

Mikko Mattila

**Hajauttamisen ongelmat ohjelmistokehityksessä
Ratkaisuna Scrum?**

Tietojärjestelmätieteen
kandidaatintutkielma
31.08.2009

Jyväskylän yliopisto
Tietojenkäsittelytieteiden laitos
Jyväskylä

TIIVISTELMÄ

Mattila, Mikko Antero

Tietojärjestelmätieteen kandidaatintutkielma

Jyväskylä, Jyväskylän yliopisto, 2009, 28 s.

Tutkielmassa käsitellään hajautetussa ohjelmistokehityksessä esiintyneitä ongelmia, sekä pohditaan soveltuuko Scrum ratkaisuksi niihin. Esittelen yleisiä hajautetun ohjelmistokehityksen ongelmia ja niiden ratkaisuja, sekä käytännön kokemuksia hajautetuista Scrum-projekteista. Lopuksi vertailen yleisiä hajautetun kehityksen ongelmia case-tutkimuksissa esiintyneisiin ongelmiin, sekä arvioin Scrumista saatavia hyötyjä hajautetussa ohjelmistokehityksessä.

Hajautetun ohjelmistokehityksen ongelmat voidaan kategorisoida viiden keskipakoisvoiman avulla: kommunikaatio, koordinaatio, kontrolli, koheesio ja kulttuuri.

Totesin Scrumin lieventävän useita hajauttamisesta johtuvia ongelmia. Scrum parantaa kommunikaatiota sekä eri toimipisteiden välillä, että yhden toimipisteen sisällä. Se myös helpottaa projektin kontrollointia ja koordinointia. Eräiden Scrumin käytäntöjen, kuten suunnittelukokousten, havaittiin parantavan myös toimipisteiden välistä yhteishenkeä, eli koheesiota. Scrumin todettiin jossain määrin lieventävän myös kulttuurieroista johtuvia ongelmia.

Voidaan siis sanoa, että Scrum auttaa useimpien hajauttamisen aiheuttamien ongelmien voittamisessa, ja soveltuu hyvin hajautettuun ohjelmistokehitykseen, vaikka se ei pystykään kaikkia ongelmia ratkaisemaan.

AVAINSANAT: tutkielma, Scrum, hajauttaminen, ketterä ohjelmistokehitys, ongelmat

Ohjaaja: Timo Käkölä
Tietojenkäsittelytieteiden laitos
Jyväskylän yliopisto

Tarkastaja: Timo Käkölä
Tietojenkäsittelytieteiden laitos
Jyväskylän yliopisto

SISÄLLYSLUETTELO

1 JOHDANTO.....	5
2 HAJAUTTAMISEN AIHEUTTAMAT ONGELMAT	7
2.1 Viisi keskipakoisvoimaa	7
2.1.1 Communication breakdown.....	8
2.1.2 Coordination breakdown.....	9
2.1.3 Control breakdown.....	9
2.1.4 Cohesion barriers	10
2.1.5 Culture clash	10
2.2 Keskipakoisvoimien yhteisvaikutukset.....	11
2.3 Ongelmien ratkaiseminen	12
3 KÄYTÄNNÖN KOKEMUKSIA HAJAUTETUISTA PROJEKTEISTA	15
3.1 Tuottavuutta Scrumilla ja XP:llä.....	15
3.2 Scrumin käyttö suuressa projektissa.....	16
3.3 Kokemuksia Inteliltä ja HP:lta	19
4 ANALYYSI.....	21
4.1 Formalisointia ja informalisointia	21
4.2 Havaintoja ja johtopäätöksiä.....	22
YHTEENVETO	25
LÄHTEET	27

1 JOHDANTO

Tämä tutkielma keskittyy hajautetuissa projekteissa esiintyneisiin ongelmiin, sekä tapoihin, joilla niitä voidaan lieventää tai ratkaista. Otan kantaa myös siihen, kuinka ketterät menetelmät – erityisesti Scrum – soveltuvat hajautettuun ohjelmistokehitykseen. Tutkimusta tehdään projektinhallinnallisesta näkökulmasta, joten strategisen tason ongelmat jätetään käsittelyn ulkopuolelle. Myöskään ulkoistamisen ongelmiin ei puututa.

Scrum on iteratiivinen ja inkrementaalinen ohjelmistokehitykseen tarkoitettu menetelmä. Scrum keskittyy erityisesti projektityön hallintaan. Se helpottaa jäljellä olevan työmäärän arviointia ja parantaa näkyvyyttä projektin edistymisen suhteen. Sen avulla kehitystiimit voivat tarjota toimivaa ohjelmistoa jokaisen iteraation päätteeksi. Tämä mahdollistaa ketterän reagoinnin muuttuviin vaatimuksiin, sekä auttaa havaitsemaan vaatimusmuutokset aikaisemmassa vaiheessa. Scrumin käyttö parantaa henkilöstön motivaatiota, sekä lisää aktiivista kommunikointia. (Mahnic & Drnovscek, 2005) Eräissä tapauksissa Scrumilla on saavutettu jopa 10 kertaa parempaa tuottavuutta, kuin vesiputousmalliin perustuvilla menetelmillä. (Cohn, 2004)

Ennakkotapaukset erinomaisesta tuottavuudesta (Cohn, 2004; Sutherland, ym., 2007; Sutherland, ym., 2008) tekevät Scrumista houkuttelevan vaihtoehdon monille ohjelmistotaloille. Ohjelmistokehityksen nykytrendiin kuitenkin kuuluu myös offshoring, eli kehityksen siirtäminen maihin, joissa esimerkiksi kustannustaso on alhaisempi, tai osaavaa työvoimaa on paremmin saatavilla. Joissain tapauksissa offshoringia käytetään kilpailuedun saavuttamiseen; joissain tapauksessa sen käyttäminen on kilpailutilanteen sanelema pakko. (Carmel & Tjia, 2005, s. 10) Offshoring kuitenkin tarkoittaa monessa tapauksessa sitä, että projektityötä joudutaan hajauttamaan, mikä tuo mukanaan omat ongelmansa.

Ketterien menetelmien soveltamisesta hajautetuissa ympäristöissä on vain vähän dokumentoituja kokemuksia. Vaikuttaa kuitenkin siltä, että useat yritykset ovat kiinnostuneet niiden käyttämisestä, tai ovat jo alkaneet käyttää niitä. (Paasivaara, ym. 2008)

Ketterien menetelmien käyttöönotto hajautetussa ohjelmistokehityksessä ei kuitenkaan ole yksinkertaista. Ne pohjautuvat vahvasti kasvotusten tapahtuvaan kommunikaatioon, joka on harvoin mahdollista hajautetuissa projekteissa. (Paasivaara, ym. 2008) Eräät case-tutkimukset (Sutherland, ym., 2003; Paasivaara, ym., 2008) kuitenkin osoittavat, että ketterien menetelmien, kuten Scrumin ja XP:n soveltaminen hajautetuissa projekteissa on mahdollista, ja jopa hyödyllistä.

