



Vaasan yliopisto  
UNIVERSITY OF VAASA

Weckman Teppo

## **Kansainvälinen ohjelmistokehitys**

Ongelmat projektinhallinnassa ja hajautetuissa tiimeissä

Tekniikan ja innovaatiojohtamisen yksikkö  
Tietojärjestelmätieteen pro gradu -tutkielma  
Digitaalisen liiketoiminnan kehittäminen

Vaasa 2022

---

**VAASAN YLIOPISTO****Tekniikan ja innovaatiojohtamisen yksikkö**

<b>Tekijä:</b>	Weckman Teppo	
<b>Tutkielman nimi:</b>	Kansainvälinen ohjelmistokehitys : Ongelmat projektinhallinnassa ja hajautetuissa tiimeissä	
<b>Tutkinto:</b>	Kauppätieteiden maisteri	
<b>Oppiaine:</b>	Digitaalisen liiketoiminnan kehittäminen	
<b>Työn ohjaaja:</b>	Tomi Pasanen	
<b>Valmistumisvuosi:</b>	2022	<b>Sivumäärä:</b> 66

---

**TIIVISTELMÄ:**

Globalisaatio on prosessi, joka on jo vuosia vaikuttanut voimakkaasti yhteiskunnan kehitykseen. Ohjelmistokehitys ei tässä trendissä muodosta poikkeusta. Ohjelmistokehityksessä globalisaatio näkyy suunnittelu- ja kehitystyön, etenkin ohjelmistojen toteutuksen ja testauksen, hajautumisena useamman valtion alueelle. Kyse on pienimmillään siitä, että työ suoritetaan eri maassa, kuin mistä se tilataan. Laajimmillaan hajautuksessa työtä suoritetaan useassa eri maassa tai kohteessa. Tässä työssä tarkastellaan teknologisen kehityksen mahdollistamaa kansainvälistä ja hajautettua ohjelmistokehitystä. Kansainvälinen ohjelmistokehitys on lisääntynyt jo vuosia ja tiettyillä toimialoilla sen voi ajatella olevan jo alan yleinen normi. Kansainvälisesti hajautetulla ohjelmistokehityksellä saavutetaan monia etuja, mutta siihen liittyy myös vaikeuksia, joita ei aina osata ottaa riittävästi huomioon. Tämän takia hajautettujen ohjelmistokehitysprojektien epäonnistumisprosentti on korkea verrattuna paikallisesti toteutettuihin kehitysprojekteihin. Työn tarkastelun näkökulmana on projektinhallinta kansainvälisessä sovelluskehityksessä sekä hajautettujen tiimien toiminta. Työssä selvitetään, millaisia ongelmia kyseisten projektien hallintaan sekä virtuaalitiimien toimintaan liittyy ja miten ongelmien vaikutuksia voi minimoida tai poistaa.

Työ on toteutettu narratiivisena kirjallisuuskatsauksena. Aineiston haussa on käytetty Internetistä löytyviä hakutietokantoja, joista on ollut luontevaa olettaa löytyvän tietojärjestelmätieteen liittyviä tutkimuksia ja artikkeleita. Aineistohaku on rajattu englanninkieliseen aineistoon. Hakusanoina tietokannoissa on käytetty kansainväliseen ja hajautettuun sovelluskehitykseen ja virtuaalitiimeihin liittyviä sanoja.

Merkittävimpinä hyötyinä ohjelmistokehityksen kansainvälistymisessä nähdään muun muassa siitä saatavat kustannussäästöt sekä mahdollisuus rekrytoida osaavaa työvoimaa. Kansainvälisesti hajautetun ohjelmistokehityksen esteet liittyvät työskentelyn maantieteelliseen, ajalliseen, kulttuurilliseen ja organisatoriseen etäisyyteen. Nämä etäisyydet monimutkaistavat hajautetun tai virtuaalisen tiimin toimintaa verrattuna paikallisesti työskentelevään tiimiin. Esteisiin on pyrittävä vastaamaan sopeuttamalla käytetyt projektinhallinnan menetelmät hajautettuun työskentelyyn. Erityistä huolta on pidettävä siitä, että hajautetun työryhmän roolit ja sisäinen vastuunjako ovat kaikille ryhmän jäsenille selkeitä. Jos hajautetun työryhmän jäsenet tulevat erilaisilta kielialueilta ja kulttuurisista taustoista, on ryhmän johtamisessa varmistettava, että kaikki osallistujat jakavat yhteisen vision tehtävästä ja kommunikaatio hoidetaan niin, että väärinkäsitykset pystytään minimoimaan. Kansainvälisesti hajautetulla ohjelmistokehityksellä saavutetun kokonaishyödyn laskeminen on vaikeaa, koska osa kustannuksista on piileviä ja sen takia hankalia vertailla saavutettuihin hyötyihin. Kansainvälisellä ohjelmistokehityksellä katsotaan saavutetun merkittäviä hyötyjä, joskin hyödyt ovat monesti tavoiteltuja vaatimattomampia. Työssä tunnistetut ongelmat vastaavat keskeisiä projektinhallinnan yleisiä osa-alueita. Odotetusti hajautettujen tiimien keskeiset vaikeudet liittyivät tiedonvälitykseen ja kommunikointiin.

---

**AVAINSANAT:** Offshore, GSD, Ohjelmistokehitys, Alihankinta, Ulkoistus, Virtuaalitiimi

## Sisällys

1	Johdanto	6
1.1	Työn tavoite ja rajaus	6
1.2	Aikaisempi tutkimus	7
1.3	Tutkimusmenetelmät	7
1.4	Työn rakenne	9
2	Globaali ohjelmistokehitys	10
2.1	Historia	12
2.2	Nykytilanne	12
2.3	Hyödyt	13
2.4	Haitat	16
2.5	GSD-prosessi	20
2.5.1	Ketterät menetelmät	20
2.5.2	Perinteinen ohjelmistokehitys	22
3	Projektinhallinta globaalissa kontekstissa	24
3.1	Globaali projektinhallinta	24
3.1.1	Projektiryhmän jäsenet	26
3.1.2	Projektirakenne	28
3.2	GSD projektin hallinta	30
3.2.1	Integraation hallinta	31
3.2.2	Tietämyksen hallinta	32
3.2.3	Laajuuden ja resurssien hallinta	35
3.2.4	Suorituksen johtaminen	36
3.3	Riskienhallinta ja toteutettavuus	39
3.3.1	Riskienhallinnan viitekehyksiä	41
3.3.2	Riskienhallinta ja GSD	43
3.4	Kulttuurilliset seikat	44
3.4.1	Kulttuurien väliset erot	44
3.4.2	Kommunikaatio	45

4	Virtuaalitiimit	47
4.1	Kommunikaatio	48
4.2	Hajautetun tiimin rakenne	48
4.3	Virtuaalitiimin johtaminen	50
4.4	Virtuaalisuuden aste ja etäisyys	52
5	Johtopäätökset	56
5.1	Tutkimuksen luotettavuus	57
5.2	Jatkotutkimuksen aiheita	58
	Lähteet	59

## Kuviot

Kuvio 1.	Keskitetty projektirakenne.	29
Kuvio 2.	Hajautettu projektirakenne.	30
Kuvio 3.	GSD-projektinhallinnan viitekehys.	31
Kuvio 4.	GSD Tietämyksen hallinta.	33
Kuvio 5.	Riskienhallinnan pääprosessit.	42
Kuvio 6.	Riskienhallinta projektinhallinnassa.	43
Kuvio 7.	Virtuaalitiimi.	49
Kuvio 8.	Löyhästi kytketty tiimi.	49
Kuvio 9.	Virtuaalitiimin johtamisen tärkeimmät osatekijät.	50
Kuvio 10.	Virtuaalitiimin yleisimmät ulottuvuudet.	53

## Taulukot

Taulukko 1.	Hajautuksen ja yhteistyön mallit.	11
Taulukko 2.	GSD:n hyödyt.	15
Taulukko 3.	GSD:n hyötyjen toteutuminen.	18

## Lyhenteet

CSD	Collaborative Software Development (Yhteistyö ohjelmistokehitys)
DSD	Distributed Software Development (Hajautettu ohjelmistokehitys)
FDD	Feature-driven development
GSD	Global Software Development (Globaali ohjelmistokehitys)
GSE	Global Software Engineering (Kansainvälinen ohjelmistosuunnittelu)
SAFe	Scaled Agile Framework
XP	Extreme Programming

# 1 Johdanto

Globalisaatio on ollut vallitseva trendi yhteiskunnassamme jo vuosien ajan. Se vaikuttaa kaikkiin elämän osa-alueisiin, myös ohjelmistokehitykseen. Ohjelmistokehityksen osalta globalisaatiota ovat vauhdittaneet teknologian kehittyminen, halpeneminen sekä tiedonsiirtonopeuksien kasvu. Nyt eri puolilla maailmaan työskentelevien henkilöiden on mahdollista kommunikoida keskenään käytännössä ilman viiveitä.

Koska yhä useampi ohjelmistoalalla toimiva organisaatio hakee menestystä kansainvälisyydellä ja hajauttamalla tai ulkoistamalla toimintaansa, on tärkeää ymmärtää, mitkä tekijät vaikuttavat kansainvälisten projektien onnistumiseen ja mitkä seikat toisaalta vähentävät todennäköisyyttä onnistua. Jain ja Suman (2018) esittävät, että merkittävä osa kansainvälisistä ohjelmistoprojekteista ei ole täyttänyt niille asetettuja vaatimuksia. Näin ollen on keskeistä selvittää, miten kansainvälisissä ohjelmistoprojekteissa onnistumisen todennäköisyyttä voi kasvattaa ja miten sen pahimpia ongelmakohtia on mahdollista välttää.

Työskentelen itse yrityksessä, jossa tehdään töitä kansainvälisissä tiimeissä ja yhteistyön on onnistuttava yli maantieteellisten ja kulttuurillisten rajojen. Kyseiseen aiheeseen on itselläni myös omakohtainen tuntuma ja kiinnostus.

## 1.1 Työn tavoite ja rajaus

Työn tarkastelun kohteena on projektinhallinta sekä hajautettujen tiimien toiminta kansainvälisessä ja hajautetussa ohjelmistokehityksessä. Työssä pyritään selvittämään, millaisia ongelmia kansainvälisten ohjelmistoprojektien projektinhallintaan sekä virtuaalitiimien toimintaan liittyy. Lisäksi pyritään selvittämään, miten työssä tunnistettujen ongelmien vaikutuksia voisi minimoida tai poistaa kokonaan.

Ensisijaisina tutkimuskysymyksinä ovat: Mitkä ovat kansainväliseen/ hajautettuun ohjelmistokehitykseen liittyvän projektinhallinnan vaikeudet? Mitkä ovat virtuaalitiimiin/ hajautettuun tiimin liittyvät vaikeudet? Toissijaisena kysymyksenä on, miten näitä vaikeuksia voi minimoida.

## **1.2 Aikaisempi tutkimus**

Kansainvälisestä ohjelmistokehityksestä ja siihen liittyvästä projektinhallinnasta on tehty melko paljon tutkimusta. Tutkimukset yleisesti käsittelevät kuitenkin vain jotain projektinhallinnan osa-alueita. Project Management Instituten (PMI) Project Management Body of Knowledge (PMBOK) käsittelee projektinhallintaa laaja-alaisesti, mutta sekään ei huomioi kansainvälisen tai hajautetun ohjelmistokehityksen vaatimuksia (Jain & Suman, 2018).

## **1.3 Tutkimusmenetelmät**

Työ on toteutettu narratiivisena eli kuvailevana kirjallisuuskatsauksena. Kangasniemi ja muut (2013) mukaan narratiivisessa kirjallisuuskatsauksessa valittua aihetta tarkastellaan ja analysoidaan valitun kirjallisuuden pohjalta. Nimensä mukaisesti, narratiivisessa kirjallisuuskatsauksessa tuotetaan valitun aineiston pohjalta laadullinen, kuvaileva vastaus tutkimuskysymyksiin. Varsinaisen työn vaiheistin tutkimuskysymyksen valintaan, aineiston valintaan, aineiston käsittelyyn sekä tulosten analysointiin.

Aineiston haussa olen käyttänyt seuraavia hakutietokantoja:

- Academic Search Elite
- ACM Digital Library
- IEEE Xplore
- Research Gate
- Science Direct

- SpringerLink
- Wiley Online Library

Lisäksi olen käyttänyt tietyissä tilanteissa Google Scholar-hakupalvelua.

Aloitin aineiston haun tekemällä haun seuraaviin hakutietokantoihin:

- ACM Digital Library
- IEEE Xplore
- Research Gate
- Science Direct.

