintaa [Työ02].

7.1.2 Strategiamalli

Yhtenä ratkaisuehdotuksena henkilöstön huomioonottamisessa voidaan pitää *Balanced Scorecard* -ajatusmallia (=BSC). Se on yritysstrategiamalli, jonka tarkoitus on saada aikaan tasapaino yrityksessä. Tasapainoa etsitään asiakkaiden tyytyväisyysmittareiden ja taloudellisten reportaasien sekä prosessin erinomaisuuden ja organisatorisen oppimisen suuntiin (ks. kuva 18). Mittarit keskittyvät organisaation *visioon* tasapainosta *asiakkaiden toiveiden (customer expectations)*, *sisäisen työtuloksen (internal performance)*, *tulevaisuuden kasvun (future growth)* ja *osakkaiden arvon (shareholder value)* kesken. Ajatusmallia käytetään myös visioiden muuttamiseen todelliseksi toiminnaksi ja apuna erilaisten johtamisen tarpeiden havaitsemisessa. Nimenomaan *tulevaisuuden kasvu* -osio

kohdistuu henkilökuntaan ja sen mahdollisiin odotuksiin, koulutushalukkuuteen ja -tarpeeseen. BSC auttaa koko organisaatiota ymmärtämään päämääriä ja tavoitteita.



Kuva 18: Balanced Scorecard [KaN92] [Mar00]

BSC:iin kuuluu muitakin kuin taloudellisia mittareita. Balanced Scorecardin hyödyt saadaan käyttöön yrityksessä esimerkiksi BSC:ia tukevan ohjelmiston avulla ja henkilöstön koulutuksella [KaN92], [Mar00].

7.2 NORMIT

Eri työpaikoissa voi olla hieman eri käytäntöjä ja kirjoittamattomia käyttäytymissääntöjä. Kysyttäessä ihmiset haluavat erilaisia ihmisiä, mutta usein odottavat ihmisten toimivan samoin kuin itse. Tämä on ihan ymmärrettävää ja useimmiten ihmiset käyttäytyvätkin tiettyjen normien puitteissa samalla tavalla. Ongelmaksi voivat muodostua esimerkiksi jokaisen työpaikan erilaiset kirjoittamattomat normit eli säännöt, joita jokainen pidempään

talossa ollut noudattaa ja joihin uudet työntekijät nopeasti alistetaan. Erilaisuutta ja sitä kautta tulevia uusia ideoita pitäisi kuitenkin osata arvostaa. Normeista poikkeaminen vaatii kuitenkin suurta rohkeutta ja voi johtaa pahoihinkin konflikteihin. Joskus tällaisia normien rikkojia tarvitaan, jotta kehitystä voi tapahtua.

7.2.1 Työmoraali

Vaikka IT-alalla on kiire, ei perustavaa laatua olevia normeja sovi laiminlyödä. Esimerkiksi jokaisen yksilön on edelleen tärkeää viedä työnsä huolellisesti loppuun asti. Jokaisen tulee tehdä annettu työ loppuun ilman jatkuvaa valvontaa ja muistuttelua. Kun asioita ruvetaan tekemään vain ”sinnepäin”, kärsii laatu ja saatetaan jopa rikkoa lakia. Jos esimies joutuu jatkuvasti muistuttamaan ”itsestään selvistä” asioista, kuluvat hänen voimavaransa tyhjänpäiväisten asioiden hoitoon tärkeiden asioiden kustannuksella. Yleensä yksilö oppii hahmottamaan tällaiset yleiset käytöstavat *häpeän* tunteensa avulla [Tur90].

7.2.2 Kirjoitetut säännöt

Normien perusta lienee kuitenkin toimia moraalisesti ja eettisesti oikein ja tasa-arvoisesti työtovereita kohtaan [Juu92]. Yrityksellä voi olla myös kirjoitetut eettiset säännöt, jotka voivat olla esimerkiksi yrityksen laatukäsikirjassa.

Tietotekniikan liitto ry on julkaissut Tietotekniikan ammattilaisen Eettiset Säännöt, jotka ovat lyhennettynä seuraavanlaiset [Tie02]: (Tietotekniikan ammattilaisen Eettiset Säännöt ovat kokonaisuudessaan liitteessä)

Tietotekniikan liitto ry: Tietotekniikan ammattilaisen Eettiset Säännöt:

1. *Tietotekniikan ammatissani toimin vastuullisesti, oikeudenmukaisesti, rehellisesti ja rohkeasti.*
2. *Tietotekniikan ammatissani en vahingoita luontoa.*
3. *Edistän kulttuurien välistä ymmärtämystä.*
4. *Edistän ihmisarvon toteutumista yhteiskunnassa.*
5. *Tuotteeni ja toimintani vastaavat eettisesti hyväksyttäviin inhimillisiin ja sosiaalisiin tarpeisiin.*
6. *Kunnioitan yksityisyyden suojaa. En hanki minulle kuulumattomia tietoja enkä käytä väärällä tavalla saatavilla olevia tietoja.*
7. *Kunnioitan toisen henkistä omaisuutta ja tekijänoikeuksia.*

8. Kehitän jatkuvasti itseäni ja erityisesti ammatillista osaamistani.

9. Pidän huolta itsestäni.

Toisten ihmisten huomioiminen on mukana kohdissa 3., 4., 5., 6. ja 7. ja oman itsensä huomioiminen kohdissa 1., 5., 8. ja 9. Kirjoitetut, luettavissa olevat, säännöt helpottavat työelämän pelisääntöjen ymmärtämistä.