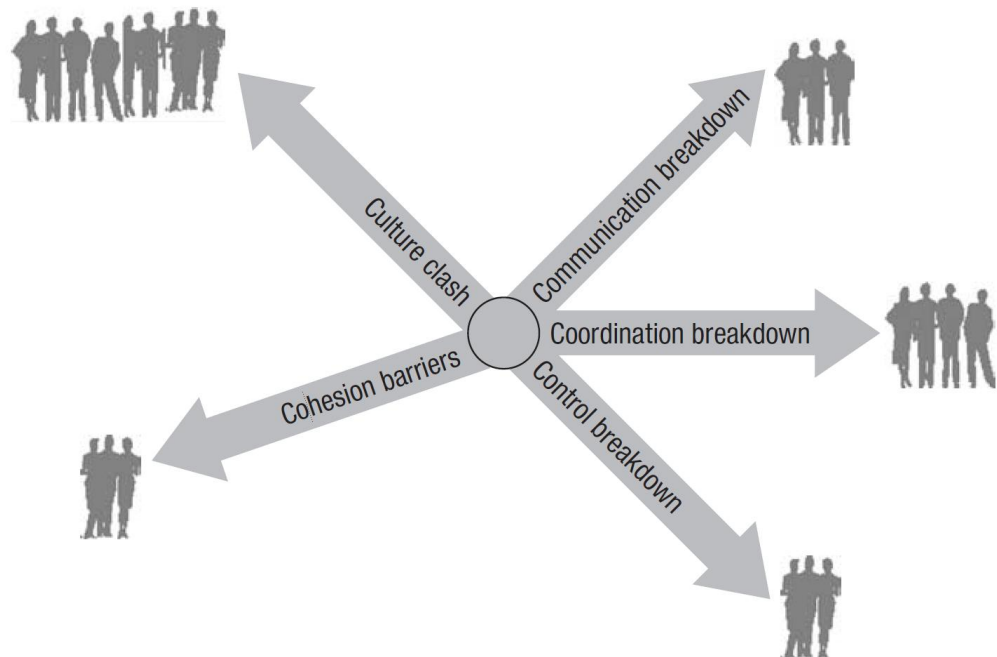
Ensimmäisessä luvussa esittelen yleisiä hajautetun ohjelmistokehityksen ongelmia, sekä niiden ratkaisuja. Toisessa luvussa esittelen case-tutkimuksia Scrumin käytöstä hajautetussa ohjelmistokehityksessä. Kolmannessa luvussa vertailen yleisiä hajautetun kehityksen ongelmia case-tutkimuksissa esiintyneisiin ongelmiin. Lisäksi analysoin ensimmäisen ja toisen luvun pohjalta Scrumin tuomia hyötyjä. Arvioin Scrumin soveltuvuutta hajautettuun ohjelmistokehitykseen, sekä vastaan tutkielman otsikon kysymykseen, onko Scrum ratkaisu hajautetun ohjelmistokehityksen ongelmiin?

2 HAJAUTTAMISEN AIHEUTTAMAT ONGELMAT

Hajautetussa ohjelmistokehityksessä on useita ongelmia. Kaikkia hajauttamisesta johtuvia ongelmia ei voida täysin ratkaista, mutta niiden vaikutukset voidaan minimoida. (Carmel & Tjia, 2005, s. 149) Useimpien ongelmien perimmäiset syyt voidaan kategorisoida Carmelin ja Tjian (2005) esittämään viiteen keskipakoisvoimaan. Tässä luvussa käsitellään hajautetun ohjelmistokehityksen ongelmia, ja pyritään sijoittamaan ne näihin kategorioihin. Lopuksi esittelen Carmelin ja Tjian (2005) ehdottamia ratkaisuja ongelmiin.

2.1 Viisi keskipakoisvoimaa

Carmen ja Tjia (2005) esittävät hajautukseen liittyvät ongelmat viiden ”keskipakoisvoiman” avulla (Kuva 1). Kukin voima etäännyttää projektin henkilöstöä toisistaan, tehden työstä vaikeampaa. Seuraavaksi kerron kunkin voiman merkityksestä, niihin johtavista syistä, sekä niiden seurauksista.



Kuva 1: The five centrifugal forces that make software work difficult. (Carmel & Tjia, 2005, s.151)

2.1.1 Communication breakdown

Onnistunut kommunikaatio koostuu viestin välityksestä, vastaanottamisesta, kuittaamisesta, ymmärtämisestä ja toimimisesta. Jos kommunikaation osapuolet ovat kaukana toisistaan, väärinymmärrysten todennäköisyys kasvaa. Väärinymmärrykset taas voivat johtaa selvennyspyynnöistä johtuviin viiveisiin, virheellisiin toteutuksiin ja virheellisistä tulkinnoista johtuviin henkilökohtaisiin konflikteihin. (Carmel & Tjia, 2005, s.152)

Syitä ja seurauksia

Aikaerot aiheuttavat ongelmia sekä kommunikaatioon että koordinaatioon. Suurin ongelma on viive vastauksissa. Mitä vähemmän päällekkäistä työaikaa projektin eri osapuolilla on, sitä vaikeampaa projektinhallinta on. (Holmström, ym. 2006) Päällekkäisen työajan ulkopuolella joudutaan turvautumaan asynkronisiin viestintävälineisiin, kuten sähköpostiin, joka taas voi pahimmillaan aiheuttaa yksinkertaistenkin asioiden ratkaisemisen kestävän päiväkausia. (Kiel, 2003)

Läheisyys johtaa sattumanvaraisiin spontaaneihin keskusteluihin ("kahvihuonekeskustelut"), jotka ovat organisaatioissa arvokkaita, sillä ne auttavat ihmisiä tutustumaan toisiinsa. (Carmel & Tjia, 2005, s.149) Spontaanit keskustelut parantavat myös ympäristötietoisuutta siitä, minkä parissa ihmiset parhaillaan työskentelevät, mitkä ovat heidän erityisosaamisalueitaan, ja missä tilassa projektin eri komponentit ovat. Näiden spontaanien keskusteluiden puute voi aiheuttaa yllättäviä tilanteita, kuten komponenttien yhteensovitusongelmia. (Herbsleb & Moitra, 2001)

Eräät maat, kuten Intia, Filippiinit ja Singapore ovat pärjänneet IT:n ulkoistamiskilvassa osittain vahvan englanninkielen osaamisensa takia. Yhteinen kieli kaventaa eri kansallisuuksien välisiä kulttuurieroja. (Carmel &