Hakutietokannoissa käytin seuraavia haku-ehdotuksia:

- ("global software development" or "distributed software development") and ("Risk management" or "quality" or "project management")
- ("global software development" or "distributed software development") and ("problem" or "challenge")
- "virtual team"
- "distributed team"

Rajasin haku-ehdot englanninkielisiin artikkeleihin. Artikkelien julkaisuvuodeksi rajasin välin 2015–2020. Näin löydettyistä artikkeleista lähdin liikkeelle. Ensimmäisessä vaiheessa hakutietokannoista löytämistäni artikkeleista olen pureutunut vielä syvemmälle tutkimalla lähteitä, joihin ko. artikkeleissa viitataan. Käyttämäni artikkelit olen valinnut sen perusteella, miten hyvin ne mielestäni osuvat käsiteltävään aiheeseen. Tässä vaiheessa hyväksyin mukaan artikkeleita, jotka eivät kuuluneet julkaisuvuoden osalta alkuperäisen rajauksen piiriin.



## 1.4 Työn rakenne

Ensimmäisen luvun, johdannon, jälkeen käyn luvussa kaksi läpi yleisesti globaalia ohjelmistokehitystä. Siinä esittelen, mitä kyseisellä käsitteellä tarkoitetaan. Yleisellä tasolla esitellään myös siitä saatavat hyödyt ja ilmeisimmät siihen liittyvät haitat.

Luku kolme käsittelee kansainvälistä ohjelmistokehitystä projektinhallinnan näkökulmasta. Luvussa käydään läpi projektinhallinnan eri osa-alueita ja hajautetun ohjelmistokehityksen vaikutuksia niihin.

Luvussa neljä käsitellään virtuaalitiimejä ja niihin liittyviä vaikeuksia. Luvussa esitellään erilaiset tavat modularisoida virtuaalitiimi sekä tavat, joilla virtuaalitiimi voidaan hajauttaa.

Luku viisi sisältää yhteenvedon ja kehitysehdotuksia jatkotutkimukseksi.

## 2 Globaali ohjelmistokehitys

Yleisesti ohjelmistojen kehitysprosessi koostuu ohjelmiston rakentamiseksi tarvittavista toimista ja tehtävistä. Vaikka PMBOKin tarjoama projektikehityksen viitekehys ei suoraan ole Jainin ja Sumanin (2018) mukaan sovellettavissa globaaliin ohjelmistokehitykseen (Global Software Development, jatkossa GSD), se tarjoaa kuitenkin hyvän perusrungon yleiselle ohjelmistokehityksen lisäksi myös GSD:lle. Näin ollen voidaan katsoa, että ohjelmistokehitysprosessin tavoitteena on myös GSD:ssä tuottaa laadukkaita ohjelmistoja budjetin ja aikataulun rajoissa. Prosessi ohjeistaa ohjelmiston kehitysprojektiin osallistuvat henkilöt suorittamaan vaadittavat tehtävät, jotta ohjelmisto saadaan toteutettua. (Project Management Institute, 2013, s. 3–6; Jain & Suman, 2018)

Globalisaatiolla on vahva vaikutus kaikkeen, myös ohjelmistokehitykseen. GSD:hen liittyy omia vaatimuksiaan, joihin on vastattava, jotta siinä on mahdollista menestyä. GSD:n erityiset tarpeet erottavat sen keskitetyimmistä sovelluskehityksestä. Näiden tarpeiden takia prosessit, joita GSD:ssä sovelletaan, esimerkiksi perinteisemmät kehitysmenetelmät tai ketterät menetelmät, on räätälöitävä siihen sopiviksi. (Jain & Suman, 2015)

Kansainvälisen ohjelmistokehityksen termit eivät kuitenkaan ole alan kirjallisuudessa täysin vakiintuneet. Esimerkiksi Sangwan ja muut (2006, s. 3–4) määrittelevät GSD:n työskentelyksi, jossa ohjelmistokehitystä tekevät tiimit toimivat maantieteellisesti eri paikoissa. Yhteistyötä tekevät tiimit voivat myös kuulua samaan tai eri organisaatioihin. Sen sijaan ul Haq ja muut (2011) mukaan GSD on asiakkaan ja toimittajan välinen sopimussuhde, jossa asiakas ulkoistaa kaiken tai osan ohjelmistokehityksestä toimittajalle, joka tarjoaa sovitut palvelut sovittua korvausta vastaan.

Muita GSD:n lisäksi käytettyjä termejä ovat muun muassa, hajautettu ohjelmistokehitys (Distributed Software Development, DSD), kansainvälinen ohjelmistosuunnittelu (Global Software Engineering, GSE) ja yhteistyö ohjelmistokehitys (Collaborative Software Development, CSD). Näitäkään termejä ei kirjallisuudessa käytetä aivan yksiselitteisesti. (Sharma ja muut, 2015; Denhere ja muut, 2015). Lisäksi käytössä ovat termit offshore ja

onshore. Offshore tarkoittaa sitä, että työ tehdään jossain toisessa maassa ja onshorella tarkoitetaan tilannetta, jossa työ tehdään samassa maassa, jossa se tilattiin (El Bajta ja muut, 2018).

Prikladnicki ja Audy (2010) esittävät loogisen jaottelun DSD:lle, GSD:lle, offshorelle ja onshorelle sekä ulkoistamiselle (outsourcing). Offshore ei tarkoita sitä, että työ olisi ulkoistettu kolmannelle osapuolelle. Myös sama yritys voi teettää työtä offshorena ilman ulkoistusta, omassa ulkomailla sijaitsevassa toimipisteessään. Eli offshore-työtä voi teettää ilman ulkoistusta ja ulkoistaa voi niin, ettei työtä tehdä offshorena. Käytännössä GSD on offshore-työtä. Prikladnicki ja Audy (2010) määrittelevät hajautetun ohjelmistokehityksen (DSD) niin, että siinä tiimin jäsenet työskentelevät maantieteellisesti hajautettuina. Globaalissa ohjelmistokehityksessä (GSD) hajautuksen mittakaava on suurempi ja hajautuksen taso vähintäänkin kansainvälinen. Taulukossa 1 on selvennetty termistöä maantieteellisen etäisyyden ja organisaatioiden suhteen.

**Taulukko 1. Hajautuksen ja yhteistyön mallit.**

		Organisaatio	
		Sama organisaatio	Eri organisaatio
Maantieteellinen etäisyys	Sama fyysinen paikka	Keskitetty työskentely	Keskitetty työskentely + ulkoistus
	Sama maa	DSD	DSD + ulkoistus
	Eri maa	GSD/ offshore	GSD/ offshore + ulkoistus

## 2.1 Historia

Kansainvälisten kehitysryhmien toteuttamilla ohjelmistoilla on pitkä historia. Jo 1960-luvulla IBM teki ohjelmistokehitystä hajautetusti useammassa eri maantieteellisessä paikassa ja jopa useammalla mantereella (Cusumano, 2008).

Suuret kansainväliset trendit, kuten globalisaatio ja tietoteknisen koulutustason nousu ympäri maailmaa, ovat vaikuttaneet myös kansainvälisen ohjelmistokehityksen muotoutumiseen (Denhere ja muut, 2015). Globalisaation myötä esteet kansainvälisen ohjelmistokehityksen toteuttamisen edessä ovat pienentyneet ja viimeistään 1990-luvulta lähtien kansainvälinen ohjelmistokehitys on kasvanut voimallisesti (Prikladnicki ja muut, 2006).

Koska iso osa ohjelmistokehityksestä on kommunikointia kehitykseen osallistuvien muiden henkilöiden kanssa ja koska henkilökohtainen vuorovaikutus on kansainvälisessä ohjelmistokehityksessä maantieteellisten erojen vuoksi hankalampaa, on yksi tärkeä kansainvälisen ohjelmistokehityksen mahdollistanut tekijä ollut kommunikaatioteknologian kehittyminen (Herbsleb & Mockus, 2003). Kommunikaatioteknologian kehitys on tuonut kansainvälisen ohjelmistokehityksen käyttöön monia työkaluja, kuten esimerkiksi sähköpostin, videoneuvottelut, erilaiset chat-työkalut ja mahdollisuuden jakaa henkilökohtaisen tietokoneen työpöytä neuvottelukumppanin kanssa. Lisäksi on kehitetty erilaisia projektinhallinnan työvälineitä, jotka helpottavat projektin hallintaa kansainvälisessä kontekstissa. (Jain & Suman, 2018)

## 2.2 Nykytilanne

Jainin ja Sumanin (2015) mukaan ohjelmistokehitys jakautuu yhä enemmän maantieteellisesti ja kansainvälisestä sovelluskehityksestä on tullut tietyillä sovellusalueilla vallitseva normi. Ohjelmistokehityksen globalisaatio tapahtuu joko niin, että kehitys tai osa siitä annetaan organisaation ulkopuolisen yhteistyökumppanin hoidettavaksi (offshore

outsourcing) tai sitten organisaatio perustaa ulkomaille oman erillisen osaamiskeskukseen (offshore insourcing). Silloin kun sovelluskehitykseen osallistuvat kehitystiimit jakautuvat työskentelemään maantieteellisesti useamman valtion alueelle, voidaan puhua globaalista ohjelmistokehityksestä (GSD).

Keskitettyssä ympäristössä tapahtuvan sovelluskehityksen avuksi löytyy standardoituja prosessimalleja. Hajautettua kehitystä tukevista malleista sen sijaan on puutetta. Yrityksissä on hajautetussa sovelluskehityksessä sovellettu pidemmän aikaa perinteisiä prosesseja, kuten vesiputousmallia. Koska hajautettujen projektien epäonnistumisaste on ollut erittäin korkea, yritykset ovat alkaneet enenevästi siirtyä käyttämään ketteriä menetelmiä, joilla toivotaan päästävän parempaan onnistumisprosenttiin. Ketterien menetelmien edut näkyvät erityisesti muutoksenhallinnassa. (Cusumano, 2008)

### **2.3 Hyödyt**

GSD on nykyaikainen liiketoimintamalli, joka mahdollistaa yritykselle kansainvälisen kilpailun merkittävien kustannussäästöin. Ohjelmistoyhtiöt siirtävät kehitystyötään kansainvälisemmäksi strategisten ja taloudellisten etujen toivossa (Jiménez ja muut, 2009). Jain ja Suman (2018) esittävät, kehitystoiminnan siirtämiselle kansallisten ja organisatoristen rajojen yli, tärkeimmäksi hyödyksi mahdollisuuden vähentää kustannuksia käyttämällä vähemmän kehittyneiden talouksien tarjoamaa koulutettua työvoimaa. Muina hyötyinä kustannussäästöjen ohella Conchúir ja muut (2009a) mainitsevat mahdollisuuden palata osaavia työntekijöitä ja päästä lähemmäs markkinoita ja kuluttajia sekä näin hyödyntää markkinoita lähellä toimivan toimittajan paikallistuntemusta. Tietyissä tapauksissa yritykset saattavat myös hyötyä kohdevaltion suotuisasta politiikasta ja verohelpouksista (Gopal ja muut, 2002).

Ulkomailla toimivalta yhteistyökumppanilta palveluita hankkiva yritys saattaa hyötyä järjestelystä lisäksi sitä kautta, että yhteistyökumppani on kokenut toimija vastaavanlaisissa kehitysprojekteissa. Tällöin sen työn laatu ja palvelun taso on päässyt kehittymään

korkeaksi ja lisäksi sillä saattaa olla hallussaan esimerkiksi kyseisessä tehtävässä tarvittavaa spesifistä teknologiaa (Khan ja muut, 2010). Asiakasyrityksen motivaationa yhteistyölle saattaa myös olla keskittyminen puhtaasti omaan ydinliiketoimintaan, jolloin prosessien ja järjestelmien toteutus ulkoistetaan yhteistyökumppanille. Yhteistyökumppanin kanssa toteutettava toimintojen ulkoistaminen saattaa tuottaa hyötynä vielä sen, että asiakasorganisaatio saa jaettua ohjelmistokehitykseen liittyvän riskin toisen organisaation kanssa (Niazi ja muut, 2016b).