Eettisten sääntöjen määrittäminen tietojenkäsittelijälle ei ole helppoa. Eettisten sääntöjen tekemisessä tulee ottaa huomioon *arvostukset* ja mahdolliset *moraaliset ongelmat*. Kaikkia ongelmatilanteita ei ole vielä tiedossakaan alan uutuuden vuoksi. Miten tehdä riittävän tarkat, mutta samalla laajasti hyväksyttävät ohjeet? Olisi hyvä muistaa myös, että liiallinen sääntöjen ja lakien tekeminen voi olla pahasta. Jos itse lain noudattamisesta tulee tärkeämpää kuin moraalisesti oikein toimimisesta, olemme pulassa [Sev97]. Olisi siis tärkeää, että ihmiset toimisivat oikein siksi, että se on heistä moraalisesti oikein eikä siksi, että laki sanoo niin. Atk-alan ammattilaisilla ei ole vielä sellaista kansainvälistä eettistä koodia kuten esimerkiksi lääkäreillä (Hippokrateen vala). Tehtyjen sääntöjen vikana on yleensä se, että ne ovat toistaiseksi hyvin yleisiä, joten niitä voi soveltaa varsin vapaasti.

7.3 OPPIVA ORGANISAATIO

Työn laadun ja tuottavuuden kehittämiseksi on kehitetty oppivan organisaation –malli. Oppivan organisaation käsite on tullut Suomeen 1990-luvun vaihteessa. Oppiva organisaatio tarkoittaa organisaatiota, joka kykenee jatkuvasti kehittymään. Kehittymisen mahdollistaa *jatkuva oppiminen* ja *tiedon jakaminen*. Oma tietämys jaetaan myös muiden saataville. Näin opitaan myös toisten onnistumisista ja epäonnistumisista. Oppiva organisaatio tarvitsee asenteen, jossa tietoa ei pidetä vain itsellä [SaS97]. Pitkällä tähtäimellä on mielekästä, että organisaatiossa useampi henkilö kykenee samoihin tehtäviin eikä kukaan ole korvaamaton.

Myös prosessijohtamiseen liittyy ajatus prosessien kehittämisestä tiedon jakamisen avulla. Se tieto, mikä on jonkun ihmisen päässä on tarkoituksenmukaista siirtää koko organisaation tietovarastoon. Tähän sopii hyvin oppivaan organisaatioon kuuluva tietämyksenhallintajärjestelmä. [Han94]

Asiantuntijan voi kuitenkin olla vaikea jakaa tietoaan. Tiedon jakamisen voidaan pelätä johtavan siihen, että kyseistä asiantuntijaa ei enää tarvita. Työntekijän stressi voi kuitenkin hellittää, kun hänen ei tarvitse tehdä kaikkea itse ja hän voi jäädä myös sairauslomalle ilman

paniikkia siitä, miten yritys pärjää. Työyhteisössä pitäisi olla joku, joka pystyy tuuraamaan toista tarvittaessa. On siis tärkeää opettaa toisillekin se, minkä jo itse osaa.

Oppivan organisaation strategiaa voidaan tarkastella monesta näkökulmasta. Mukana on kuitenkin yleensä *liiketoiminnan kehittäminen*, mutta myös *yksilöiden hyvinvointi ja kehittäminen* on tärkeää. Ei riitä, että yksilöt ovat oppivia vaan organisaatiosta tulee löytyä oppimista tukevia prosesseja [Moi99]. Ohjelmistotalossa tällaisia voisivat olla tarkastustilaisuudet ja projektien päättämistilaisuudet, jos ne viedään läpi oppimisen näkökulma muistaen.

Oppimiseen liitetään kiinteästi *reflektointi* [RaW97]. Syvällistä oppimista ei voi tapahtua kiireessä, sillä omaa oppimistaan pitää ehtiä reflektoida, jolloin siitä voi tulla syvällistä. Reflektointi voi olla itsenäistä asian käsittelyä jälkeinpäin tai ryhmässä tapahtuvaa keskustelua aiheesta.

Yksilön oppimista voidaan tukea tiimityöskentelyllä. Organisaation tieto voidaan koota tietämyksenhallintajärjestelmään, mutta sinne ei voida tallentaa taitoja. Eksplisiittistä tietoa voidaan kirjoittaa ja puhua, mutta *hiljaiseksi tiedoksi* nimetty tiedon muoto on taitoa, jota ei voi kirjasta lukemalla oppia. Sen voi oppia vain tekemisen kautta ja se perustuu kokemukseen. Tämän hiljaisen tiedon siirtämiseen tarvitaan työparia tai tiimiä, jonka keskuudessa tätä hiljaista tietoa pystyy siirtymään [NoT95, Käh02]. Taulukkoon 1 on koottu tiimioppimista edistäviä ja estäviä tekijöitä.