Agarwal, 2001) Mikäli eri osapuolten äidinkieli ei ole sama, voi kommunikaatio olla vaikeampaa. (Kiel, 2003)

2.1.2 Coordination breakdown

Ohjelmistokehitystyössä täytyy tehdä jatkuvasti pieniä muutoksia ja korjauksia. Koordinaatio on näiden muutosten tekemistä niin, että ne auttavat päätavoitteen saavuttamista: kysymyksiä, selvennyspyyntöjä, pieniä parannuksia, spontaania keskustelua. Kun koordinointi pettää, esimerkiksi ongelmanratkaisu voi viivästyä niin paljon että se on lopulta kallista korjata. (Carmel & Tjia, 2005, s.152)

Syitä ja seurauksia

Hajautetussa työssä koordinointi on vaikeaa, sillä normaalisti se tapahtuu pienien asioiden, kuten spontaanin keskustelun avulla. Ei-hajautetussa projektissa näiden muutosten tekeminen on helpompaa, sillä työntekijät ovat paremmin tietoisia muiden vastuualueista ja tekemisistä. (Carmel & Tjia, 2005, s.152)

Koordinoinnista saattaa tulla ongelmia myös johtuen toimipisteiden välisistä eroavaisuuksista prosesseissa. Esimerkiksi erilainen määritelmä yksikkötestauksesta voi aiheuttaa vääriä odotuksia ja konflikteja. (Mockus, ym., 2001)

2.1.3 Control breakdown

Kontrolli on prosessi joka varmistaa, että työ on tavoitteellista, sekä standardeja ja käytäntöjä noudattavaa. Kontrolli onnistuu parhaiten, kun projektipäälliköt voivat olla alaisensa keskuudessa (management by walking around). Jos projektipäälliköt eivät voi fyysisesti tavata alaisiaan, heidän täytyy käyttää teknologiaa tiedon keräämiseen saadakseen kokonaiskuvan siitä, mitä

projektissa tapahtuu. Tämä on huomattavasti vaikeampaa puhelimitse tai sähköpostitse, kuin kasvotusten. (Carmel & Tjia, 2005, s.153)

Syitä ja seurauksia

Huono kontrolli hajautetussa työssä voi johtaa päällekkäiseen työhön, ongelmien myöhäiseen havaitsemiseen ja toistuviin korjaustoimenpiteisiin. (Carmel & Tjia, 2005, s.153)

2.1.4 Cohesion barriers

Ryhmässä, jossa on hyvä koheesio (tai yhteishenki), jäsenet luottavat toisiinsa, pitävät toisistaan ja voivat auttaa toisiaan. Luottamus on hankala käsite hajautetussa työssä, sillä osapuolten on luotettava toisiinsa, jotta tehokas yhteistyö on mahdollista. Toisaalta luottamuksen rakentaminen pitkien välimatkojen takaa on hankalaa. (Carmel & Tjia, 2005, s.153)

Syitä ja seurauksia

Hajautetussa työssä on vaikea saavuttaa ryhmän sisäistä koheesiota, mikäli ryhmän jäsenet eivät ole työskennelleet aiemmin samassa paikassa. (Carmel & Tjia, 2005, s.153) Hajautetussa työssä samassa paikassa työskentely on usein mahdollista vain lyhyitä aikoja kerrallaan; eikä se tiukan budjetin projekteissa ole välttämättä mahdollista lainkaan.

2.1.5 Culture clash

Kulttuuri määrittelee yksilön periaatteet, arvot, uskomukset ja käyttäytymisen, mukaan lukien kommunikaatiokäyttäytymisen. Tämän seurauksena missä tahansa kulttuurienvälisessä kommunikaatiossa viestejä ymmärretään todennäköisemmin väärin. (Carmel & Tjia, 2005, s.153)

Kulttuureissa on eroja monissa suhteissa, kuten kurintarpeessa, suhtautumisessa hierarkkisuuteen, ajankäsitelyssä ja kommunikaatiotyylissä. (Hofstede, 2005)

Syitä ja seurauksia

Pienetkin kulttuurierot voivat johtaa ristiriitoihin, epäluottamukseen ja konflikteihin. (Carmel & Tjia, 2005, s.153)

Hajautettu ohjelmistokehitys vaatii tiivistä yhteistyötä ihmisiltä, joilla on erilaiset kulttuurilliset taustat. Hofsteden (2005) mainitsema erilainen käsitys ajasta voi johtaa konflikteihin esimerkiksi aikataulujen noudattamisen tärkeydestä. Kulttuurilliset erot voivat usein kärjistä myös kommunikaatio-ongelmia. Kun ihmiset eivät ole varmoja, kuinka vastata oudolta tuntuviin viesteihin, niihin jätetään usein vastaamatta, tai niiden perusteella muodostetaan negatiivisia mielikuvia lähettäjän luonteesta tai aikomuksista. (Herbsleb & Moitra, 2001)

2.2 Keskipakoisvoimien yhteisvaikutukset

Kukin viidestä keskipakoisvoimasta johtaa ongelmiin, ja joissain tapauksissa epäonnistuneisiin projekteihin. Herbslebin (2003) mukaan näissä epäonnistuneissa tapauksissa esiintyy erityisesti kahdentyypisiä ongelmia: hajautetut tiimit eivät pysty hallitsemaan odottamattomia tapahtumia, eivätkä ne pysty tekemään päätöksiä vaikeissa ongelmissa. Kummatkin näistä ongelmista juontavat juurensa keskipakoisvoimista. (Carmel & Tjia, 2005, s. 154)

Keskipakoisvoimien välillä vallitsee myös riippuvuuksia. Erityisesti muutokset kommunikaatiossa vaikuttavat muihin voimiin. Carmelin ja Agarwalin (2001) mukaan kommunikaatio on välittävä tekijä joka vaikuttaa sekä koordinaation, että kontrollin onnistumiseen. Heikentynyt kommunikaatio vaikuttanee negatiivisesti myös tiimin koheesioon, sekä hankaloittaa kulttuurierojen

käsitlemistä. Toisaalta hyvä kommunikaatio kompensoi puutteita muiden keskipakoisvoimien hallinnassa. Mockus:n ym. (2001) mukaan toimipisteiden välisen kommunikaation puute on globaalien ohjelmistokehityksen suurimpia ongelmia.

Herbslebin ja Mockus:n (2003) mukaan tehtävän tekeminen hajautettuna kestää keskimäärin noin 2,5 kertaa kauemmin, kuin saman tehtävän tekeminen paikallisesti. Tämä kertoo hajautetun työn olevan paikallista työtä ongelmallisempaa. Toisaalta myöhemmin käsiteltävässä J. Sutherlandin (2007) case-tutkimuksessa esitellään hajautettu Scrum-projekti, joka pääsi tuottavuudessaan selvästi paikallisesti toteutettujen ohjelmistoprojektien keskivertotuottavuuden edelle. Tämä antaa viitteitä siitä, että edellä mainitut ongelmat ovat ratkaistavissa.