Edellä mainittuja GSD:n hyötyjä Ågerfalk ja muut (2008) nimittävät ”tunnetuiksi” hyödyiksi. ”Tunnettujen” hyötyjen lisäksi he ovat tunnistaneet vielä joukon ”tuntemattomia” hyötyjä. ”Tuntemattomat” hyödyt he ryhmittelevät organisaatioon liittyviin hyötyihin, tiimiin liittyviin hyötyihin sekä prosesseihin ja tehtäviin liittyviin hyötyihin. Heidän mukaansa ”tuntemattomista” hyödyistä kaksi liittyy pääasiassa organisaatioon. Organisaatiotasolle kytkeytyvät hyödyt ovat innovaatiot ja jaetut parhaat käytännöt sekä parantunut resurssien allokaatio. Innovaatioita ja hyviä käytäntöjä syntyy ja jaetaan, kun erilaisista taustoista tulevat henkilöt tekevät töitä keskenään. Resurssien parantunut allokaatio voi syntyä muun muassa siitä, että halvan työvoiman maista saadaan haettua kustannussäästöjä, jolloin kustannustasoltaan arvokkaammat henkilöresurssit voidaan sijoittaa organisaation kannalta strategisesti kannattavampiin tehtäviin.

Tiimin tasoon liittyviä ”tuntemattomia” hyötyjä Ågerfalk ja muut (2008) tunnistavat kolme kappaletta. Ne ovat parantunut tehtävien modularisaatio, vähentyneet koordinaatiokustannukset ja tiimien lisääntynyt itsenäinen toiminta. Prosesseihin ja tehtäviin liittyviä ”tuntemattomia” hyötyjä he esittävät kolme kappaletta. Niitä ovat määrämukainen viestintä, parantunut dokumentaatio ja selkeästi määritellyt prosessit. Ågerfalk ja muut (2008) esittämät hyödyt GSD:lle on esitetty taulukossa 2.

**Taulukko 2.** GSD:n hyödyt (Ågerfalk ja muut, 2008).

Organisaation hyödyt	Tiimin hyödyt	Prosessin/ tehtävän hyödyt
<p>”Tunnetut” hyödyt</p> <ul style="list-style-type: none"> <li>• Kustannussäästöt</li> <li>• Osaavan työvoiman saatavuus</li> <li>• Lyhyempi aika markkinoille pääsyyn</li> <li>• Markkinoiden ja asiakkaan läheisyys</li> </ul> <p>”Tuntemattomat” hyödyt</p> <ul style="list-style-type: none"> <li>• Innovaatiot ja jaetut parhaat käytännöt</li> <li>• Parantunut resurssien allokointi</li> </ul>	<p>”Tuntemattomat” hyödyt</p> <ul style="list-style-type: none"> <li>• Parantunut tehtävien modularisointi</li> <li>• Vähentyneet koordinaatiokustannukset</li> <li>• Lisääntynyt tiimien itsenäinen toiminta</li> </ul>	<p>”Tuntemattomat” hyödyt</p> <ul style="list-style-type: none"> <li>• Määrämuotoinen viestintä, josta jää auditoitava jälki</li> <li>• Parantunut dokumentaatio</li> <li>• Selkeästi määritellyt prosessit</li> </ul>

Ebertin ja De Neven (2001) mukaan GSD ohjaa parhaassa tapauksessa tiimit jakamaan tehtävänsä järkeviin osakokonaisuuksiin. Osiin jakaminen mahdollistaa sen, että osakokonaisuudesta vastaava taho pystyy ottamaan omasta osuudestaan kokonaisvastuun.

Ebertin (2011, s.18) mukaan innovaatiot lisääntyvät ja parhaat käytännöt leviävät ympäristöissä, joissa erilaisista koulutuksellisista, organisatorisista ja kulttuurillisista lähtökohdista tulevat tiimin jäsenet työskentelevät yhdessä. Näin ollen erilaisista taustoista tulevat projektiryhmän jäsenet lisäävät parhaassa tapauksessa ryhmän tuottavuutta tuomalla ryhmän toimintaan erilaisia tapoja työskennellä sekä ratkaista ongelmia.

Espinosa ja Carmel (2004) ovat esittäneet, että kansainvälisten tiimien on mahdollista tietyin rajoituksin nopeuttaa kehitystyötä käyttämällä ”follow-the-sun”-periaatetta. ”Follow-the-sun”-periaatteessa kansainväliset tiimit on sijoitettu maantieteellisesti eri

aikavyöhykkeille niin, että tiimien työajat eroavat toisistaan. Tällöin tiimit eivät työskentele samaan aikaan työstäen samoja tehtäviä. ”Follow-the-sun”-periaate mahdollistaa tarvittaessa ympärivuorokautisen työskentelyn, jolloin työt voidaan siirtää tiimiltä toiselle edellisen tiimin lopettaessa työskentelyn seuraavan tiimin aloittaessaan vuoroaan. Jatkuvasti käytettynä, etenkin ohjelmistokehitystyössä, ”follow-the-sun” ei heidän mukaansa kuitenkaan ole toimiva ratkaisu, koska siihen liittyvät koordinaatio-ongelmat kasvavat liian suuriksi.

Hajaantuneisuus lisää tiimin itsenäisyyttä. Itsenäisyyden lisääntyminen johtuu siitä, että hajautumisen kasvamisen myötä kommunikaatio hidastuu. Lisäksi koska hajautetussa ympäristössä joudutaan kommunikoimaan enemmän kirjallisesti ja käyttämällä erilaisia kommunikaatioteknologioita, tallentuu suuri osa viesteistä sähköisessä muodossa. Se luo viestintään määrämuotoisuutta ja dokumentoi toimintaa. (Gumm, 2006; Conchúir ja muut, 2009a)

## 2.4 Haitat

Hyötyjen lisäksi GSD-projekteihin liittyy myös lukuisia ongelmia. Maantieteellisesti hajautettujen projektien hallinta on monimutkaisempaa kuin paikallisesti toteutettujen projektien. Hajautetun projektin hallinnassa kohdataan normaalin projektinhallinnan vaikeudet ja sen lisäksi vielä hajautuksesta syntyvät omat tarpeensa (Casey, 2010). Jainin ja Sumanin (2018) mukaan GSD-projektien epäonnistumisriskiä lisäävät maantieteelliseen, ajalliseen, kulttuuriseen ja organisatoriseen hajauttamiseen liittyvät riskit. Näitä riskejä ovat muun muassa korkeammat koordinaatiokustannukset, vähentynyt henkilökohtainen kommunikaatio ja kulttuuriin liittyvät väärinkäsitykset. Herbsleb ja Mockus (2003) painottavat riskeinä vielä erityisesti eri puhuttuihin kieliin liittyviä ongelmia. Näiden potentiaalisten uhkien takia projektin yhteydessä on tehtävä riittävä riskiarvio ja projektin aikana on panostettava riskienhallintaan. Jos GSD-projektia on toteuttamassa osallistujia eri organisaatioista, he saattavat noudattaa erilaisia prosesseja, joka johtaa helposti ongelmiin integraatiossa. Tästä aiheutuu helposti turhautumista,



epäluottamusta ja väärinkäsityksiä, jotka puolestaan johtavat viivästymisiin, virheisiin ja heikentyneeseen laatuun. Useammassa eri maassa toimittaessa, pitää myös ottaa huomioon se, että eri maiden lainsäädännöissä ja vallitsevissa käytännöissä voi olla eroa (Saleem, 2019). Khan ja muut (2011) esittävät GSD-projektien riskiä lisäävän lisäksi sen, että monet organisaatiot aloittavat globaalien ohjelmistokehityksen niin, ettei heillä ole etukäteen käsitystä oman projektijohtamisensa kyvystä selviytyä kansainvälisen toiminnan vaatimuksista. Epäonnistumiset projektijohtamisessa lisäävät riskiä projektin kustannusten kasvusta ja aikataulun pettämisestä.

Herbsleb ja Moitra (2001) jaottelevat artikkelissaan GSD:hen liittyvät vaikeudet yllä esitetystä vielä hieman eri tavalla. He jaottelevat ne strategisiin, kulttuurillisiin, kommunikaatioon, tiedon hallintaan, projektin ja prosessien hallintaan sekä teknisiin kysymyksiin. Strategiset hankaluudet liittyvät työtehtävien ja työkokonaisuuksien jakamiseen sekä joissain tilanteissa GSD:tä kohtaan tunnettuun vastustukseen. Mahdolliset kulttuurilliset ongelmat muodostuvat erilaisten kulttuurien keskinäisestä hankauksesta ja kulttuurieroista johtuvista näkemyseroista. Kommunikaation ja tiedon hallinnan vaikeudet liittyvät GSD:ssä suurelta osin siihen, että epävirallinen kommunikaatio puuttuu, eikä tieto ja tietämys siirry eri fyysisten sijaintipaikkojen välillä. Projektin ja prosessien hallinnan hankaluudet koostuvat eri sijaintipaikoissa työskentelevien tiimien työn synkronoinnista. Tekniset ongelmat liittyvät käytettyihin teknologioihin, kuten tiedonsiirtoteknologiaan ja muihin työvälineisiin.

Conchúir ja muut (2009a) mukaan GSD:n potentiaalisesti tarjoamat edut on kuitenkin mahdollista saavuttaa vain osittain hajautetusta työskentelystä johtuvien ongelmien takia. Heidän mukaansa ne voidaan jaotella maantieteellisiin, ajallisiin ja sosiaalis-kulttuurillisiin ongelmiin. Ne vaikuttavat keskeisimmin viestintään, joka vaikeutuu ja kallistuu hajautuksen myötä, mutta myös koordinointiin ja yhteistyöhön. Tutkimuksessaan Conchúir ja muut (2009b) seurasivat kolmea kansainvälistä ohjelmistokehitystä tekevää organisaatiota. Heidän havaintonsa GSD:n oletettujen hyötyjen toteutumisesta on kuvattu taulukossa 3.

**Taulukko 3.** GSD:n hyötyjen toteutuminen (Conchúir ja muut, 2009b).

Oletettu hyöty	Toteuman määrä/ havainnot	Arvio toteumasta
Alentuneet kustannukset	<ul style="list-style-type: none"> <li>Selvä palkkaero länsimaiden ja kehittyvien maiden välillä</li> <li>Hajautettuna olivat vain vähemmän monimutkaiset ja alhaisen lisäarvon työt.</li> <li>Merkittäviä lisäkustannuksia viestinnässä ja koordinoinnissa yms.</li> </ul>	Osittain toteutunut
Aikavyöhykkeiden hyödyntäminen	<ul style="list-style-type: none"> <li>Eri aikavyöhykkeillä työskentely ei tuonut hyötyjä, mutta se vähensi yhteistä työskentelyaikaa sekä pakotti joustamaan työajoissa.</li> <li>”follow-the-sun”-periaatetta käytettiin vain vähän, esimerkiksi testauksessa.</li> </ul>	Ei hyötyä
Toimipisteiden välinen työn modulaarisuus	<ul style="list-style-type: none"> <li>Modulaarisuus voi vähentää tarvittavan viestinnän määrää toimipisteiden välillä.</li> <li>Voi haitata tiimiytymistä toimipisteiden välillä.</li> </ul>	Osittain toteutunut
Ammattitaitoisten työntekijöiden saavuus	<ul style="list-style-type: none"> <li>GSD mahdollistaa ammattitaitoisten työntekijöiden rekrytoinnin.</li> <li>Työntekijöiden vaihtuvuus voi olla korkea.</li> <li>Kaikille tarvittaville osaamisalueille ei välttämättä löydetä työntekijöitä helposti.</li> </ul>	Osittain toteutunut

Oletettu hyöty	Toteuman määrä/ havainnot	Arvio toteumasta
	<ul style="list-style-type: none"> <li>Sosio-kulttuurillisia ongelmia on runsaasti.</li> </ul>	
Innovaatio ja jaetut parhaat käytännöt	<ul style="list-style-type: none"> <li>Tiedonjakaminen heikkenee, jos matlapalkkaiset kollegat koetaan uhaksi.</li> </ul>	Ei hyötyä
Lyhyt etäisyys markkinoihin ja asiakkaaseen	<ul style="list-style-type: none"> <li>Vaikka lyhyt etäisyys tarjoaa paremman kontaktin asiakkaaseen, kulttuuriin liittyvät ongelmat lisääntyvät.</li> </ul>	Osittain toteutunut

Herbslebin ja Mockusin (2003) mukaan hajautettujen tiimien jäsenillä on selvästi vähemmän henkilökohtaista ja epävirallista kanssakäymistä keskenään. He kuuluvat erilaisiin kulttuureihin, puhuvat mahdollisesti eri kieliä ja työskentelevät eri aikavyöhykkeillä, joten heidän yhteistyönsä ja tuottavuutensa ei pääse kasvamaan samalla tavalla kuin keskitetyimmässä ympäristössä (Anantatmula & Thomas, 2010).