Taulukko 2. Tiimioppimista edistäviä ja estäviä tekijöitä [SaS97]

	TIIMIOPPIMISTA EDISTÄVÄT	TIIMIOPPIMISTA ESTÄVÄT
MOTIVOINTI	Ryhmä on ryhmänä tulosvastuullinen aikaansaannoksistaan. Saadaan aikaan toteutuneita kehittämishankkeita.	Esimies yksin vastuussa aikaansaannoksista. Kehittämishankkeet jäävät kesken.
ESIMIESTEN ROOLI	Esimiehet seuraavat ja tukevat aktiivisesti toimintaa.	Esimiehet ovat passiivisia tai vastaan haraavia
RYHMÄTOIMINTA JA RYHMÄDYNAMIIKKA	Yhteistyö sujuu. Ryhmä on kiinteä ja sen koko on 4-7 henkilöä. Kokoukset säännöllisiä ja toiminta hyvin jäsentynyt. Ryhmällä on innostunut vetäjä	Jäsenillä vaikeuksia irtautua työstä kokoukseen. Ryhmän kiinteys ja yhteistyökyky heikko. Toiminta on epäsäännöllistä.
TYÖN KOHDE JA TYÖYHTEISÖ	Ryhmää yhdistää yhteinen työn kohde. Koko työyhteisö on mukana kehittämishankkeessa tai ainakin tukee ryhmän työtä.	Luontevaa yhteistä kehittämishanketta ei löydy.
ORGANISAATION RAKENNE JA TYÖNJAKO	Työntekijät voivat itse suunnitella omaa työtään. Työntekijöillä on kehittämisvastuu, mutta myös kehittämisvapaus. Mikro- ja makrotason kehittäminen muodostaa johdonmukaisen kokonaisuuden. Matala organisaatiohierarkia. Pyritään monitaitoisuuteen.	Työn suunnittelu ja sen tekeminen on erotettu toisistaan. Koulutus on sosiaalista vaihtelua tai palkintona hyvin tehdystä työstä.
OPPIMINEN	Kehittämistyö johtaa lisätiedon hankintaan, asioiden käsitteellistämiseen ja refleктоivan, kriittisen työotteen syntymiseen. Halu laaja-alaisempaan asioiden oppimiseen ja opiskeluun herää.	Oppiminen satunnaista ja pelkästään arkikokemukseen perustuvaa. Tietoinen oppimisenäkökulma puuttuu.

8 POHDINTA

Ohjelmistotuote on altis monille inhimillisille tekijöille elinkaarensa aikana. Tärkeimpiä näistä ovat motivaatio ja johtamistaito. Kognitiiviset taidot ovat tärkeitä, mutta myös sosiaaliset taidot ovat tärkeässä asemassa ohjelmistotuotantoprosessissa, johon osallistuu useita erilaisia ihmisiä. Inhimillisten tekijöiden huomioiminen voi tuottaa laadukkaita ja tehokkaita prosesseja esimerkiksi siksi, että ihmiset ovat motivoituneempia tekemään työnsä hyvin.

Olioparadigman odotettiin parantavan ohjelmistojen sekä projektityön laatua, mutta oliot eivät onnistuneet täyttämään kaikkia odotuksia. Mielenkiintoista on, miten tulevaisuuden *komponenttisysteemit* onnistuvat huomioimaan inhimilliset tekijät. Vaikuttaako uudenlainen työnjako projektien muokkautumiseen hallittavammiksi kokonaisuuksiksi ja lisääntykö ohjelmistojen uudelleenkäytettävyys ja johtaako se työmäärän vähenemiseen IT-alalla?

Mitkä mahtavat olla *tiimien* tulevaisuuden mahdollisuudet? Jos ihmisten erilaisuuden hyödyntäminen on periaatteena, miten hyvin se toimii todellisuudessa? Jaksavatko ihmiset arvostaa toistensa erilaisuutta riittävästi, jotta erilaisuus nähdään voimavarana eikä hankaluutena? Ryhmän toimintaa voi helpottaa kuitenkin tieto siitä, että konfliktivaiheen jälkeen on mahdollisuus siirtyä kiinteän ryhmän vaiheeseen. Konfliktit pitäisi ottaa osana muutosprosessia.

Mitä ohjelmistotuotannon jatkuvalla *kiireelle* pitäisi tehdä? Kiireessä ei voi olla luova, empaattinen eikä oppiva. Kiireestä syntyy huonoa laatua ja stressiä. Toisaalta kiireelle on myös perustelunsa; jos tuote ei ole ajoissa markkinoilla, ei tuotetta kannata tehdä ollenkaan. Tulee siis pyrkiä *riittävään laatuun*, siksi on erittäin tärkeää, että yrityksessä on selkeästi määritelty yrityksen laadun taso. Myös päteviä työntekijöitä pitää olla riittävästi. Siksi tietoa ja taitoa on hyödyllistä jakaa esimerkiksi *oppivan organisaation* mallin mukaisesti.