Kritiikki

Kuva 1 esittää kaikki viisi keskipakoisvoimaa samanarvoisina. Esittelen kuitenkin tässä tutkielmassa useita lähteitä, jotka korostavat kommunikaation tärkeyttä ja sen vaikutuksia muihin keskipakoisvoimiin. (Vax, ym. 2008; Carmel & Agarwal, 2001) Hyvä kommunikaatio voi vaikuttaa myönteisesti muihin keskipakoisvoimiin kun taas huonosti toimiva kommunikaatio pahentaa muita hajautetun ohjelmistokehityksen ongelmia entisestään. Kuva 1 olisikin todenmukaisempi, mikäli kommunikaation roolia olisi korostettu. Kommunikaatio voitaisiin esittää ikään kuin pyramidin alimpana kerroksena, joka on kaiken muun perusta.

2.3 Ongelmien ratkaiseminen

Hajautetun projektiorganisaation rakennetta suunniteltaessa tulisi ottaa huomioon kaksi perusperiaatetta. Projektiin osallistuvien hajallaan olevien yksiköiden määrän tulisi olla mahdollisimman alhainen. Lisäksi projektin riippuvuuksien määrä tulisi pitää minimissä. (Carmel & Tjia, 2005, s. 172)

Ihanteellisessa tilanteessa toimipisteet pystyvät toimimaan mahdollisimman itsenäisesti, mutta toimipisteiden välinen kommunikaatio on silti helppoa, joustavaa ja tehokasta. (Herbsleb & Moitra, 2001)

Projektin arkkitehtuurin tulee siis olla rajapintojen osalta hyvin suunniteltua. Silti, hyväkään organisatorinen rakenne ei täysin poista riippuvuuksista aiheutuvia ongelmia, jotka aiheuttavat usein ongelmia integrointi- tai projektin päättövaiheessa. Nämä ongelmakohdat vaativat erityistä huomiota projektin johdolta. (Carmel & Tjia, 2005, s. 172)

Tämän luvun loppuosa perustuu Carmelin ja Tjian (2005) kirjaan, s. 154-159.

Carmel ja Tjia esittävät ratkaisuja viiden keskipakoisvoimansa ongelmiin. Informaaleja usein toistuvia käytäntöjä tulisi formalisoida, ja uusien "informalismien" luomiseen tulisi panostaa. Luomalla formalismeja projektin kommunikaatiosta saadaan sulavampaa ja standardoidumpaa. Formalismit vähentävät päällekkäisten yritysten ja erehdysten määrää. Informalismien luonnilla tarkoitetaan sosiaalisten suhteiden syntymisen edesauttamista. Etäiset kollegat toimivat paremmin yhdessä, mikäli heillä on jonkinlainen sosiaalinen suhde.

Kanssakäymisen rytmittäminen säännöllisillä palavereilla pakottaa projektin henkilöstön toistuvaan kanssakäymiseen, joka auttaa synkronoimaan eri toimipisteiden välistä työtä, sekä koordinoimaan sitä. Myös iteraatioiden pitäminen lyhyinä helpottaa työn synkronointia. Samalla projekti tuottaa näkyviä tuloksia usein (frequent deliverables).

Yhteisten standardoitujen viestintäkanavien ja terminologian käyttö ja dokumentointi helpottaa kommunikointia. Kommunikoinnissa on hyvä käyttää sovittua protokollaa, joka helpottaa asioiden ymmärtämisen varmistamista, sekä kiireellisyyden ilmaisua. Esimerkiksi vastausta vaativaan sähköpostiin on hyvä vastata heti, kun sen on lukenut. Mikäli asiaa ei voi hoitaa heti, voi siihen vastata: "Sain viestisi. Vastaan kysymyksiisi perjantaihin mennessä".

Projektin infrastruktuurin tulisi tukea ympäristötietoisuutta. Koko henkilöstöstä voi tehdä luettelon joka sisältää nimet, yhteystiedot, työajat, sekä täydet tiedot kunkin henkilön erityisosaamisesta, työhistoriasta, sekä nykyisistä tehtävistä ja vastuista.

Yhteishengen, eli koheesion, ja luottamuksen luominen toimipisteiden välille on tärkeää. Koheesiota voidaan edistää esimerkiksi esittelemällä työntekijät toisilleen ammatillisten meriittien kera, jotta he saavat ansaitsemansa ammatillisen kunnioituksen. Mikäli projektin budjetti sen sallii, vapaamuotoiset tapaamiset kasvotusten edistävät myös koheesiota. Yhteenkuuluvuudentunne ryhmässä paranee yhteisten epämuodollisten kokemusten myötä. Nämä kokemukset voivat olla esimerkiksi yhdessä syömistä, juomista tai pelaamista. Eräs edullisempi tapa luoda yhteishenkeä on luoda virtuaalinen "leikkikenttä" tai keskustelufoorumi. Projektin tulisi rohkaista työntekijöitään reaaliaikaiseen viestintään ja suosia säännöllistä toimipisteiden välistä matkustusta. Koheesiota voi parantaa myös pienillä merkityksettömiltä vaikuttavilla asioilla, kuten lisäämällä sähköposteihin persoonallista otetta ja lämpöä kirjoittamalla tarkoituksellisen epämuodolliseen tyyliin ja yksinkertaisesti käyttämällä hymiöitä.

3 KÄYTÄNNÖN KOKEMUKSIA HAJAUTETUISTA PROJEKTEISTA

Tässä luvussa esittelen kolme case-tutkimusta, joissa on käytetty Scrumia. Näistä kahdessa tapauksessa käytettiin myös XP:tä. Esittelyssä on lähtökohtana erityisesti se, onko Scrum koettu hyväksi menetelmäksi hajautetussa ohjelmistokehityksessä. Mainitsen myös tärkeimpiä Scrumin käytön yhteydessä esiintyneitä ongelmia.

3.1 Tuottavuutta Scrumilla ja XP:llä

Tämän osion case-tutkimus on J. Sutherlandin ym. (2007) artikkelista "Distributed Scrum: Agile Project Management with Outsourced Development Teams". Luvun 3.1 teksti perustuu kyseiseen artikkeliin, ellei toisin mainita.