Vaikka kaikki hajautetun tiimin jäsenet puhuisivatkin yhteistä kieltä, se ei Ebertin ja De Neven (2001) mukaan vielä riitä takaamaan täydellistä yhteisymmärrystä. Yhteisestä puhutusta kielestä huolimatta kaikki tiimin jäsenet eivät välttämättä ymmärrä kaikkien käsitteiden merkitystä täysin samalla tavalla. Esimerkiksi englantia äidinkielenään puhuvalle tietyt sanat tai lauserakenteet saattavat merkitä asioita, jotka eivät toisenlaisesta kielitaustasta tulevalle henkilölle helposti aukea. Kielen täsmällisen merkityksen ymmärtäminen korostuu erityisesti, kun ollaan tekemässä kirjallisia sopimuksia.

GSD:n kokonaiskustannuksia ja sillä saavutettavia hyötyjä voi olla vaikea laskea, koska osa GSD:n kustannuksista tulee piilokustannuksina. GSD:n kustannussäästöjen arvioinnissa tulisi ottaa huomioon muun muassa eri aikavyöhykkeillä työskentelystä aiheutuvat kustannukset, tarvittavien työvälineiden käyttökustannukset sekä työntekijöiden pätevyys ja heidän kokemuksensa. Jotta GSD olisi kannattavaa, pitäisi hajautettuun työhön liittyvistä ylimääräisistä kustannuksista, kuten matkustamisesta, työvälineistä tai

koulutuksista, aiheutuvien kulujen jäädä pienemmiksi kuin GSD:n kustannushyödyt. (Jain & Suman, 2018)

## 2.5 GSD-prosessi

Globaali ohjelmistokehitys (GSD) sekä globaalisti hajautetut tiimit yleistyvät jatkuvasti. Perinteinen syy GSD:hen on ollut kustannussäästöjen hakeminen siirtämällä työtä halvemman kustannustason maihin. Halvempien henkilöstöresurssien ohella GSD:n avulla voidaan päästä palkkaamaan myös uusia työntekijöitä uusilta alueilta. Tämä auttaa lisäksi uusille markkina-alueille siirtymisessä. Perinteisille johtamismalleille GSD tuo vaahteita. GSD lisää erilaisten toimijoiden ja yhteistyötahojen määrää, jolloin johtamismalleja voidaan joutua sopeuttamaan vastaamaan tilannetta. (Noll ja muut, 2016)

Suurimmassa osassa tapauksista kansainvälistä sovelluskehitystä tehdään asiakas- ja palveluntarjoajayritysten kesken. Tällöin asiakas hyötyy kustannussäästöistä ja palveluntarjoaja saattaa hyötyä valtion myöntämistä tuista, esimerkiksi verohelpotuksista. Asiakas pystyy myös jakamaan riskejään, ulkoistamaan teknistä työtä ja keskittymään ydinliiketoimintaansa. (Niazi ja muut, 2016b; Jain & Suman, 2015)

### 2.5.1 Ketterät menetelmät

Vaikka ketterien menetelmien juuret juontuvat 1990-luvulle, ketterien menetelmien alkuna pidetään yleensä vuotta 2001, jolloin ajatus ketterästä ohjelmistokehityksestä virallisesti esiteltiin suurelle ohjelmistokehittäjien yhteisölle (Agile Alliance, 2020). Ketterien menetelmien pohjalla on tarve vastata muutokseen kesken ohjelmistokehitysprojektin. Ketterä kehitys on ylätasoinen käsite ja sen alle kuuluu lukemattomia viitekehyksiä ja käytäntöjä, jotka perustuvat ketterän ohjelmistokehityksen julistukselle: Agile Manifestolle. Ketterän kehityksen viitekehyksiä ovat muun muassa Scrum, Extreme Programming (XP), Feature-driven development (FDD) ja Scaled Agile Framework (SAFe).

Näiden kahden suuren ohjelmistokehitystrendin, ketterien menetelmien ja kansainvälisesti hajautetun työskentelyn, yhdistäminen muodostaa kuitenkin yhden suurimmista hajautettuun ohjelmistokehitykseen liittyvistä ongelmista (Cusumano, 2008). HankaluuDET tulevat selvemmin esiin, kun ohjelmistokehitystiimit kasvavat suuriksi. Ketterien menetelmien käyttöönotosta haetaan tällä hetkellä ohjelmistokehityksessä hyötyjä ja se korostaa niiden ja GSD:n välisiä ongelmakohtia. Layman ja muut (2006) tuovat esiin, että ketterässä ohjelmistokehityksessä epämuodollinen ja spontaani kommunikaatio on keskeisessä osassa. Hajautetussa työskentelyssä tällaisen kommunikaation mahdollisuus kuitenkin vähenee. Lisäksi myös mahdolliset kulttuurilliset erot heikentävät kommunikaation tehokkuutta. Ebert ja Paasivaara (2017) näkevät hajautetun ohjelmistokehityksen ja ketterien menetelmien välisenä ongelmana skaalautuvuuden. Ketterien menetelmien mallit, kuten Scrum, on kehitetty paikallisempaan työskentelyyn ja monesti myös kohtalaisen pienten kehitystiimien käyttöön. Niiden käyttöönotto suurissa kansainvälisesti hajautetuissa kehitystiimeissä on hankalaa.

Käytettäessä ketteriä ohjelmistokehitysmenetelmiä hajautetusti toteutetussa ohjelmistokehitysprojektissa, joudutaan ketteriä menetelmiä mukauttamaan kyseiseen ympäristöön. Robinson (2019) väittää, ettei ainakaan Scrum sellaisenaan sovellu käytettäväksi globaalissa ohjelmistokehityksessä ilman modifiointia. Modifioiduilla ketterillä menetelmillä on kuitenkin saavutettu onnistuneita tuloksia kansainvälisissä ohjelmistoprojekteissa. Ketterän ja hajautetun ohjelmistokehityksen yhdistelmästä käytetään nimitystä Distributed Agile Software Development. Distributed Agile Software Development pyrkii vastaamaan vaatimukseen, jota näiden kahden ohjelmistokehitystrendin yhteensovittamisessa on havaittu, kuten esimerkiksi kommunikaatioon ja koordinaatioon liittyviin ongelmiin. Yhdistelmä pyrkii tarjoamaan etuja molemmista menetelmistä, vaikka ne perusteiltaan ovatkin monilta osin keskenään hyvin ristiriitaisia (Kaur & Sharma, 2014). Ketterissä menetelmissä korostetaan sekä tiimin jäsenten että tiimin jäsenten ja asiakkaan välistä tiivistä yhteistyötä sekä epämuodollista kommunikaatiota. GSD:ssä sen sijaan ohjelmistokehitykseen osallistuvat henkilöt on hajautettu maantieteellisesti, ja he

kommunikoivat paljon muodollisemmin käyttäen erilaisia kommunikaatiovälineitä. Viestintäohjelmistojen kehittyminen ja etenkin tiedonsiirtonopeuksien riittävä kasvu ovat kuitenkin mahdollistaneet kasvokkain tapahtuvan viestinnän korvaamisen kommunikaatiovälineiden käytöllä ketterissä kehitysprojekteissa (Robinson, 2019). Suuret aikaerot saattavat silti vaikeuttaa yhteisten säännöllisten palaverien pitämistä, vaikka viestintätekniikka sen muuten mahdollistaisikin. Vaikka Scrum ei sellaisenaan täysin soviakaan käytettäväksi GSD:ssä, sen osioita ja käytäntöjä on käytetty paljon hyödyksi myös GSD:ssä. Vallon ja muut (2018) ovat työssään (vuosina 1999–2016) huomanneet, että monissa GSD-projekteissa on omaksuttu Scrumista käyttöön esimerkiksi seisomakokoukset (standup meeting), backlogit, sprintit ja sprintteihin liittyvät retrospektiivit sekä demot. Myös normaaleista ketteristä menetelmistä tutut jatkuva integraatio, käyttötapaukset ja pariohjelmointi ovat yleisesti käytössä.

Alzoubi ja muut (2016) suosittelevat suunnittelemaan viestinnän huolellisesti, jotta ketterien menetelmien ja kansainvälisesti hajautetun ohjelmistokehityksen yhdistämisestä aiheutuvat vaikeudet saataisiin minimoitua. Keskeistä tässä on työskentelyä tukevan kommunikaatioteknologian käyttö ja dokumentaation riittävä hyvä ylläpito. Jotta hajautetun ryhmän reaaliaikainen kommunikaatio olisi mahdollista, tulisi työajat saada sovittua niin, että ryhmän jäsenillä olisi yhteinen aikaikkuna, jolloin on mahdollista pitää reaaliaikaisia palavereja. Sikäli kuin mahdollista, myös fyysisiä tapaamisia tulisi järjestää. Näitä voisivat olla esimerkiksi ryhmäkokoukset projektien alussa ja vierailut projektin aikana eri toimipisteiden tai ryhmien välillä.

### **2.5.2 Perinteinen ohjelmistokehitys**

Ketterien menetelmien ja hajautetun ohjelmistokehityksen yhteensovittamisesta aiheutuvien vaatimusten johdosta monet yritykset ovat tehneet tietoisien ratkaisun käyttää GSD-projektiensa projektinhallinnassa perinteisempiä, vesiputousmallin kaltaisia, menetelmiä (Cusumano, 2008). Näin toimimalla menetetään iteratiivisten ja ketterien tekniikoiden tuoma etu siitä, että pystytään vastaamaan nopeasti vaatimusten muutoksiin

projektin ollessa jo käynnissä. Perinteisten projektinhallintamenetelmien käyttäminen GSD-projektissa vähentää kuitenkin kommunikointiin ja koordinointiin liittyviä ongelmia.

Cusumano (2008) kehottaa käyttämään jokaisessa projektissa kyseiseen projektiin ja tilanteeseen parhaiten sopivaa mallia ja tekniikkaa. Hänen hyväksi havaitsemansa käytäntö on se, että projekti aloitetaan iteratiivisena prosessina, toteutusvaiheessa siirrytään perinteiseen menetelmään, esimerkiksi vesiputousmalliin ja projektin lopussa vaihdetaan taas takaisin iteroivaan malliin. Käytännössä tämä voisi tarkoittaa esimerkiksi sitä, että ohjelmiston toimittajan edustaja työskentelee asiakkaan lähellä. Asiakkaan lähellä hoidetaan projektinhallinta ja vaatimusmäärittely. Yksityiskohtaisempi suunnittelu, toteutus, testaus ja integraatio hoidetaan hajautettuna työskentelynä. Tämän jälkeen toteutetut osat tuodaan asiakkaalle hyväksymistestausta varten, jonka jälkeen tehdään uusi iteraatio.

### 3 Projektinhallinta globaalissa kontekstissa

Project Management Institutun (2013, s. 3–5) mukaan projekti on tilapäinen panostus tuotteen, palvelun tai muun vastaavan tekemiseksi. Projektinhallinnassa projektiin sovelletaan sopivia tietoja, taitoja, työkaluja sekä tekniikoita, jotta projektin vaatimukset saadaan täytettyä. Projektinhallintaan kuuluu viisi prosessiryhmää, jotka ovat: aloitus, suunnittelu, toteutus, seuranta ja valvonta sekä projektin päättäminen.

#### 3.1 Globaali projektinhallinta

Bannerman ja muut (2012) mukaan globaalilla ohjelmistokehityksellä (GSD) tarkoitetaan ohjelmistokehitystä, jota toteutetaan useamman valtion tai mantereiden alueella. GSD lisää tuntuvasti projektien potentiaalia ja kuten Ågerfalk ja muut (2008) esittävät, se takaa muun muassa pääsyn käyttämään laajempaa osaavaa henkilöresurssipohjaa, tuo kustannusetuja ja mahdollistaa vuorokauden ympäri työskentelyn. Silti GSD lisää myös huomattavasti vaatimuksia projektinhallintaan. Nämä vaatimukset syntyvät GSD-projektien monimuotoisuudesta ja monimutkaistumisesta (Holmström ja muut, 2006). Maantieteellisen jakautumisen lisäksi hajautettu projektiryhmä kohtaa mahdollisesti erilaisista aikavyöhykkeistä ja kulttuurieroista johtuvia vaikeuksia. Kulttuurieroihin kuuluvat muun muassa erilaiset kielet, kansalliset perinteet, arvot ja käyttäytymisnormit (Jaanu ja muut, 2012).

Saleem (2019) väittää, että syynä useiden GSD-projektien epäonnistumiseen on yrityksen tai organisaation lähteminen mukaan kansainväliseen ohjelmistokehitykseen ilman riittävää valmistautumista etukäteen. Kompastuskiveksi muodostuu monesti projektinhallinta, jota ei osata sopeuttaa oikein vastaamaan GSD:n vaatimuksia. Ebert ja De Neve (2001) havaitsivat, että suurissa kansainvälisesti hajautuneissa kehitysorganisaatioissa saattaa syntyä ongelmia kehitystiimien välisessä töiden koordinoinnissa. Äärimmäinen esimerkki on tilanne, jossa eri tiimit työstävät yhteisestä globaalista, niin sanotusta perustuotteesta, paikallisia omia versioitaan. Tällöin vaarana on, että kehitysversiot



saattavat eriytyä toisistaan niin, ettei niiden keskitetty, globaali kehittäminen ja eri versioiden synkronointi ole enää jatkossa mahdollista.