Tutkielmassa on havaittu, että haasteellinen työ ja sopiva vastuu voidaan kokea myös *motivoivana*. Jatkossa tutkimisen arvoista on, haalitaanko ohjelmistoalalla liikaa töitä itselle vai annetaanko ne ylhäältä, sillä erityisen tärkeää on *yksilöllinen kokemus* siitä, miten hyvin asioihin voi itse vaikuttaa. Myös *turvallisuuden tunne* on hyvin tärkeä

motivaatiotekijä, jota ei useinkaan tulla ajatelleeksi. Turvallisuuden tunnetta voidaan lisätä riittävällä tiedottamisella ja hidastamalla muutosvauhtia. Jos ihminen kokee olonsa turvalliseksi työpaikallaan, hän ei todennäköisesti suunnittele työpaikan vaihtoa. Huomionarvoista on, että motivaation lähteeksi riittää työ itsessään, mutta lisämotivaationa toimii hyvin esimiehen asenne ja työryhmän kiinteys.

Ihmisten onnellisuus ei suoraan johda tehokkaampiin prosesseihin, mutta tyytymättömyys ja motivoitumattomuus voivat johtaa huonoon ilmapiiriin sekä aiheuttaa katkoksia tiedonkulussa. Jo nämä tekijät riittävät aiheuttamaan sekä prosessin että tuotteen laadun huononemisen.

Tehokkuuteen ja tuottavuuteenkin on kehitetty niin sanottuja pehmeitä ratkaisuja, kuten *tilannejohtaminen* autoritaarisen johtamisen sijaan tai *organisaation baskerimalli* pipomallin sijaan. Inhimillisten tekijöiden tiedostaminen ei ole kuitenkaan pelkästään johdon tehtävä, sillä alaisilta vaaditaan myös omien rajojen tuntemista, kuten stressinsietokyvyn tunnistamista ja rohkeutta kieltäytyä mahdottomista tehtävistä. Organisaation ja esimiesten tulee mahdollistaa yksilön hyvä *työvire*, joka syntyy esimerkiksi riittävästä palkkioista ja vapaa-ajan mahdollisuuksista. Jokainen yksilö on myös vastuussa organisaatiolle omasta työvireestään, jota jokainen pitää yllä itselle sopivimmalla tavalla, esimerkiksi riittävän yöunen, harrastusten ja liikunnan avulla.

Lisäksi on hyvä tunnistaa omat motivaation lähteensä, jotta niillä voi motivoida itseään uusissa tilanteissa. Tämä on yksi osa-alue *tunneälykkään* ihmisen taidoissa. Tunneälyn avulla ihminen kykenee hyödyntämään omia, mutta myös toisten, tunteita vuorovaikutustilanteiden helpottamiseksi, mikä johtaa muun muassa parempaan ryhmähenkeen ja miellyttävämpiin esimies-alainen –suhteisiin.

Inhimillisten tekijöiden huomioon ottamiseksi *on olemassa* erilaisia *ehdotuksia*. Joitakin näistä on esitelty tässä tutkielmassa. Eri asia kuitenkin on, kuinka hyvin näitä oppeja sovelletaan tai voidaan soveltaa käytännössä. Toimintatavan muuttaminen ei onnistu yhdessä yössä eikä ole tarkoituksenmukaistakaan. Jos toimintatapaa halutaan muuttaa, pitää muutoksen lähteä yksilötasolta. Ensimmäinen askel on *asioiden tiedostaminen*. Tämän tutkielman lukeminen mahdollistaa tuon ensimmäisen - mutta tärkeimmän - askeleen tekemisen, minkä jälkeen voidaan miettiä, *tarvitaanko muutoksia*.

Jatkotutkimusta kannattaisi tehdä siitä, miten inhimilliset tekijät on huomioitu yrityksissä ja miten yrityksissä työskentelevät ihmiset sen kokevat. Mielenkiintoista on myös nähdä,

miten pitkään ylitöitä jaksetaan tehdä. Tällä hetkellä ylitöitä jaksetaan, koska työtä itsessään pidetään motivoivana, mutta miten käy, kun ylitöitä tehdään vuosia ja vuosikymmeniä. Lisäksi olisi hyvä etsiä organisaation, johdon sekä yksilötason strategioita selviytyä nopeista muutoksista, koska *joustavuus* näyttää olevan tekijä, joka johtaa parempaan menestymiseen kaikilla näillä tasoilla.