Projekti koostui 56 henkilöstä, jotka oli hajautettu Utahiin, Denveriin, Coloradon, Waterlooohon (Kanada) ja Pietariin (Venäjä). Käytössä oli Scrum, mutta sitä täydennettiin XP:n käytännöillä. Kyseessä olevat yritykset olivat SirsiDyrix ja StarSoft (Venäjä). Projektissa tuotettiin lähes 700 000 riviä Javakoodia. Projekti tuotti 15,3 toimintopistettä kuukaudessa kehittäjää kohti, joka tekee siitä yhden tuottavimmista koskaan dokumentoiduista projekteista. Lukema on noin viisinkertainen hajautettujen projektien keskivertotuottavuuteen nähden. (Sutherland, ym. 2007) Vertailun vuoksi mainittakoon, että tuottavuus oli lähes samaa tasoa, kuin erään erittäin tuottavan ei-hajautetun Scrum-tiimin, joka saavutti 17,8 toimintopistettä kuukaudessa kehittäjää kohti. (Cohn, 2004)

Ongelmallisinta hajautetussa työssä oli työn synkronointi toimipisteiden välillä. Tähän ongelmaan kuitenkin löydettiin useita lievittäviä ratkaisuja. Tiimit jaettiin toiminnallisten alueiden kannalta tasaisesti kahden suurimman toimipisteen välillä (n. 4 henkilöä per toimipiste). Yhtä komponenttia siis teki yhtä moni henkilö kummassakin suuressa toimipisteessä. Lisäksi Daily Scrum -

kokousta muutettiin niin, että kukin vastasi Scrumin kolmeen kysymykseen kirjallisesti ennen kokousta. Näin jokaisella oli mahdollisuus lukea muiden vastaukset jo etukäteen. Tämä lyhensi kokousten kestoa ja helpotti kielimuurin voittamisessa. Itse kokoukset pidettiin puhelinkokouksina.

Työn synkronointia tehtiin myös muiden säännöllisten kokousten avulla: Scrum of Scrums -kokous pidettiin kerran viikossa. Sen tarkoituksena oli hallinnoida varsinaisia Scrum-tiimejä. Lisäksi käytössä oli kaksi muuta viikoittaista kokousta: arkkitehtuurikokous ja Product Ownereiden kokous. Kaikki Product Ownerit, arkkitehdit ja Scrum Masterit sijaitsivat Utahissa. Tästä johtuen korkeamman tason kokoukset voitiin pitää kasvotusten.

Synkronointia helpottivat myös lyhyet, kaksi viikkoa kestäneet sprintit. Lyhyiden sprinttien ansiosta projektin edistyminen oli koko ajan selkeästi nähtävissä. Sprinttien välilläkin koko projektin henkilöstöllä oli mahdollisuus seurata muiden työn edistymistä yhteisen globaalin build-järjestelmän, sekä yhteisten seuranta- ja raportointityökalujen ansiosta.

Muita projektin kohtaamia ongelmia olivat tehokkaiden kommunikaatiomekanismien puute, kulttuurierot käytännöissä, prosesseissa ja teknologioissa, sekä epäyhteensopivat dataformaatit, skeemat ja standardit.

3.2 Scrumin käyttö suuressa projektissa

Tämän osion case-tutkimus on M. Paasivaaran, ym. (2008) artikkelista "Distributed Agile Development: Using Scrum in a Large Project". Luvun 3.2 teksti perustuu kyseiseen artikkeliin, ellei toisin mainita.

Tutkittava organisaatio oli 40 hengestä koostuva Norjaan ja Malesiaan jakautunut Scrum-organisaatio. (Paasivaara, ym. 2008) Scrumia käytettiin Sutherlandin ym. (2007) esittämän "Totally Integrated Scrum" -mallin mukaisesti, jossa Scrum-tiimit voivat olla maantieteellisesti hajautettuja. Jokaisesta Scrum-tiimistä oli edustus Scrum of Scrumsissa, jonka avulla koordinoitiin suurempaa kokonaisuutta. Kaikki Product Ownerit ja Scrum

Masterit sijaitsivat Norjassa, yhtä Scrum Masteria lukuun ottamatta. Scrum oli ollut käytössä organisaation muissa osissa 1,5 vuotta ennen sen käyttöönottoa kyseisessä hajautetussa projektissa. Kyseinen projekti oli toiminut hajautetusti jo vuosia ennen Scrumin käyttöönottoa.

Daily Scrum -kokous pidettiin puhelinkonferenssina web-kameroiden kera. Viikoittaiseen Scrum of Scrums -kokoukseen osallistui yksi tiimin jäsen kustakin tiimistä, sekä yleensä kaikki Scrum Masterit. Kokouksessa käytiin läpi Scrumin kolme peruskysymystä kunkin tiimin näkökulmasta, sekä "Oletteko tehneet jotain joka hankaloittaa muiden tiimien toimintaa?" ja "Aiotteko tehdä jotain joka hankaloittaa muiden tiimien toimintaa?". Näiden kysymysten tavoitteena oli varmistaa integraation onnistuminen. Daily Scrum ja muut Scrum-kokoukset ratkaisivat osittain toimipisteiden välisen kommunikaation puutteen.

Sen jälkeen kun päivittäiset Scrum-kokoukset aloitettiin, myös epävirallinen kommunikaatio toimipisteiden välillä lisääntyi. Scrum-kokoukset tuovat projektin kommunikaatioon rytmiä ja säännöllisyyttä.

Yksi Scrum Master Norjasta matkusti Malesiaan ensimmäisen sprintin ajaksi. Hän työskenteli yhdessä malesialaisten kanssa Scrum-käytäntöjen alkuun saattamisessa. Lisäksi ensimmäisen puolivuotisen aikana useat norjalaiset työskentelivät Malesian toimipisteessä täysipäiväisesti. Myös tämän jälkeen matkustaminen oli yleistä; erityisesti projektin kriittisimmissä vaiheissa, kuten viimeisen sprintin aikana ennen julkaisua. Case-tutkimusta tehtäessä noin puolet kummankin toimipisteen henkilöstöstä oli käynyt toisessa toimipisteessä vähintään kerran.

Projektiin sisältyi vuosittainen virkistystapaaminen, johon kaikki osallistuivat. Viimeisin kokoontuminen oli pitkä viikonloppu Roomassa, johon sisältyi myös lyhyt tulevaisuudenkartoituskokous, sekä tiiminrakennusharjoitus. Vuosittainen tapaaminen koettiin johdon puolesta "sijoituksena tulevaisuuteen", ja se oli kaikkien mielestä onnistunut tapahtuma.

Ohjelmistokehitys on kuitenkin harvoin ongelmatonta. Seuraavaksi käsittelen tärkeimpiä esille tulleita ongelmia.

Ongelmallista projektissa oli, että yhteistä työaikaa, jolloin molempien toimipisteiden henkilöstö oli paikalla, jäi vuodenaikasta riippuen vain kaksi tai kolme tuntia.