Huono kommunikaatio ja koordinointi ovat usein syynä siihen, että hajautetussa ohjelmistonkehityksessä syntyy ongelmia. Kun ihmiset työskentelevät keskitetysti samassa paikassa, he jakavat luontaisesti tietoa keskenään epävirallisessa kanssakäymisessä. Tieto esimerkiksi asiantuntemuksesta, tehtävien prioriteetista ja tilanteesta siirtyy virallisten ja puolivirallisten kokousten lisäksi myös erilaisissa kahvi- ja käytäväkeskusteluissa. Tällaista tiedonsiirtoa ei pääse tapahtumaan yhtä helposti hajautetussa työskentely-ympäristössä, ja tutkimukset osoittavatkin vahvasti kommunikaation vähenevän, kun työtoverit ovat fyysisesti erossa toisistaan. Nämä hajautetussa ympäristössä menetetyt kommunikaation yksityiskohdat ovat kriittisen tärkeitä tehokkaan viestinnän ja koordinoinnin muodostamisessa. Näiden puuttuminen saattaa herkästi vaikeuttaa kommunikaatiota ja hidastaa työskentelyä. Koska hajautetussa ympäristössä suora ja reaaliaikainen kommunikaatio vähenee, korostuu epäsuoran viestinnän rooli. (Herbsleb & Mockus, 2003; El Bajta ja muut, 2018)

GSD lisää projektien monimutkaisuutta verrattuna paikallisesti toteutettuihin projekteihin. Sen tähden projektinhallinta on erityisen tärkeässä roolissa GSD-projekteissa koko projektin keston ajan. Täsmällisellä projektinhallinnalla varmistetaan projektin tavoitteiden asianmukainen saavuttaminen. El Bajta ja muut (2018) mukaan GSD-projektien projektinhallinnassa tulisi erityisesti panostaa tarkkaan etukäteissuunnitteluun sekä projektin aikaiseen seurantaan ja valvontaan. Ohjelmistokehitysprojekteissa käytetään enenevässä määrin ketteriä ohjelmistokehitysmenetelmiä. Sen osalta GSD-projektit eivät muodosta poikkeusta ja ketterät menetelmät ovatkin usein käytössä myös GSD:ssä. Paikallisesti toteutettuna ketterät menetelmät tukeutuvat pitkälti nopeaan ja säännölliseen kommunikaatioon sekä välittömään palautteen antoon. Samalla ketterissä menetelmissä määrämuotoisen dokumentaation laatiminen priorisoidaan alemmas kuin perinteisissä ohjelmistonkehitysmenetelmissä. GSD:ssä työskentelyn hajauttaminen kuitenkin vähentää projektiryhmän sisäistä kommunikaatiota ja heikentää epävirallista viestintää.

Se myös hidastaa palautteen antamista. Ketteriä menetelmiä joudutaan näiltä osin usein muokkaamaan GSD:hen sopiviksi. (El Bajta ja muut, 2018)

Ebert ja muut (2016) kritisoivat GSD:n mukanaan tuomaa ”follow-the-sun” –periaatetta, jossa kaikki tiimit eivät työskentele samaan aikaan, mutta joku tiimi on mahdollisesti koko ajan työssä. Tällainen työskentelymalli saattaa tehdä projektinhallinnasta vaikeampaa ja monimutkaisempaa. Ongelmia voi seurata myös esimerkiksi työvälineiden ylläpidossa ja varmuuskopioiden ottamisessa, jos niihin liittyvien huoltotöiden tekemistä varten ei jää tarpeellisia huoltoikkunoita.

Menestyäkseen kansainvälisessä ohjelmistokehityksessä yrityksen tulee hallita yllä mainittuja GSD-projekteihin liittyviä riskejä. Parhaimmillaan globaalisti toteutettu projekti yhdistelee ja sekoittaa erilaisia tapoja sekä ideoita ja luo näin tehokkaasti uusia innovaatioita (Ebert & De Neve, 2001). Saleem ja muut (2019) mukaan GSD:ssä kohdataan kuitenkin usein hankaluuksia. GSD:n vaikeudet liittyvät suurelta osin kehitysprojektin sidosryhmien sisäiseen viestintään ja yhteistyöhön.

### **3.1.1 Projektiryhmän jäsenet**

Ebert ja De Neve (2001) ovat todenneet tekemässään tapaustutkimuksessa, että samaan maantieteelliseen paikkaan keskitetyt projektitiimit saavuttivat yli 50 prosenttia suuremman tehokkuuden ohjelmiston suunnittelussa ja toteutuksessa sekä tehtyjen virheiden havaitsemisessa, kun niitä verrattiin hajautettuihin tiimeihin. Työskentelyn tehokkuutta heikensivät myös osa-aikainen työskentely ja työn kierto. Projektityöskentelyn tehokkuuden optimoimiseksi projektin jäsenet tulisi sijoittaa työskentelemään lähelle toisiaan, mielellään samaan tilaan. Heidän työaikansa tulisi allokoita projektille täysipäiväisesti. Näillä toimenpiteillä voidaan heidän mukaansa päästä merkittävään työskentelyn tehokkuuden lisäykseen. Hajautetussa projektissa tällaiset työjärjestelyt eivät yleensä ole kuitenkaan mahdollisia.

Tutkimuksensa pohjalta Ebert ja De Neve (2001) ehdottavat projektin henkilöiden allokaatioon seuraavia parannuksia:

- Kokopäiväinen allokaatio. Sijoitetaan mahdollisimman suuri osa projektin työhön osallistuvista henkilöistä kokopäiväisiksi tai melkein kokopäiväisiksi projektin työntekijöiksi.
- Luotettava allokaatio. Varmistetaan, että projektin käyttöön varatut henkilöt ovat varmasti käytettävissä. Tässä auttaa, jos henkilöiden käytettävyydestä sovitaan selkeillä alkamis- ja päättymispäivillä.
- Paikallinen työskentely. Pyritään varmistamaan, että projektiryhmä mahdollisuuksien mukaan toimii keskitetysti yhdessä paikassa. Hajautetusti työskentelevä tiimi työskentelee tehottomammin kuin keskitetysti samassa paikassa työskentelevä tiimi.
- Kehityksen erottaminen ylläpidosta. Erotetaan uusien toiminnallisuuksien kehittäminen ohjelmiston ylläpitotöistä ja virheiden korjauksesta. Kehitys ja ylläpito tulisi organisoida erikseen. Henkilöiden allokoiminen molempiin tehtäviin sotii kokopäiväistä allokaatiota vastaan, vaikka henkilö työskentelisikin saman ohjelmiston parissa täyspäiväisesti.

Ebert ja De Neve (2001) esittävät lisäksi, että työtehtävien suunnittelussa pitäisi pyrkiä tiimin jäsenten asiantuntemuksen laajentamiseen. Liian kapea asiantuntemus tai osaa-misalue projektissa, saattaa heikentää tiimiläisen kykyä tai halua ottaa kokonaisvastuuta projektin tuloksista. Tiimiläisten asiantuntemuksien laajentaminen tuo tehokkuutta tiimin työskentelyyn sekä parantaa motivaatiota ja kokonaisnäkemystä projektista. Jos projektin työskentelykulttuuri on aikaisemmin tukenut kapea-alaista asiantuntemusta, saattaa muutos tuntua aluksi vaikealta, mutta se kannattaa.

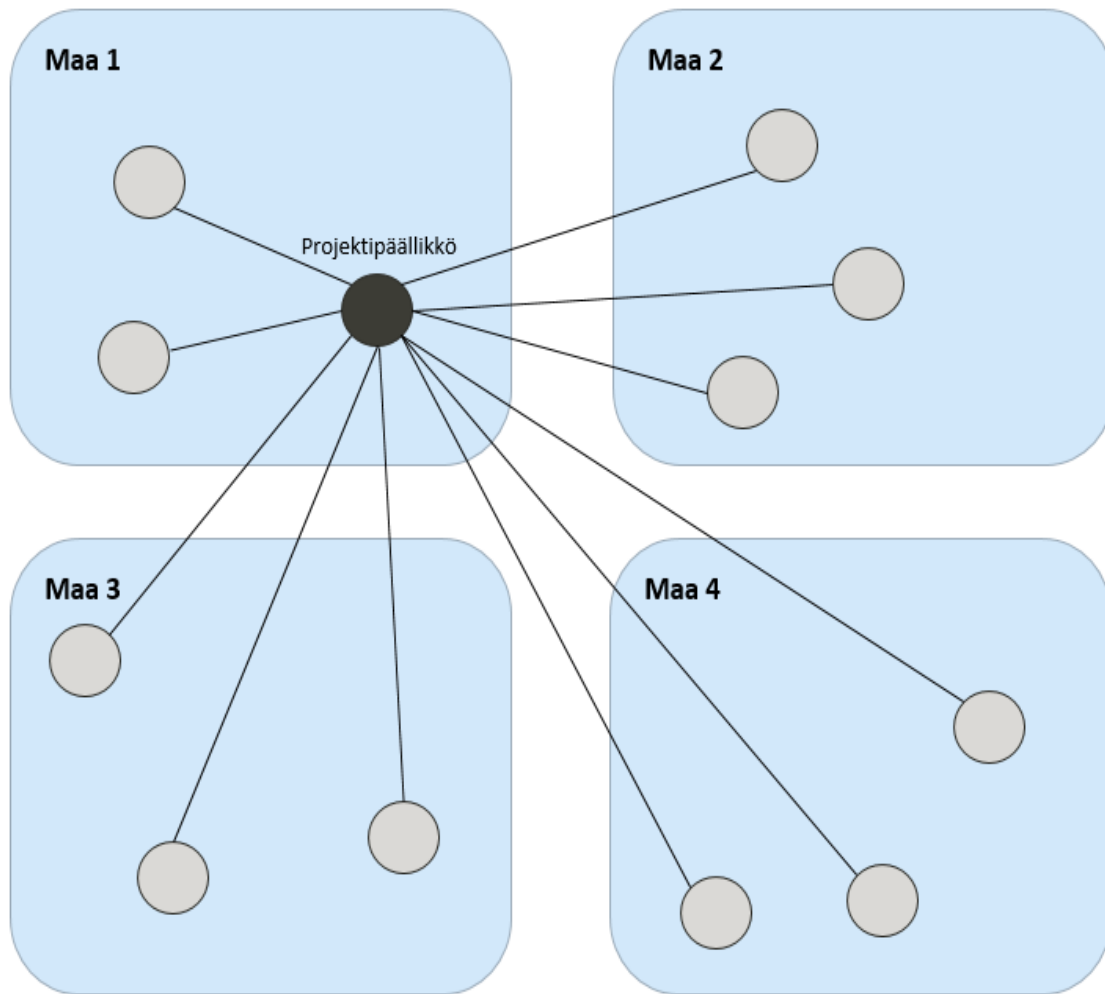
Ebertin ja De Neven (2001) mukaan henkilöstön vaihtuvuus ja sitoutuneisuus vaihtelee eri puolilla maailmaa. Euroopan ulkopuolella vaihtuvuus saattaa monesti olla selvästi suurempaa kuin Euroopassa. Yllätyksenä saattavat tulla lisäksi paikalliset työaikaan, organisaatioihin tai ammattiliittoihin liittyvät säännöt. Lisäkustannuksina pitää varautua

mahdollisiin matkustamiseen, viestintään ja testaus- sekä työvälineisiin liittyviin kuluihin. Nämä kulut saattava syntyä hieman yllättäen vasta myöhäisemmässä projektin vaiheessa. Oman kokemukseni mukaan saattaa myös olla, ettei osaa työvälineistä tai ympäristöistä välttämättä saa käyttää joka maassa. Tällaisia saattavat olla esimerkiksi laitteet, joihin liittyy voimakasta salausteknologiaa. On lisäksi varauduttava siihen, että kulttuurierot ja erilaiset kielitaidot voivat vaikuttaa työskentelyilmapiiriin.