LÄHTEET

- [AvF95] Avison, D.E. & Fitzgerald, G.: *Information systems development: Methodologies, Techniques and Tools*. McGraw-Hill, London 1995.
- [Car02] Caruso, David R.: *General Emotional Intelligence Home Page*. Saatavilla www-muodossa: <URL: <http://www.emotionaliq.com/Gdefault.htm>> viitattu 3.5.2002.
- [Car99] Caruso, David R.: *Applying the Ability Model of Emotional Intelligence to the World of Work*. 1999. Saatavilla www-muodossa: <URL: <http://www.cjwolfe.com/whatis.htm>> viitattu 2.5.2002 .
- [Che76] Chen, P.: *The Entity-Relationship Model – Toward a Unified View of Data*, ACM transactions on Database Systems, Vol.1, No.1, Mar 1976, pp 9-36.
- [ChJ00] Chatfield, Carl S. & Johnson, Timothy D.: *Microsoft Project 2000 Trainer*. Englanninkielisestä käsikirjoituksesta Microsoft Project 2000 Step by Step suomentanut Avoin yhtiö Kuumaa: Leena ja Antti Kivivalli, Oy Edita Ab, 2000.
- [CoD98] O'Connor, Joseph & McDermot, Ian: *NLP:n aakkoset*. Helsinki ai-ai, 1998.
- [Cop99] Coplien, James O.: *A Pattern Language for Writers' Workshops*. Proceeding of EuroPLoP '97 (Siemens Technical Report—Forthcoming), Bell Laboratories, 1999.
- [CoY90] Coad, Peter & Yourdon, Edward: *Object-Oriented Analysis*. Englewood Cliffs Prentice Hall, New Jersey, 1990.
- [Dah68] Dahl, Ole-Johan: *Simula 67 : common base language*. Oslo: Norsk Regnesentral, 1968.
- [DeM78] DeMarco, Tom: *Structured Analysis and System Spesification*. Yourdon Press/Prentice Hall, 1978.
- [Duc86] Duck, Steve: *Human relationships, An Introduction to Sosial Psychology*. SAGE Publications, London, 1986

- [Dun98] Dunderfeld, Tony: *Henkilökemia: Yhteistyö erilaisten ihmisten välillä*. WSOY - Kirjapainoyksikkö, Juva, 1998.
- [Erg02] *Ergonetti*. Kuopion yliopiston Koulutus- ja kehittämiskeskus, Avoin yliopisto. Saatavilla www-muodossa:
<URL:<http://www.oppi.uku.fi/ergonetti/index.html>>, viitattu 22.4.2002.
- [EyK95] Eysenck, M.W. & Keane, M.T.: *Cognitive psychology. A student's handbook* (3. edition). Hove: Psychology Press, 1995.
- [Fow00] Fowler, Martin: *UML distilled : a brief guide to the standard object modeling language*. Addison-Wesley, 2000.
- [Gol97] Goleman, Daniel: *Tunneäly*. Englanninkielisestä käsikirjoituksesta Emotional Intelligence suomentanut Kankaanpää Jaakko, Otava, 1997.
- [Gol99] Goleman, Daniel: *Tunneäly työelämässä*. Englanninkielisestä käsikirjoituksesta Working with Emotional Intelligence suomentanut Kankaanpää Jaakko Otava, 1999.
- [Hag91] Hageman, Gisela: *Motivoinnin taito*. Englanninkielisestä käsikirjoituksesta The Motivation Manual suomentanut Ullan Arkki Oy, Ulla Ropponen, Weilin-Göös, 1991.
- [Hak95] Hakanen, Jari: *Vuorovaikutustaidot ryhmässä*, TSL-opintokeskus, Helsinki, 1995.
- [HaM01] Haikala, Ilkka & Märijärvi, Jukka: *Ohjelmistotuotanto*. Tummavuoren Kirjapaino Oy, Vantaa, 2001
- [Han94] Hannus, Jouko: *Prosessijohtaminen – Ydinprosessien uudistaminen ja yrityksen suorituskyky*. Gummerus Kirjapaino Oy, Jyväskylä, 1997.
- [Haw01] Hawryszkiewicz, Igor, T.: *Introduction to Systems Analysis and Design*. Prentice Hall, Australia, 2001.
- [HeB90] Hersey, Paul & Blanchard, Kenneth: *Tilannejohtaminen, tuloksiin ihmisten avulla*. Gummerus Kirjapaino Oy, Jyväskylä 1990.
- [HeS00] Herzum, Peter & Sims, Oliver: *Business component factory: a comprehensive overview of component-based development for the enterprise*. New York Wiley cop., 2000.