Myöskään tietoliikenneyhteydet toimipisteiden välillä eivät olleet riittävän nopeat. Tämä hankaloitti kommunikaatiota, sillä oikeita videoneuvotteluvälineitä ei voitu käyttää, vaan oli tyydyttävä web-kameroihin, joiden kuvanlaatu jätti paljon toivomisen varaa.

Vaatimusten väärinymmärtäminen oli yleistä. Pahimmissa tapauksissa väärinymmärrys huomattiin vasta sprintin demossa. Ongelmaa lievennettiin siten, että Product Ownerit pyysivät kehittäjiä kertomaan vaatimuksista omin sanoin, ennen niiden toteuttamista. Näin voitiin varmistua, että vaatimukset oli ymmärretty oikein.

Scrumin käyttöönoton alkuvaiheissa kokoukset olivat lyhyitä, koska ihmiset eivät olleet varmoja kuinka yksityiskohtaisesti asioista tulisi puhua. Alkukankeuden jälkeen tämä ongelma lieventyi, mutta sitä esiintyi jossain määrin loppuun asti, varsinkin suunnittelukokouksissa.

Norjalaisilla oli tapana kysellä enemmän, kun taas malesialaisilla oli tapana kuunnella. Toisinaan norjalaiset joutuivat kyselemään pitkän hiljaisuuden jälkeen, vieläkö linja oli auki. Hiljaisuuden vuoksi norjalaiset eivät voineet toisinaan olla varmoja, olivatko malesialaiset ymmärtäneet asian.

Päivittäiset Scrum-kokoukset pakottivat kommunikaatioon ja lisäsivät sitä merkittävästi. Henkilöstölle tehtyjen kyselyiden perusteella kommunikaatio ja yhteistyö parantuivat Scrumin käyttöönoton jälkeen. Kommunikaation lisääntyminen helpotti väärinymmärrysten havaitsemista, sekä ongelmien ratkaisemista ennen niiden paisumista liian suuriksi.

Verrattuna Scrumia edeltävään aikaan, tuotteen laatu parani. Jokaisen sprintin päätteeksi oltiin lähellä laatua, joka aiemmin oli saavutettu vasta julkaisun lähestyessä.

Projektista jäi kaikille haastatelluille positiivinen kokonaiskuva ja Scrumin käyttöönottoon oltiin tyytyväisiä. Myös työmotivaation mainittiin parantuneen. Scrumin käyttöönoton onnistumiseen vaikutti kokenut organisaatio, pätevä henkilöstö, johdon sitoutuneisuus ja se, että organisaatiolla oli jo ennestään toimiva versionhallinta ja build-järjestelmä.

Kaikkien haastateltujen mielestä Scrum on hyvä ratkaisu moniin hajautetun ohjelmistokehityksen ongelmiin, erityisesti kommunikaation lisäämiseen ja se soveltuu hyvin hajautettuihin projekteihin.

3.3 Kokemuksia Inteliltä ja HP:lta

H. Holmströmin, ym. (2006) case-tutkimus ”Agile Practices Reduce Distance in Global Software Development” käsittelee ketterien menetelmien, erityisesti Scrumin ja XP:n käyttämistä globaalisti hajautetussa ohjelmistokehityksessä. Tutkimuksen pääkohtina olivat globaalisti hajautetun ohjelmistokehityksen ongelmat, sekä kuinka ketterät menetelmät lieventävät näitä ongelmia. Luvun 3.3 teksti perustuu kyseiseen artikkeliin, ellei toisin mainita.

Tarkastelussa olivat HP ja Intel. Kummallakin yrityksellä on useita toimipisteitä ympäri maailmaa. Toiminnot keskittyvät Yhdysvaltoihin, mutta sieltä käsin tehdään hajautettua ohjelmistokehitystä esimerkiksi Irlannin, Puolan, Kiinan, Intian ja Malesian toimipisteiden kanssa. Case-tutkimus perustuu Irlannin toimipisteissä tehtyihin johtohenkilöiden haastatteluihin, eikä käsittele mitään yksittäistä projektia.

Toimipisteiden välinen aikaero aiheuttaa yhteisen työajan vähäisyyttä. Tämä puolestaan hankaloittaa sekä kommunikaatiota, että työn kontrollointia ja koordinoitua. Erityisesti viiveet asynkronisessa kommunikaatiossa koettiin

turhauttaviksi. Toisaalta useat kehittäjät olivat valmiita siirtämään työaikojaan päällekkäisen työajan lisäämiseksi.

Toimipisteiden väliset suhteet olivat hyvät johdon tasolla, mutta kehittäjät eivät yleensä tunteneet toisiaan. Tästä syystä hajautetut tiimit eivät useinkaan tunteneet olevansa osa yhtä tiimiä. Scrumin suunnittelukäytännöt kuitenkin koettiin hyödyllisiksi yhteishengen parantamisessa.

Kulttuurillisista asioista erityisesti kielelliset ongelmat nousivat esiin. Ongelmia esiintyi sekä sanavaraston, että tulkinnan osalta. Sekä Intelillä, että HP:lla oli koettu paljon väärinymmärryksiä ja sekaannuksia, jotka olivat johtuneet ongelmista ymmärryksessä. Tällaiset väärinymmärrykset vaikuttavat kommunikaation, koordinaation ja kontrollin onnistumiseen, ja vaikeuttavat yhteisymmärryksen muodostumista tiimien välille.

Scrum tai XP eivät olleet kummassakaan yrityksessä käytössä ”puhtaina” muotoinaan, vaan niiden tarjoamia keinoja käytettiin niiltä osin, kun ne nähtiin soveltuviksi. Useimmat XP:n käytännöt, kuten pariohjelmointi ja refaktorointi, kuitenkin olivat käytössä sekä Intelillä, että HP:lla, ja ne koettiin toimiviksi myös hajautetussa työssä. Myös Scrum todettiin hyödylliseksi erityisesti suunnittelussa ja projektin seurannassa.

Yhteenvetona todettiin, että ketterät menetelmät lisäävät kommunikaation määrää. Scrumin suunnittelu- ja seurantakäytännöt taas ajoivat kehitystyötä yhteistyöorientoituneempaan suuntaan. Sosiokulttuuralliset eroavaisuudet nähtiin ongelmana, jota ketteryys, tai mikään muukaan yksittäinen metodologia ei yksin voisi ratkaista. Ketteryyden kehittäjäkeskeisyys kuitenkin sai ihmiset osallistumaan ja kommunikoimaan enemmän. Lopuksi todettiin, että ketterät menetelmät soveltuvat globaalisti hajautettuun ohjelmistokehitykseen paremmin, kuin aiempien tutkimusten perusteella on luultu.