### **3.1.2 Projektirakenne**

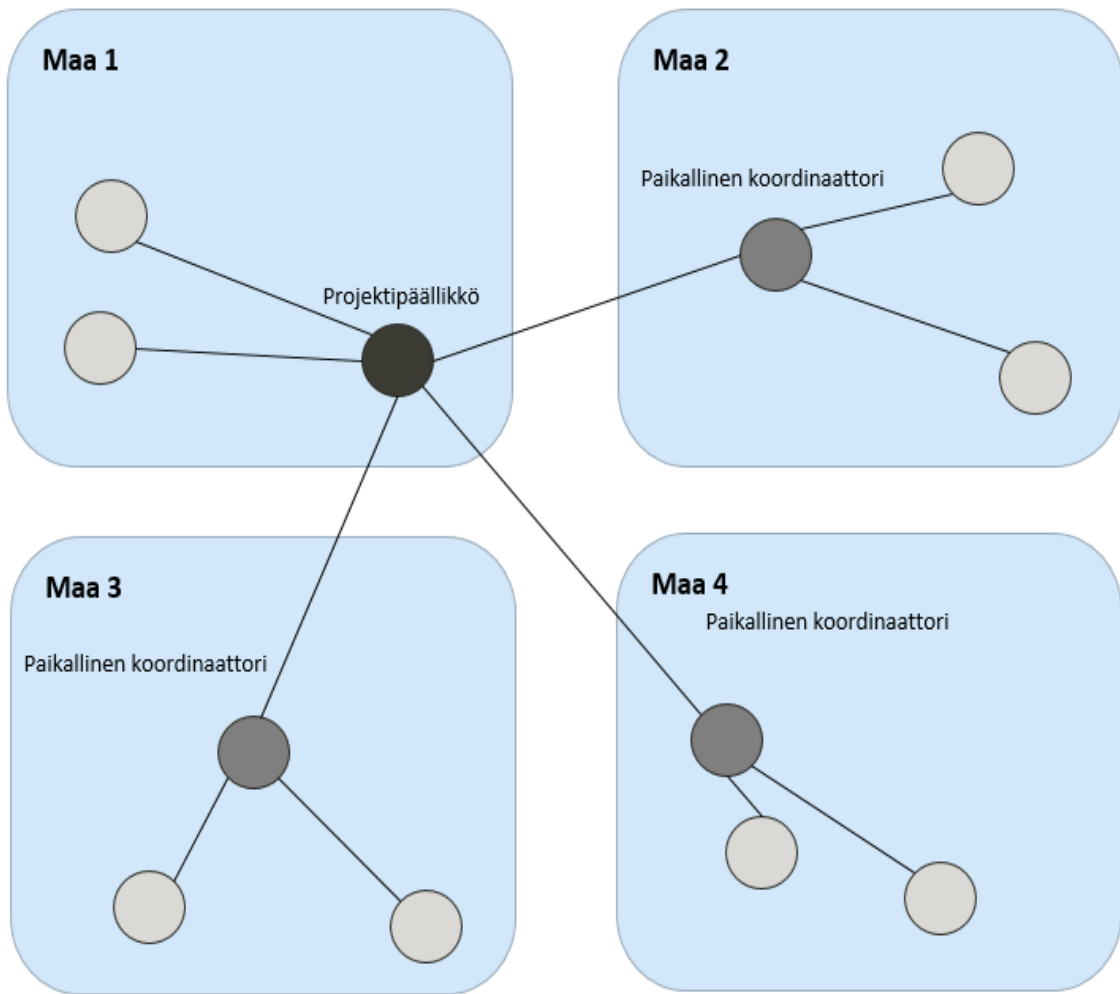
Yksi globaalien projektien johtamiseen vaikuttavat tekijä on globaali projektirakenne. Binder (2007) esittelee kaksi projektirakenteen perustyyppiä: keskitetyn ja hajautetun. Projektin rakenteeseen vaikuttavat muun muassa projektin monimutkaisuus, projektia työstävien tahojen organisaatorakenne sekä organisaation kokemus GSD-hankkeissa. Projektirakenteen kaksi päätyyppiä on esitetty kuvioissa 1 ja 2.

Keskitetyssä rakenteessa kaikki projektiryhmän jäsenet tai suurin osa heistä, raportoi suoraan projektipäällikölle (Binder, 2007). Keskitetty rakenne sopii parhaiten projekteille, jotka eivät ole kooltaan kovin suuria tai rakenteeltaan liian monimutkaisia. Keskitettyä rakennetta puoltaa vielä se, jos projektin jäsenet ovat voimakkaasti hajautuneet keskenään, he ovat tottuneet työskentelemään itsenäisesti ja heillä on riittävästi kokemusta hajautetusta työskentelystä. Keskitettyä rakennetta voidaan joutua käyttämään myös tilanteissa, joissa paikallinen koordinointi ei jostain syystä muuten ole mahdollista. Keskitetty rakenne ei ole monestikaan paras vaihtoehto kansainvälisten projektien toteutuksessa.



**Kuvio 1. Keskitetty projektirakenne (Binder, 2007).**

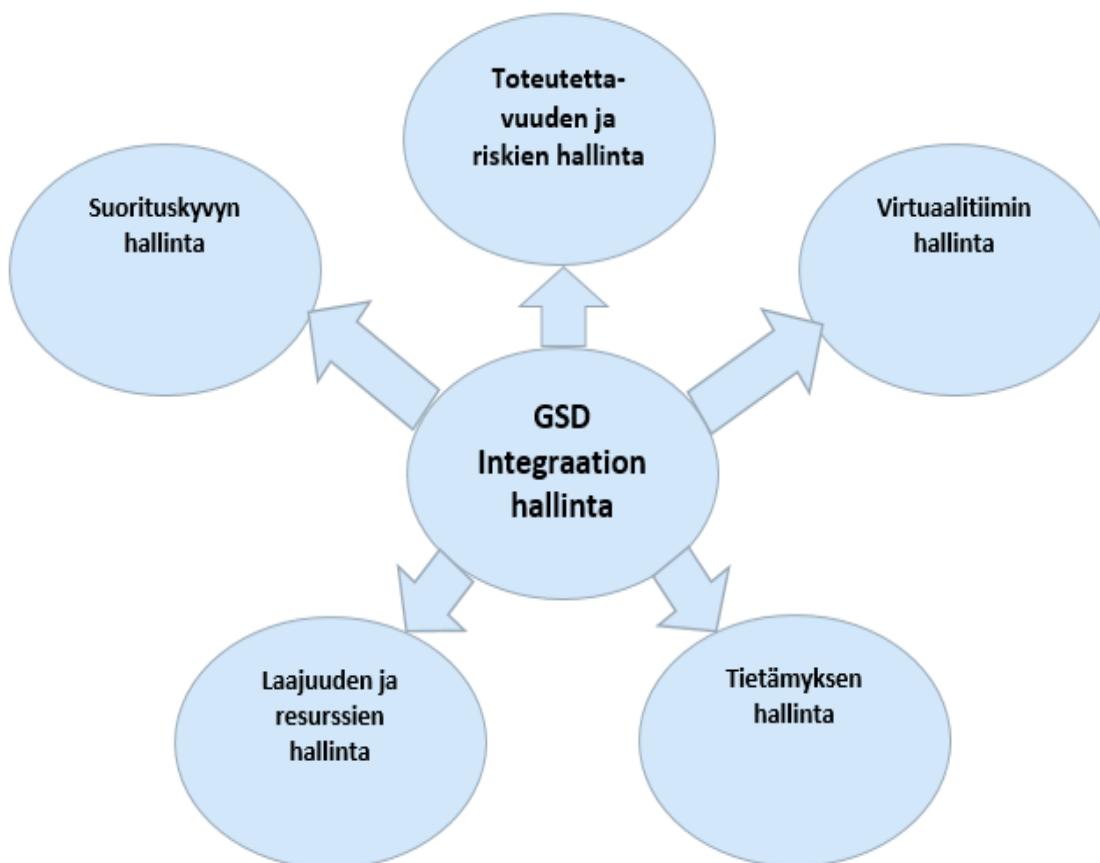
Hajautetussa rakenteessa projektiryhmän jäsen raportoi paikalliselle koordinaattorilleen, joka edustaa paikallisesti projektipäällikköä (Binder, 2007). Paikallinen koordinaattori raportoi projektipäällikölle ja saa häneltä ohjeistuksensa. Hän toimii tavallaan projektipäällikkönä omalle aliprojektilleen. Hajautettu rakenne on suositeltavin vaihtoehto suurimassa osassa kansainvälisiä projekteja.



Kuvio 2. Hajautettu projektirakenne (Binder, 2007).

### 3.2 GSD projektin hallinta

Jain ja Suman (2018) ovat esittäneet viitekehyksen GSD-projektinhallinnalle. Viitekehyksen keskeiset osat on esitelty kuviossa 3. Viitekehys koostuu projektiin liittyvistä sidosryhmistä, työkaluista, tekniikoista sekä osaamisalueista (knowledge area). Viitekehyksessä mainitut osaamisalueet mukailevat PMI:n PMBOKissa kuvattuja osaamisalueita GSD:n kannalta keskeisin osin. Osa-alueet ovat GSD Integraation hallinta, toteutettavuuden ja riskien hallinta, virtuaalitiimin hallinta, tietämyksen hallinta, laajuuden ja resursien hallinta sekä suorituskyvyn hallinta.



Kuvio 3. GSD-projektinhallinnan viitekehys (Jain & Suman, 2018).

Seuraavassa käydään tarkemmin läpi Jainin ja Sumanin (2018) esittämän projektinhallinnan viitekehysten osa-alueista integraation, tietämyksen, laajuuden ja resurssien hallintaa sekä suorituskyvyn hallintaa. Riskienhallintaa ja virtuaalitiimejä käsitellään myöhemmin omina kohtinaan.

### 3.2.1 Integraation hallinta

Jainin ja Sumanin (2018) mukaan GSD Integraation hallinnassa nimensä mukaisesti integroidaan ja koordinoitetaan muiden GSD-projektinhallinnan viitekehysten osa-alueiden keskinäistä toimintaa. Siihen kuuluu muun muassa etätiimien hallinta, projektin suunnittelu, projektin hallinnointi sekä tarvittaessa projektisuunnitelman päivittäminen. GSD

Integraation hallinnan kaksi keskeistä osa-aluetta ovat viestinnän sekä koordinaation hallinta.

Daim ja muut (2012) mukaan viestintä lisää luottamusta sekä parantaa ihmissuhteita. Viestintää tukevalla yrityskulttuurilla on tässä suuri merkitys. Viestintävälineiden käyttö parantaa viestintää hajautetussa tiimissä ja vähentää näin hajautuksesta aiheutuvia negatiivisia vaikutuksia. Tämä vaatii kuitenkin Jainin ja Sumanin (2018) mukaan sitä, että viestintä suunnitellaan riittävän hyvin. Viestinnän eri osa-alueet, kuten viestintätavat, siihen käytetyt työvälineet ja se miten usein projektiryhmän jäsenet ovat yhteydessä toisiinsa, on suunniteltava ja sovittuja käytäntöjä on valvottava työskentelyn aikana. Viestintä voi pahimmillaan jopa häiritä työntekoa, jos se on suunnittelematonta tai sitä on liikaa.

Koordinaation hallinta on eri toimipisteiden välillä toteutettavien, toisistaan riippuvaisien, toimintojen yhteensovittamista (Jain & Suman, 2018). Riittävä kommunikaatio on perusedellytys integraation onnistumiselle. GSD:ssä projektitiimin hajautuminen on myös otettava huomioon tehtävien jakamisessa, jotta toisistaan riippuvat tehtävät voidaan hoitaa mahdollisimman tehokkaasti. Hajautettujen tiimien välistä koordinaatiota voidaan parantaa muun muassa tiimiläisten vierailujen, yhteisesti sovittujen prosessien, kokousten ja työvälineiden sekä riittävän kommunikaation ja tietämyksenhallinnan avulla.

### **3.2.2 Tietämyksen hallinta**

Jainin ja Sumanin (2018) esittämän määritelmän mukaan tietämyksen hallinta muodostaa prosessin, jonka tehtävänä on hallita organisaation tietoa. Hallintaan kuuluu tiedon luominen, käyttäminen ja tiedon jakaminen. Tietämyksen hallinta on tärkeä ja keskeinen osa ohjelmistoprojekteja. Tiimin jäsenten kulttuurillisen, maantieteellisen ja ajallisen hajautumisen myötä tietämyksen hallinta korostuu vahvasti GSD-projekteissa. GSD on hyödyttänyt sovelluskehitystä monin tavoin, mutta samalla hajautetun tiimin jäsenten



välinen tiedon jakaminen on yksi sen keskeisiä huolenaiheita. GSD:n maantieteelliset ja ajalliset etäisyydet heikentävät tiedon jakamista. Lisäksi GSD:n sosiaalis-kulttuuriset ja organisatoriset etäisyydet saattavat aiheuttaa eroja siihen, miten tietoa ja tietämystä käsitellään sekä siihen, miten tietoa tulkitaan ja ymmärretään. (Razzak & Smite, 2015; Dingsøy & Smite, 2014)

Earl (2001) on ehdottanut 7-osaista viitekehystä tietämyksen hallinnan strategioille. Jainin ja Sumanin (2018) mukaan näistä seitsemästä osasta kuutta voidaan käyttää GSD:n tietämyksen hallinnassa. GSD:hen sopivat osat on esitetty kuviossa 4.