- [Hof93] Hofstede, Geert: *Kulttuurit ja organisaatiot - Mielen ohjelmointi*. WSOY, Helsinki, 1993.
- [Hum98] Humphrey, Watts S.: *Managing technical people: innovation, teamwork and the software process*. Addison-Wesley Longman, Inc, Massachusetts, United States of America, 1998.
- [Iso91] Isoherranen, Kaarina: *Sosiaalipsykologiaa keskiasteelle*. Kirjayhtymä, Helsinki, 1991.
- [Jac83] Jackson, Michael: *System development*. Prentice-Hall. 1983.
- [Jac98] Jacobson, Ivar: *Object-oriented software engineering: A Use case driven approach*. Addison-Wesley, 1998.
- [Jal99] Jalote, Pankai: *CMM in practice: processes for executing software projects at infosys*. Addison-Wesley, 1999.
- [JBR99] Jacobson, I., Booch, G. & Rumbaugh, J.: *The unified software development process*. Addison-Wesley, 1999.
- [Juu92] Juuti, Pauli: *Organisaatiokäyttäytyminen*. Kustannusosakeyhtiö Otavan painolaitokset, Keuruu 1992.
- [Juu95] Juuti, Pauli: *Johtaminen ja organisaation alitajunta*. Kustannusosakeyhtiö Otavan painolaitokset, Keuruu, 1995.
- [Käh02] Kähkönen, Tuomo: *Ihminen ohjelmistokehitysprosessissa, todellista tietoa ja oikeaa osaamista*. Sytyke ry - Systemityölehti 1/02, 2002.
- [KaN92] Kaplan, Robert, S. & Norton, David, K.: *The Balanced Scorecard: Measures that drive performance*. Harvard Business Review, 1992.
- [Ken87] Kendall, Penny A.: *Introduction to Systems Analysis and Design, A structured approach*. Allyn and Bacon, Boston, United States of America, 1987.
- [Kok01] Kokkonen, Marja: *Emotion Regulation and Physical Health in Adulthood: A Longitudinal, Personality-Oriented Approach*. University of Jyväskylä, 2001.
- [Kos00] Koskimies, Kai: *Oliokirja*. Gummerus Kirjapaino Oy, Jyväskylä, 2000.

- [Kru00] Kruchten, Philippe: *The Rational Unified Process an introduction*. Addison-Wesley, 2000.
- [KSO00] Korpela, Mikko, Soriyan, H.A. & Olufokunbi, K.C: *Activity Analysis as a Method for Infomation Systems Development*. Scandinavian Journal of Information Systems (2000:12, s.191-210), 2000.
- [LoG96] Lockyer, Keith & Gordon, James: *Project Management and Project Network Techniques*. London Pitman, Great Britain, 1996.
- [Mar00] Marshall, Chris: *Enterprise modelling with UML: designing successful software through business analysis*. Addison-Wesley cop, Massachusetts, United States of America, 2000.
- [Mau96] Maunula, Reijo: *Kirje esimiehelle*. Johtamistaidon opisto JTO ja Oy Edita Ab, Helsinki, 1996.
- [Mey88] Meyer, Bertrand: *Object-oriented Software Construction*. The University Press, Campridge, 1988.
- [Mod01] *Moderni kognitiotiede*, toim. Saariluoma, Pertti & Kamppinen, Matti & Hautamäki, Antti. Yliopistopaino, Helsinki, 2001.
- [Moi99] Moilanen, Raili: *Oppiva organisaatio?* Sytyke ry - Systeemyölehti 3/99, 1999.
- [NoT95] Nonaka, Ikijiro & Takeuchi, Hirotaka: *The knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press, 1995.
- [Oli92] *Oliot systeemyössä*. Systeemyöyhdistys Sytyke ry. Suomen Atk-kustannus Oy, Espoo, 1992.
- [Pel96] Pelin, Risto: *Projektihallinnan käsikirja*. Gummerus Kirjapaino Oy, 1996.
- [Pir95] Pirnes, Unto: *Kehittyvät tiimit*. Tammer-Paino Oy, Tampere, 1995.
- [Poh02] Pohjonen, Risto: *Tietojärjestelmien kehittäminen*. Docendo, Jyväskylä, 2002.
- [Put94] Putkonen, Anne: *A methodology for supporting analysis, design and maintenance of object-oriented systems*. Kuopio University Printing Office, Kuopio, 1994.

- [Pyl99] Pylkkö, Juha: *Ohjelmistotyön laadun mittaaminen*. Sytyke ry -
Systeemyölehti 1/99, 1999.
- [RaW97] Rauste-von Wright, Majjaliisa & von Wright, Johan: *Oppiminen ja koulutus*.
WSOY:n graafiset laitokset, Juva, 1997.
- [RoR99] Robertson, Suzanne & Robertson, James: *Mastering the requirements
process*. Addison-Wesley, 1999.
- [Roy70] Royce, W.W.: *Managing the development of large software systems*. IEEE
WESTCON, Los Angeles, 1970.
- [Roy98] Royce, Walker: *Software Project Management, A Unified Framework*.
Addison-Wesley, 1998.
- [RuS98] Ruben, Bren D. & Stewart, Lea P.: *Communication and human behavior*.
Allyn & Bacon cop, Boston, 1998.
- [SaS97] Sarala, Urpo & Sarala, Anita: *Oppiva organisaatio – oppimisen, laadun ja
tuottavuuden yhdistäminen*. Tammer-Paino Oy, Tampere, 1997.
- [Sch99] Schach, S.R.: *Classical and object-oriented software engineering with UML
and C++*. 4th edition, McGraw Hill Collage Div, 1999.
- [Sev97] Severson R.J. : *The Principles of Information Ethics*. Armonk (N.Y): M.E.
Sharpe, cop. 1997.
- [Som95] Sommerville, Ian: *Software Engineering*. Addison-Wesley Publishers Ltd.,
Edinburgh Gate, 1995.
- [Ste96] Stenlund, Heikki: *Projektijohtamisen perusteet*. Oy Edita Ab, Helsinki, 1996.
- [SuR99] Suutama, T. & Ruoppila, I.: *Iäkkäiden ihmisten tiedonkäsittelytoimintojen
muutokset kahdeksan vuoden aikana testitulosten itsearviointien perusteella*.
Teoksessa Suutama, T., Ruoppila, I. & Laukkanen, P. (toim.), Iäkkäiden
henkilöiden toimintakyvyn muutokset. Havaintoja Ikivihreät-projektin 8-
vuotisesta seuruututkimuksesta (s.99-116). Helsinki: Hakapaino Oy, 1999.
- [Szy98] Szyperski, Clemens: *Component Software Beyond Object-Oriented
Programming*. Harlow ACM Press, United States of America, 1998.
- [TaG96] Takang, Armstrong A. & Grubb, Penny A.: *Software Maintenance: Concepts
and Practice*. International Thomson Computer Press, London, 1996.