4 ANALYYSI

4.1 Formalisointia ja informalisointia

Carmel ja Tjia (2005) ehdottivat hajauttamisen ongelmien ratkaisemiseksi formalisoinnin ja informalisoinnin periaatteita. Näihin periaatteisiin sopivia ratkaisuja löytyi case-tutkimuksista useita. Jo Scrum itsessään sisältää piirteitä, jotka ovat Carmelin ja Tjian tarkoittamaa formalisointia. Esimerkiksi Daily Scrum -käytäntö on kommunikaation rytmittämistä, ja sitä sovellettiin kaikissa Scrumia käyttäneissä tapauksissa. (Sutherland, ym. 2007; Paasivaara, ym., 2008; Holmström, ym., 2006)

Vastaavia hyötyjä formalisoinnista saavutettiin myös esimerkiksi Vaxin, ym., (2008) case-tutkimuksessa, jossa mainittiin onnistuneen kommunikaation olleen kriittinen menestystekijä. Onnistuminen saavutettiin noudattamalla tiukkaa kommunikaatiosuunnitelmaa.

SirsiDynixin tapauksessa Daily Scrumin formalisointi vietii vielä pidemmälle jakamalla Scrumin kolmen peruskysymyksen vastaukset kirjallisesti ennen kokousta. Tällä toimenpiteellä onnistuttiin lisäksi lieventämään kielimuurin aiheuttamia ongelmia. (Sutherland, ym. 2007)

Eräänlaista formalisointia on myös Scrumin iteratiivinen käytäntö, jossa jokaisen sprintin päätteeksi nähtävillä on toimivaa ohjelmistoa. Sutherlandin, ym. (2007) case-tutkimuksessa todettiin lyhyiden kahden viikon sprinttien helpottaneen toimipisteiden välistä synkronointia.

Paasivaaran, ym. (2008) case-tutkimuksessa projektin henkilöstö matkusti säännöllisesti toiseen toimipisteeseen. Työmatkojen lisäksi projektilla oli vuosittainen virkistystapaaminen, johon kaikki osallistuivat. Matkustamisen osalta ilmeni siis sekä formalisointia (säännöllisyys), että informalisointia (suhteiden luominen). Matkustaminen parantaa ryhmän koheesiota, sillä se auttaa luomaan henkilökohtaisia suhteita ihmisten välille (Mockus, ym., 2001).

4.2 Havainnot ja johtopäätöksiä

Eräissä case-tutkimuksissa Scrum lisäsi kommunikaatiota merkittävästi Scrumia edeltäneeseen aikaan verrattuna, ja suuntasi sen ongelmanratkaisuun. Lisääntynyt kommunikaatio helpotti väärinymmärrysten havaitsemista aiemmassa vaiheessa. Ketterät menetelmät nähtiin hyvin soveltuvana hajautettuun kehittämiseen. (Paasivaara, 2008)

Ketteryyden hyödyllisyyttä hajautetussa ohjelmistokehityksessä tukee myös Danait:n (2005) case-tutkimus, sekä Vaxin, ym. (2008) case-tutkimus.

Kaikkia ongelmia Scrum ei kuitenkaan ratkaise. Paasivaaran case-tutkimuksessa vaatimusten väärinymmärtäminen oli yleistä. Pahimmissa tapauksissa väärinymmärrys huomattiin vasta sprintin demossa. (Paasivaara, 2008) Kaikissa tässä tutkielmassa esitellyistä case-tutkimuksista törmättiin kulttuurieroista johtuviin ongelmiin. Holmströmin, ym. (2006) tutkimuksessa todettiin, ettei Scrum, tai mikään muukaan yksittäinen menetelmä voisi ratkaista globaalin ohjelmistokehityksen sosiokulttuurialaisia ongelmia. Myös toimipisteiden maantieteellisten sijaintien havaittiin aiheuttavan kommunikaatio-ongelmia; eräissä tapauksissa projektin toimipisteillä oli vain muutama tunti yhteistä työaika. (Paasivaara, ym. 2008) Yhteisen työajan puute taas pakottaa asynkronisten kommunikaatiovälineiden käyttöön.

Kahdessa case-tutkimusten Scrum-projektissa yhdistäviä tekijöitä olivat useimpien Scrum Mastereiden ja Product Ownereiden sijaitseminen samassa paikassa. Carmel ja Tjia esittävät useisiin lähteisiin vedoten, että mitä monimutkaisempia tehtäviä tehdään, sitä suurempi on tarve kasvatusten tapahtuvalle kommunikaatiolle. (Carmel & Tjia, 2005, s. 152) Tästä syystä projektin johtohenkilöiden pitäminen yhdessä toimipisteessä voi olla hyödyllistä.

Kahta projektia yhdisti myös se, että kummassakin käytettiin kaikille yhteistä versionhallintajärjestelmää sekä seuranta- ja raportointityökalua. (Paasivaara, ym., 2008; Holmström, ym., 2006)

Holmströmin (2006) mukaan Scrumin ja XP:n käytännöt ovat hyödyllisiä kommunikaation, koordinaation ja kontrollin parantamisessa hajautetuissa projekteissa. Myös ryhmän koheesio parani merkittävästi Scrumin suunnittelukäytäntöjen ansiosta. Ketteryyden kehittäjäkeskeisyyden havaittiin kaventavan myös sosiokultuuraalista kuilua etäisten toimipisteiden välillä. (Holmström, 2006)

Edellä mainittujen väittämien valossa voidaankin sanoa, että ketterät menetelmät auttavat kaikissa Carmelin ja Tjian viiden keskipakoisvoiman ongelmissa.

Puhuttaessa hajautetusta ohjelmistokehityksestä, ajatellaan helposti erityisesti globaalia ohjelmistokehitystä. Herbslebin ja Moitran (2001) mukaan etäisyyksien ei kuitenkaan tarvitse olla suuria, jotta hajauttamisen ongelmat tulevat esille. Jo vierekkäisissä rakennuksissa, tai jopa saman rakennuksen eri kerroksissa työskenteleminen vähentää kommunikaatiota merkittävästi verrattuna tilanteeseen, jossa tiimit työskentelevät vieretysten. Näin ollen ratkaisut, jotka auttavat globaalisti hajautettuja tiimejä, voivat auttaa myös paljon pienempien etäisyyksien erottamia tiimejä. Olisikin jatkotutkimuksen arvoista selvittää, toimisivatko hajautettua ohjelmistokehitystä varten tehdyt pienet muutokset Scrumin käytännöissä myös paikallisissa Scrum projekteissa.

Tämä tutkielma vahvistaa yleistä käsitystä siitä, että toimiva kommunikaatio on hajautetun ohjelmistokehityksen tärkeimpiä osa-alueita. Yksin kommunikaation toimivuus tai toimimattomuus voi ratkaista projektin onnistumisen tai epäonnistumisen. Lisäksi olen osoittanut Scrumin parantaneen kommunikaation toimivuutta hajautetussa ohjelmistokehityksessä.