Kuvio 4. GSD Tietämyksen hallinta (Jain & Suman, 2018).

GSD:n tietämyksen hallinnan osa-alueet ovat Jainin ja Sumanin (2018) mukaan seuraavat:

- Järjestelmä-osiossa (System school) käytetään tietotekniikkaa työskentelyssä tarvittavan tiedon tallentamiseen sekä jakamiseen hajautetun projektitiimin jäsenille. Tähän osioon kuuluvat myös hajautetun tiimin työskentelyssä käytetyt etätyövälineet. (Jain & Suman, 2018)
- Kartografisessa osiossa (Cartographic school) kartoitetaan organisaation tietoja sekä huolehditaan niiden luotettavasta tallentamisesta (Earl, 2001).
- Teknillisessä osiossa (Engineering school) käsitellään liiketoimintaprosessien suunnittelua ja niiden parantamista (Earl, 2001). Erilaiset käytännöt, kuten daily scrumit, viikkopalaverit ja koodikatselmoinnit kuuluvat myös tähän osioon.
- Organisatorinen osio (Organizational school) käsittelee yhteistyöverkostojen muodostumista ja tiedon jakamista (Earl, 2001).
- Tila-osiossa (Spatial school) pyritään työskentelytilan suunnittelulla parantamaan tiedon vaihtoa (Earl, 2001). GSD:n tapauksessa tilat voitaneen käsittää myös virtuaalisina.
- Strateginen osio (Strategic school) korostaa organisaation strategioita, jotka voivat auttaa tietämyksen jakamisessa (Earl, 2001).

Anwar ja muut (2019) suosittelevat tekemänsä kirjallisuuskatsauksen perusteella kansainvälistä ohjelmistokehitystä tekeville organisaatioille seuraavia toimenpiteitä tietämyksen hallinnan parantamiseksi:

- Tarvittaessa organisaation tulisi tarjota sovelluskehittäjille riittävästi tukea tietämyksen jakamisen helpottamiseksi.
- Organisaatiossa tulisi valita niin sanottuja ”kulttuurilähettiläitä”, jotka tuntevat hyvin organisaatiossa edustettuja kulttuureja. He voivat auttaa tulkitsemaan erilaisista taustoista tulevien henkilöiden käytöstä ja kommunikaatiota. Heidän avullaan konfliktinhallinta saattaa myös ratkaisevasti tehostua.
- Kielieroista johtuvien ongelmien minimoimiseksi, organisaatio voi hyödyntää niin sanottuja ”kulttuuritietoisuusohjelmia”. Tämä voidaan tehdä esimerkiksi niin,

että organisaatio nimeää paremmin kieltä ja kulttuuria tuntevia henkilöitä auttamaan viestinnän oikeassa tulkinnassa.

- Organisaation tulisi mahdollisuuksien mukaan hyödyntää henkilöitä, jotka tuntevat useampia kulttuureita. Heitä tulisi käyttää välittävässä rooleissa eri kulttuureja edustavien henkilöiden välillä.

### **3.2.3 Laajuuden ja resurssien hallinta**

Laajuuden ja resurssien hallintaan kuuluvat päätökset siitä, mitä tehtäviä sisällytetään projektiin, aikataulun ja kustannusten arviointi sekä niiden seuranta. Siihen kuuluvat osat ovat laajuuden, kustannusten ja aikataulun hallinta. (Jain & Suman, 2018)

PMBOKin mukaan projektin laajuuden hallinta vastaa siitä, että projekti sisältää kaiken tarvittavan, jotta se voidaan suorittaa menestyksekkäästi. Toisaalta projektin laajuuden hallintaan kuuluu myös sen varmistaminen, ettei projektiin oteta sen toimittamisen kannalta ylimääräisiä tehtäviä (Project Management Institute, 2013, s. 105). GSD:ssä maantieteelliset, organisatoriset ja sosiokulttuurilliset etäisyydet hankaloittavat yhteisen käsityksen syntymistä projektiryhmälle siitä, mitä kaikkea projektiin kuuluu (Jain & Suman, 2018). Tämän takia eri etätiimien osallistaminen mukaan projektin laajuuden määrittelyyn on tärkeää.

Kustannusten hallinta sisältää muun muassa projektin kustannusten arvioinnin, budjetoinnin ja toteuman valvonnan (Project Management Institute, 2013, s. 193). Paikallisesti toteutettuihin projekteihin verrattuna, GSD-projekteihin tulee mukaan hajautetusta työskentelystä syntyvät lisäkulut. Näitä ovat muun muassa infrastruktuurin huoltamiseen, matkoihin ja koulutukseen liittyvät kustannukset (Jain & Suman, 2018). Lisäksi kansainvälinen työskentely saattaa vaatia myös panostusta kielellisten tai kulttuurillisten erojen minimointiin. Kansainvälisyyden ja hajautuksen tuomien kustannusten hillitsemistä helpottavat prosessien noudattaminen, tehokkaasti toimivat tiimit, asiakkaan osallistaminen ja hyvä tietämyksen hallinta.

Aikataulun hallinnassa määritellään projektin tehtävät ja niiden kestot, suunnitellaan projektin aikataulu sekä valvotaan sen toteutumista. GSD:ssä hajautettu työskentely aiheuttaa helposti aikatauluun ja koordinointiin liittyviä ongelmia, jotka heijastuvat projektin suoritukseen (Jain & Suman, 2018). Herbsleb ja Moitra (2001) esittävät, että hajautettu työskentely pidentää työskentelyyn vaadittavaa aikaa verrattuna työhön, joka tehdään keskitetysti vain yhdessä paikassa. Syinä tarvittavan työajan pidentymiseen ovat esimerkiksi viiveet kommunikaatiossa eri toimipisteiden välillä, koordinaatiovaikeudet ja väärinkäsityksistä aiheutuva ylimääräinen työ. Hajautetun työskentelyn aikataululle aiheuttamia paineita voidaan pienentää hyvällä johtamisella sekä kommunikaatioteknologian käytöllä.

Projektin laajuuteen vaikuttavat myös projektin aikana siihen tulevat muutospyynnöt. Ohjelmistokehitys on monimutkainen prosessi, jossa kaikkia tulevan järjestelmän vaatimuksia ei useinkaan pystytä määrittämään tarkasti työn alussa. Tämän takia vaatimukset monesti muuttuvat ja tarkentuvat kehitystyön edetessä ja muutokset ovatkin Anwer ja muut (2019) mukaan käytännössä väistämättömiä. Muutoksia hallinnoidaan muutostenhallinnan prosesseilla.

#### **3.2.4 Suorituksen johtaminen**

GSD:ssä suoritus ja siihen liittyvä laatu koostuvat Jainin ja Sumanin (2018) mukaan prosesseista ja tuotteen laadusta. Projektissa tulisi selkeästi määrittää kriteerit ja odotukset hyvälle suoritukselle. Hajautetun projektiryhmän jäsenen tulisi tietää varmasti, mitä häneltä odotetaan. Yleisesti projektin katsotaan suoriutuneen hyvin, jos se pystyy täyttämään sille asetetut tehtävät sovitussa laajuudessa budjetin ja aikataulun puitteissa. Lisäksi projektin työn laadun tulisi olla hyvää ja sen pitäisi täyttää asiakkaan vaatimukset. Sangwan ja muut (2006, s. 138–139) esittävät, että laadunvarmistus globaalissa kontekstissa eroaa keskitetyimmässä ympäristössä toteutetusta projektista. Hajautetussa työskentelyssä puuttuu epämuodollinen ja säännöllinen vuorovaikutus tiimin jäsenten väliltä.

Tämä vuorovaikutus tuo parhaassa tapauksessa laatuun liittyvät uhat näkyviin jo hyvin varhaisessa vaiheessa, ennen kuin ne ehtivät muodostua edes varsinaisiksi ongelmiksi. GSD-projekteissa laatuun liittyvät ongelmat huomataan monesti vasta myöhäisemmässä vaiheessa, jolloin niiden korjaaminen maksaa enemmän aikaa ja resursseja.

Jain ja Suman (2018) laskevat ohjelmistokehityksen seurantaan ja arviointiin liittyvät tehtävät prosessien johtamiseen kuuluviksi. Heidän mukaansa GSD-projekteissa korostuu erityisesti se, että projektipäällikön tulee selvästi määrittää hajautetulle tiimille vaadittavat välitavoitteet, määrääjat ja toimitettavat lopputuotteet. Projektin tilanteen hahmottamisessa ja työn ohjaamisessa apua tuovat projektinhallinnan työvälineet, eri tiimien käytössä olevien työvälineiden integroiminen toisiinsa sekä kommunikaatiovälineet. Monilla organisaatioilla on käytössä standardiprosessit sovelluskehitysprojekteja varten (Sangwan ja muut, 2006, s. 140). Projektit, ja varsinkin GSD-projektit, eroavat yleensä toisistaan niin paljon, että prosesseja joudutaan mukauttamaan kyseessä olevan projektin tarpeisiin. Sopivien prosessien valinnan lisäksi on valittava myös tapa, jolla niitä valvotaan ja seurataan. Valvonnassa keskeistä olisi kyetä seuraamaan projektin ajantasaista edistymistä. GSD-projekteissa ongelmat näkyvät helposti vasta liian myöhäisessä vaiheessa.

Sangwan ja muut (2006, s. 140–141) esittävät seuraavan heuristiikan, joka saattaa antaa vihjeen siitä, että GSD-projektissa on ongelmia.

- Äkillisesti lisääntynyt kommunikaatio. Ongelmatilanteessa ihmiset etsivät ratkaisua ongelmaansa. Hajautetussa työskentelyssä apua ja ratkaisuja joudutaan etsimään kommunikaatiovälineiden avulla. Kommunikaatiovälineiden käytöstä saatavaa olla saatavilla statistiikkaa. Kommunikaation lisääntymiseen voi olla muitakin syitä, mutta se saattaa olla yksi merkki ongelmista.
- Kehityssprintissä toimitettavan osion laajuuden epätavallinen pienentäminen. Ketterässä kehityksessä on normaalia säätää sprintin laajuutta, mutta jos kokenut kehitystiimi saa aikaan odotettua selvästi pienemmän tuotoksen, kannattaa asian juurisyy selvittää.

- Kehittäjien moraalit. Tiimiläiset, jotka päivittäin ovat tekemisissä kehitystyön kanssa, saattavat osata ennakoita tulevia ongelmia jo paljon ennen kuin niitä muilla mittareilla on mahdollista havaita. Heidän mielialojaan kannattaa kuunnella tarkasti.
- Toistuvat muutokset vaatimuksissa tai suunnittelussa. Projektin alun jälkeen, kun työskentelyn tahti on jo vakiintunut, kannattaa tarkkailla muutoksia ja selvittää niiden syyt.

Sangwan ja muut (2006, s. 142–143) mukaan ohjelmistossa voi olla erilaisia virhetyyppejä. Niitä voivat olla esimerkiksi koodausvirheet, virheelliset ohjelmiston piirteet, puuttuvat ohjelmiston piirteet, väärin ymmärretyt ohjelmiston piirteet ja yleisesti järjestelmän toimintaan liittyvät ongelmat, kuten huono suorituskyky. Ohjelmistokehityksen hajautuneisuus ei vaikuta yleensä koodausvirheiden esiintymiseen, mutta esimerkiksi väärinymmärrykset ohjelmiston toimintaan liittyen tai esimerkiksi erilaiset oletukset mitaysiköihin tai muihin paikallisesti vaihteleviin standardeihin saattavat lisääntyä ohjelmistokehityksen hajautumisen myötä. GSD-projekteissa näihin ongelmiin voi pyrkiä vaikuttamaan muun muassa noudattamalla sovittuja standardeja ja järjestämällä erilaisia katselmointeja jo ohjelmiston kehitysvaiheessa. On tärkeää varmistaa, että kaikilla hajautetun projektiryhmän jäsenillä on yhteinen näkemys työlle ja ohjelmistolle asetetuista vaatimuksista.