- [Ter94] *Terve työyhteisö - kehittämisen malleja ja menetelmiä*. Helsinki Työterveyslaitos, 1994.
- [TeS01] Tegarden, David P. & Sheetz, Steven D.: *Cognitive activities in OO development*. Int. J. Human-Computer Studies 54, (s.779-798), Academic Press 2001.
- [Tie02] Tietotekniikan liitto ry, *Eettiset säännöt*. Saatavilla www-muodossa: <URL:<http://www.ttlry.fi/eettiset.htm>>, viitattu 13.3.2002.
- [Toi00] *Toimivat ja terveet työajat*. Työterveyslaitos, Sosiaali- ja terveysministeriö, työministeriö, Helsinki, 2000.
- [Toi94] Toivanen, Helli: *Occupational Stress in Working Women and the Benefits of Relaxation Training, Studies on Bank Employees, Home Helps and Hospital Cleaners*. Kuopio University Printing Office, Kuopio, 1994.
- [ToK98] Toivonen, Veli-Matti & Kiviaho, Matti: *Tässä suhteessa; Erilaisuus, yhteys ja yhteistyö – NLP Vuorovaikutuskirja*. Gummerus Kirjapaino Oy, Jyväskylä, 1998.
- [Tur90] Turunen, Kari E.: *Ihmisen ymmärtäminen*. Atena, Jyväskylä, 1990.
- [Työ02] *Työn muutos ja hyvinvointi tietoyhteiskunnassa*, toim. Härmä, Mikko & Nupponen, Tarja., Sitran raportteja 22, 2002. Saatavilla www-muodossa: <URL: <http://www.sitra.fi/pdf/raportti22.pdf>>, viitattu 8.4.2002.
- [Typ02] *Työelämän psykologia –luentosarja*. Oulun yliopisto, 1999. Saatavilla www-muodossa:<URL:http://edtech.oulu.fi/ftp_root/sampo/seija/persoonallisuuden psykologia.htm >, viitattu 22.4.2002.
- [UuP99] Uusikykä, Kari & Piirto, Jane: *Luovuus - Taito löytää, rohkeus toteuttaa*. WSOY, Juva, 1999.
- [Val00] Valpola, Anneli: *Kehityskeskustelun mahdollisuudet*. WSOY, Juva, 2000.
- [Vir01] Virtanen, Pentti: *Empirical Study for Evaluating Component Reuse Metrics*. In Maxwell, K., Oligny S., Kusters R. and van Veenendaal E. (eds.): Proceedings of the 12th European Software Control and Metrics Conference, page(s) 125-135, Shaker Publishing, April, 2001.

- [VLK98] Vanhala, S., Laukkanen, M., Koskinen, A.: *Liiketoiminta ja johtaminen*. Otavan Kirjapaino, Keuruu, 1998.
- [VuT92] Vuorinen, Risto & Tuunala, Eliisa: *Psykologian perusteet, lukion kurssi 2*. Kustannusosakeyhtiö Otavan painolaitokset, Keuruu, 1992.
- [VuT95] Vuorinen, Risto & Tuunala, Eliisa: *Psykologian perusteet, Aivot ja psyyke*. Kustannusosakeyhtiö Otavan painolaitokset, Keuruu, 1995.
- [You89] Yourdon, Edward: *Modern structured analysis*. Prentice-Hall, New Jersey, 1989.

LIITE: Tietotekniikan ammattilaisen Eettiset Säännöt

Sääntö 1

Tietotekniikan ammatissani toimin vastuullisesti, oikeudenmukaisesti, rehellisesti ja rohkeasti.

Esimerkkejä vastuullisuudesta:

1. Osallistun vain sellaisiin projekteihin, joiden aikataulu ja resurssien käyttö voidaan arvioida kohtuullisiksi.
2. Tiedotan riittävästi työtovereilleni, esimiehilleni, asiakkailleni ja käyttäjille projektin tilasta ja edistymisestä.
3. Tiedotan tilanteista, joista voi olla vahingollisia seurauksia.

Esimerkkejä oikeudenmukaisuudesta:

1. En vahingoita kollegaani epäoikeudenmukaisella arvostelulla.
2. Osallistun vain sellaisiin tehtäviin, jotka ovat sopusoinnussa yleisen oikeustajun kanssa.
3. Toimin oikeudenmukaisten tietoteknisten palvelujen ja ratkaisujen aikaansaamiseksi.