Tässä tutkielmassa esitetyn aineiston perusteella voidaan sanoa Scrumin käytön lievittävän useimpia hajautetun ohjelmistokehityksen ongelmia. Esimerkiksi

kulttuurierot kuitenkin ovat ongelma, johon Scrum tarjoaa vain vähän lievitystä. Aihe kaipaakin jatkotutkimusta erityisesti niiltä osin, voitaisiinko Scrumia kehittää tarjoamaan parempaa tukea hajautettuun ohjelmistokehitykseen, johon osallistuu eri kulttuurien edustajia.

YHTEENVETO

Tutkielmassa tutkittiin hajautetussa ohjelmistokehityksessä esiintyneitä ongelmia. Aluksi esiteltiin yleisiä hajautetun ohjelmistokehityksen ongelmia ja niiden ratkaisuja. Seuraavaksi esiteltiin käytännön kokemuksia hajautetusta Scrumista kolmen case-tutkimuksen avulla. Lopuksi vertailtiin yleisiä hajautetun kehityksen ongelmia case-tutkimuksissa esiintyneisiin ongelmiin, sekä arvioitiin Scrumista saatavia hyötyjä hajautetussa ohjelmistokehityksessä.

Esittelin Carmelin & Tjian viiden keskipakoisvoiman periaatteen, joka kuvaa hajautettua ohjelmistokehitystä hankaloittavia voimia. Voimat ovat kommunikaatio, koordinaatio, kontrolli, koheesio, sekä kulttuuri.

Totesin Scrumin lieventävän useita hajauttamisesta johtuvia ongelmia. Scrum parantaa kommunikaatiota sekä eri toimipisteiden välillä, että yhden toimipisteen sisällä. Se myös helpottaa projektin kontrollointia ja koordinoitua. Eräiden Scrumin käytäntöjen, kuten suunnittelukokousten, havaittiin parantavan myös toimipisteiden välistä yhteishenkeä, eli koheesiota. Scrumin todettiin jossain määrin lieventävän myös kulttuurieroista johtuvia ongelmia.

Hajautettujen ohjelmistokehitysprojektien mainittiin olevan keskimääräiseltä tuottavuudeltaan selvästi paikallisia projekteja huonompia. Case-tutkimusten perusteella voidaan kuitenkin sanoa, että Scrumia käyttämällä voidaan päästä hyvään tuottavuuteen hajautetuissakin projekteissa.

Yhteenvetona voidaan siis sanoa, että Scrum auttaa useimpien hajauttamisen aiheuttamien ongelmien voittamisessa, ja soveltuu hyvin hajautettuun ohjelmistokehitykseen, vaikka se ei kaikkia ongelmia ratkaisekaan. On kuitenkin muistettava, että Scrum perustuu vahvasti kasvotusten tapahtuvaan kommunikaatioon. Sitä on siis muokattava hajauttamisen rajoitteiden vuoksi, jotta se soveltuu hajautettuun ympäristöön. Scrumin muokkaaminen siten, että se tukisi paremmin hajautettua monikulttuurista ohjelmistokehitystä onkin hyvä jatkotutkimuksen aihe.

Lopuksi todettiin joidenkin hajauttamisen ongelmien tulevan esille jo pienillä etäisyyksillä. Ratkaisut jotka auttavat globaalissa ohjelmistokehityksessä - kuten Scrumin käyttö - voivat siis auttaa pienempienkin etäisyyksien erottamia tiimejä.

LÄHTEET

Carmel, E. & Agarwal, R. 2001. Tactical approaches for alleviating distance in global software development. *IEEE Software* 18(2), 22-29.

Carmel, E. & Tjia, P. 2005. *Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce*. Cambridge: Cambridge University Press.

Cohn, M. 2004. *User Stories Applied for Agile Development*. Boston: Addison-Wesley.

Herbsleb, J.D., Mockus, A., Finholt, T.A. & Grinter, R.E. 2001. An Empirical Study of Global Software Development: Distance and Speed. *Teoksessa Proceedings of the 23rd International Conference on Software Engineering*. Toronto, Ontario, Canada, May 12-19. Los Alamitos: IEEE Computer Society, 81-90.

Herbsleb, J.D. & Moitra, D. 2001. Global software development. *IEEE Software*. 18(2), 16-20.

Herbsleb, J.D. & Mockus, A. 2003. An empirical study of speed and communication in globally distributed software development. *IEEE Transactions on Software Engineering* 29(6), 481-494.

Herbsleb, J.D. 2003. Luentokalvot, *The International Workshop in Global Software Development*. International Conference on Software Engineering. 2003.

Hofstede G.H. 2005. *Cultures and Organizations: Software of the Mind – Intercultural Cooperation and Its Importance for Survival*. 2. painos. McGraw-Hill, New York, s. 277.

Holmström, H., Fitzgerald, B., Ågerfalk, P.J. & Conchúir, E.Ó. 2006. Agile Practices Reduce Distance in Global Software Development. *Information Systems Management* 23(3), 7-18.

- Kiel, L. 2003. Experiences in distributed development: A case study. ICSE'03, International Conference on Software Engineering, Portland, Oregon.
- Mahnic, V. & Drnovscek, S. 2005. Agile Software Project Management with Scrum. Teoksessa Clark, M.J. (toim.) EUNIS 2005: Conference Proceedings, Manchester, United Kingdom, June 21-24. Manchester: Manchester Computing
- Mockus, A. & Herbsleb, J. 2001. Challenges of Global Software Development. Teoksessa Proceedings, Seventh International Software Metrics Symposium: METRICS 2001, London, England, April 4-6. Los Alamitos: IEEE Computer Society, 182-184.
- Paasivaara, M., Durasiewicz, S. & Lassenius, C. 2008. Distributed Agile Development: Using Scrum in a Large Project. Teoksessa IEEE International Conference on Global Software Engineering, 2008. ICGSE 2008, Bangalore, India, August 17-20. Los Alamitos: IEEE Computer Society Press, 87-95.
- Sutherland, J., Viktorov, A., Blount, J. & Puntikov, N. 2007. Distributed Scrum: Agile Project Management with Outsourced Development Teams. Teoksessa Sprague, R.H., Jr (toim.) Proceedings of the 40th Annual Hawaii International Conference on System Sciences, Big Island, Hawaii, January 3-6. Los Alamitos: IEEE Computer Society Press, 274.
- Sutherland, J., Schoonheim, G., Rustenburg, E. & Rijk, M. 2008. Fully Distributed Scrum: The Secret Sauce for Hyperproductive Offshored Development Teams. Teoksessa Melnik, G. (toim.) Proceedings of the Agile 2008, Toronto, Ontario, Kanada, August 4-6. Los Alamitos: IEEE Computer Society Press, 339-344.