Esimerkkejä rehellisyydestä:

1. Laskutusta varten raportoin rehellisesti asiakkaalle projektiin käytetyn työajan.
2. Teen ohjelmia vain laillisilla lisensseillä.
3. Tuon esille liian tiukan aikataulun.
4. Arvioin rehellisesti omaa toimintaani.

Esimerkkejä rohkeutta vaativista tilanteista:

1. Pelkkä passiivisuus ei riitä, eettisyys velvoittaa vastustamaan hakkerointia sekä virusten julkaisemista ja levittämistä.
2. Jos huomaan ohjelmistossa laittomasti kopioidun osan, tuon asian rohkeasti esiin.
3. Huomaan projektin loppuvaiheessa ohjelmointivirheen. Virheen korjaaminen viivästyttäisi projektin valmistumista. Virheen paljastaminen saattaa aiheuttaa yritykselle viivästyssakot ja minulle potkut. Mahdollisesti virhe ei koskaan tulisi ilmi, mutta toisaalta saattaa jossain erityistapauksessa aiheuttaa kohtalokkaan vahingon asiakkaalle tai yhteiskunnalle. Tuon asian rohkeasti esiin.

Sääntö 2

Tietotekniikan ammatissani en vahingoita luontoa.

Esimerkkejä:

1. Tietotekniikan ammattilaisena otan vastuun siitä, että tietotekniikkaa ei käytetä ympäristöä vahingoittavalla tavalla.
2. Edistän laitteiden kierrätystä ja raaka-aineen ja muun materiaalin määrän minimoimista.

jatkuu

Sääntö 3**Edistän kulttuurien välistä ymmärtämystä.**

Esimerkkejä:

1. Pidän arvossa kansallisten ja paikallisten kulttuurien korvaamatonta ja eettisesti arvokasta perintöä.
2. Rakentaessani ja sovittaessani ohjelmia ja järjestelmiä paikalliseen ympäristöön otan mukaan paikallisten kulttuurien edustajia.

Sääntö 4**Edistän ihmisarvon toteutumista yhteiskunnassa.**

Esimerkkejä:

1. Ihmisarvon käsitämmme siten, kuin se on määritelty Yhdistyneiden kansakuntien yleismaailmallisessa ihmisoikeuksien julistuksessa.
2. Edistän mm. oikeutta yksilön suojaan, tasa-arvoon, itsensä toteuttamiseen, lepoon ja virkistykseen.
3. En riistä ihmisarvon osatekijöitä millään tietoteknisillä toimilla.
4. Vaikutan siihen, että tietotekniikan avulla tehtävää työtä ei jaeta työntekijän kannalta liian pieniin osiin, jottei kadoteta työn mieltä ja kokonaisuuden hallintaa.
5. Kehitän järjestelmiä yhteistyössä käyttäjien kanssa.

Sääntö 5**Tuotteeni ja toimintani vastaavat eettisesti hyväksyttäviin inhimillisiin ja sosiaalisiin tarpeisiin.**

Esimerkkejä:

1. Pidän mielessäni, että järjestelmiä tekevät aina ihmiset ihmisille. Suunnitellessani ja toteuttaessani tietojärjestelmiä en välineellistä ihmistä prosessin osaksi.
2. Kehitän järjestelmiä, jotka tuottavat hyvän työskentely-ympäristön ja huomioivat käyttäjän edun.

Sääntö 6**Kunnioitan yksityisyyden suojaa. En hanki minulle kuulumattomia tietoja enkä käytä väärällä tavalla saatavillani olevia tietoja.**

Esimerkkejä:

1. En käytä väärin henkilörekisterin tietoja, joihin minulla on luvallinen pääsy enkä tietoja, joita voin saada eri järjestelmien tietoja yhdistelemällä.

Sääntö 7**Kunnioitan toisen henkistä omaisuutta ja tekijänoikeuksia.**

Esimerkiksi:

1. En käytä internetistä tai muusta tietoverkosta saamaani teosta tai ideaa luvatta enkä lähdeittä mainitsematta.

jatkuu

Sääntö 8**Kehitän jatkuvasti itseäni ja erityisesti ammatillista osaamistani.**

- 1.Toimin ammattimaisesti
- 2.Pyrin laajaan ammatilliseen osaamiseen. Olen valmis oppimaan jatkuvasti.
- 3.Pidän itseni ammattikuntaani koskevien sopimusten, lakien ja asetusten tasalla.
- 4.Kehitän sosiaalisia taitojani. Niitä tarvitaan esimerkiksi ryhmätyötilanteissa, esimies-alaisuudessa ja asiakasyhteyksissä.

Sääntö 9**Pidän huolta itsestäni.**

Ihminen on fyysinen ja henkinen kokonaisuus, jonka molempien puolien vaaliminen on yhtä tärkeää. Itsestä huolehtiminen on eettisen toiminnan perusta. Siksi kuuntelen itseäni ja muistan myös levätä.

Lähde:
<http://www.ttlry.fi/eettiset.htm>