Ohjelmistotuotteen laatua mitataan monesti siitä löytyneiden virheiden määrällä. Itse haluaisin sisällyttää laadun määritelmään lisäksi sen, miten laadukkaasti ja standardeja noudattaen ohjelmisto on koodattu. Pitkällä aikavälillä laadukkaasti tehdyn ohjelmistokomponentin ylläpito on selvästi helpompaa ja uskoakseni myös halvempaa kuin huonosti tehdyn.

### 3.3 Riskienhallinta ja toteutettavuus

Kirjallisuudessa riskille löytyy useita erilaisia määritelmiä. PMBOK:ssa riski määritellään epävarmana tapahtumana, jolla voi olla positiivinen tai negatiivinen vaikutus johonkin projektin osa-alueeseen, esimerkiksi laajuuteen, aikatauluun, kustannuksiin tai laatuun (Project Management Institute, 2013, s. 310). Tao (2008) puolestaan määrittelee riskin projektin epäonnistumiseen liittyvänä tekijänä. Riski kuvaa todennäköisyyttä, jolla projektille tapahtuu negatiivisia asioita. Nämä negatiiviset tapahtumat voivat liittyä muun muassa siihen, ettei projekti pysy aikataulussa, sen kustannukset kasvavat tai sen ei muuten ole mahdollista saavuttaa tavoitteitaan esimerkiksi laadun tai laajuuden osalta. Huomion arvoista tässä on, että PMBOKissa esitetyn määritelmän mukaan riskillä voi olla positiivinen tai negatiivinen vaikutus projektiin. Tässä keskitytään kuitenkin vain riskien negatiivisiin vaikutuksiin. Sangwan ja muut (2006, s. 68) painottavat, että riskissä on nimenomaan kyse tapahtumaan liittyvästä epävarmuudesta. Jos on esimerkiksi varmaa, että tapahtuma toteutuu, kyse ei ole enää riskistä, vaan silloin puhutaan jo ongelmasta.

Ebert ja muut (2016) esittelevät riskienhallinnan osa-alueina riskien tunnistamisen, analysoinnin, arvioinnin, käsittelemisen ja seurannan. Riskienhallinta on hyvin tärkeää projektin menestykselle. Jos projektin riskit tunnistetaan riittävän aikaisin ja niihin reagoidaan suunnitellusti, projektin onnistumisen mahdollisuudet kasvavat (Sangwan ja muut, 2006, s. 77). Siksi projektin riskienhallinnan pitäisikin alkaa jo riittävän varhaisessa vaiheessa projektia käynnistettäessä. Riskit ja niihin suunnitellut vastatoimet kirjataan projektin riskienhallintasuunnitelmaan, jota tulee tarkastella säännöllisesti koko projektin ajan.

Taon (2008) mukaan ohjelmistokehitysprojekteihin liittyvät riskit juontavat juurensa yleensä joko projektissa käyttöön otettuihin uusiin tekniikoihin tai teknologioihin, järjestelmävaatimuksiin, järjestelmän arkkitehtuuriin, järjestelmän suorituskykyyn tai organisaatorisiin seikkoihin.

Erilaisia lähestymistapoja riskienhallintaan on kirjallisuudessa useita. Avdoshin ja Pesotskaya (2011) esittävät, että riskienhallintaprosessiin kuuluu kiinteästi seuraavat viisi perustehtävää.

1. Riskien tunnistaminen. Kartoitetaan tyypilliset ja havaittavissa olevat riskit.
2. Riskien arviointi. Riskeille määritellään todennäköisyys ja arvioidaan vaikutukset riskin toteutumiselle.
3. Toimenpiteiden suunnittelu. Riskit analysoidaan ja niiden varalle kirjataan valmiiksi toimintasuunnitelma.
4. Riskien seuranta. Tunnistettujen riskien tilanne käydään säännöllisesti läpi koko projektin ajan. Samalla myös pyritään tunnistamaan mahdolliset uudet riskit.
5. Havaintojen dokumentointi ja toiminnan kehittäminen. Pyritään keräämään opit projektin riskienhallinnasta jatkoa ja tulevia projekteja varten.

Kun potentiaaliset riskit on tunnistettu, ne tulee arvioida. Arvioinnissa riskille pyritään määrittelemään todennäköisyys ja arvioimaan sen vaikutus toteutuessaan. Sen jälkeen tulee suunnitella jokaista riskiä varten toimenpiteet. Toimenpiteet voivat vaihdella metodologiasta riippuen, mutta neljä pääasiallista toimenpidettä Avdoshinin ja Pesotskayan (2011) mukaan ovat seuraavat:

1. Riskin pienentäminen. Tehdään suunnitelma riskin toteutumisen varalle ja sen todennäköisyyden pienentämiseksi.
2. Riskin välttäminen. Pyritään poistamaan riski esimerkiksi niin, että rajataan projektia siten, ettei riski enää kuulu projektin laajuuteen.
3. Riskin jakaminen. Pyritään esimerkiksi siirtämään riski toisen osapuolen vastuulle.
4. Riskin hyväksyminen. Ei tehdä riskille toistaiseksi mitään. Hyväksytään riski ja sen mahdollinen toteutuminen.



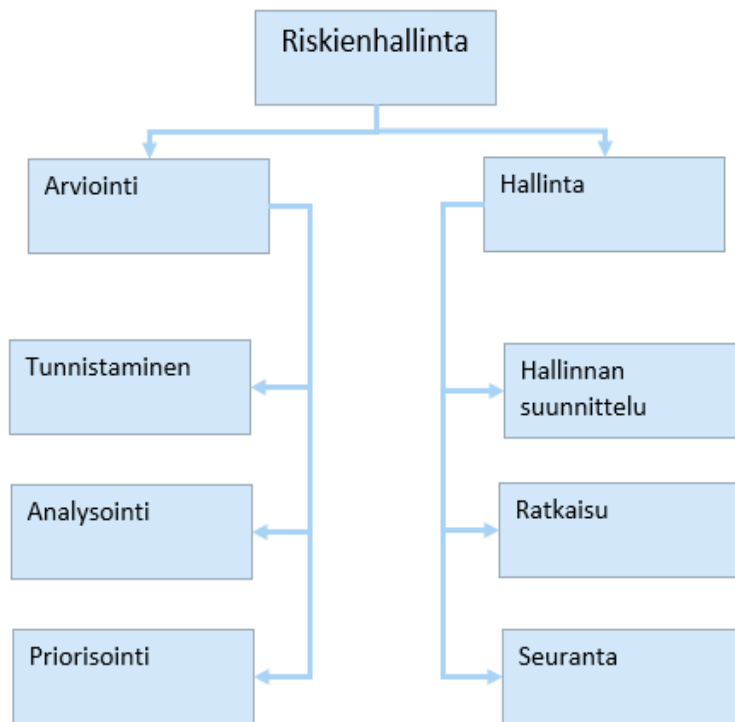
### 3.3.1 Riskienhallinnan viitekehys

Esittelen seuraavassa lyhyesti kaksi tunnettua riskienhallinnan viitekehystä.

Boehm (1989) on esittänyt yhden tunnetuimmista ja vanhimmista ohjelmistoriskejä kuvaavista malleista. Malli on esitelty kuviossa 5. Tässä mallissa riskienhallinta käsittää kaksi pääprosessia. Ylätason prosessit ovat riskien arviointi sekä riskien hallinta. Näillä kummallakin pääprosessilla on vielä kolme alaproessia, jotka edelleen jakautuvat osaprosesseiksi.

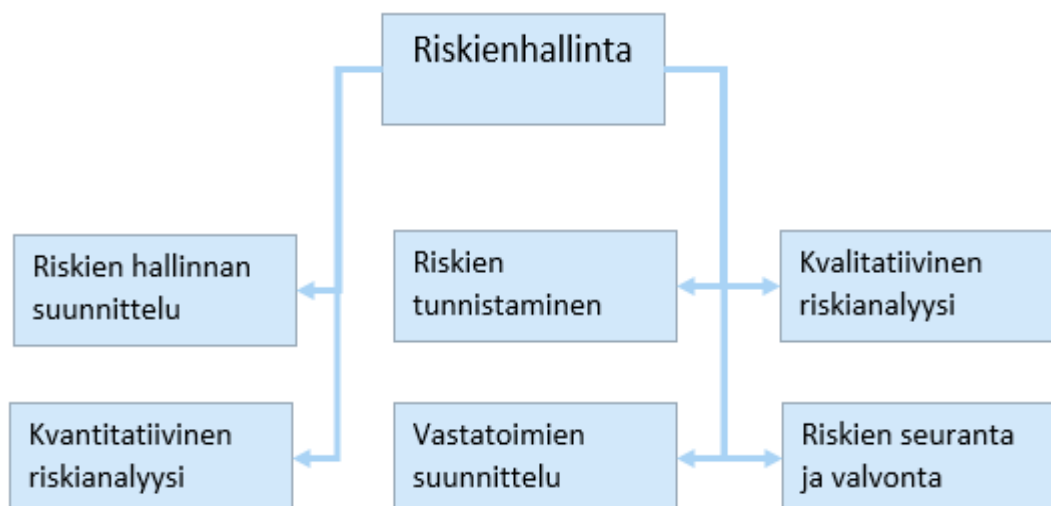
Riskien arvioinnissa pyritään tunnistamaan potentiaaliset riskit. Sen jälkeen riskit analysoidaan, jolloin niiden vaikutukset ja todennäköisyys niiden realisoitumiselle pyritään määrittämään. Sen lisäksi riskien arvioinnissa riskeille määritellään prioriteetti. Riskin prioriteetti lasketaan yleensä kertomalla sen todennäköisyys mahdollisella vaikutuksella. (Chadli ja muut, 2015)

Toinen Boehmin (1989) riskienhallinnan pääprosesseista koostuu riskien hallinnan suunnittelusta, ratkaisusta ja seurannasta. Hallinnan suunnittelussa päätetään, mitä riskille tehdään. Vaihtoehdot ovat riskin jakaminen, välttäminen, pienentäminen tai hyväksyminen. Ratkaisuvaiheessa suunnitellut toimenpiteet toteutetaan. Viimeisenä aliproessina on riskien seuranta. Siinä riskejä seurataan ja toteutetaan tarvittaessa korjaavia toimenpiteitä.



**Kuvio 5. Riskienhallinnan pääprosessit (Boehm, 1989).**

PMI:n PMBOK nimeää riskienhallinnan yhdeksi projektinhallinnan osaamisalueeksi (Project Management Institute, 2013, s. 61). PMI:n malli on esitetty kuviossa 6. Siihen kuuluvat prosessit ovat riskien hallinnan suunnittelu, riskien tunnistaminen, kvalitatiivinen ja kvantitatiivinen riskianalyysi, vastatoimien suunnittelu sekä riskien valvonta ja seuranta. PMI:n mallissa riskien lopputulos voi olla myös positiivinen, joten negatiivisten vaikutusten ehkäisemisen lisäksi tavoitteena on lisätä riskien positiivisia vaikutuksia projektille.



**Kuvio 6. Riskienhallinta projektinhallinnassa (Project Management Institute, 2013, s. 312).**

Chadli ja muut (2015) mukaan yleinen tapa riskienhallinnassa, viitekehysten lisäksi, ovat erilaiset heuristiikat ja tarkistuslistat. Ne on monesti laadittu eri sidosryhmien kokemusten perusteella. Erilaiset viitekehykset, tarkistuslistat ynnä muut sellaiset muistuttavat siitä, mitä asioita riskienhallinnassa tulee ottaa huomioon. Ne ovat kuitenkin vain ohjeita, joita on muokattava kulloisiinkin tarpeisiin.

### 3.3.2 Riskienhallinta ja GSD

GSD-projekteihin liittyvät etäisyydet tuovat kansainvälisten projektien riskeihin omat lisänsä verrattuna paikallisesti toteutettuihin projekteihin (Jain & Suman, 2018). GSD-projektit ovat usein melko laajoja kokonaisuuksia, joiden monimutkaisuutta kansainvälisyys lisää. Ei-kansainvälisiin projekteihin verrattuna GSD-projekteissa joudutaan työskentelemään maantieteellisten etäisyyksien, kulttuurillisten ja organisatoristen erojen sekä eri aikavyöhykkeiden paineessa. GSD:ssä esimerkiksi viestintä, koordinointi ja yhteistyö ovat alttiimpia ongelmille. Kaikki tämä lisääntynyt monimutkaisuus näkyy myös GSD-projektien kasvaneina riskeinä. Sangwan ja muut (2006, s. 68) mukaan GSD:n riskit liittyvät erityisesti projektin koordinointiin, kommunikointiin, tietämyksen jakamiseen ja riskien

tunnistamiseen sekä ongelmanratkaisuun. Riskinhallintaa on näiden asioiden osalta erityisesti tehostettava GSD-projekteissa.

### **3.4 Kulttuurilliset seikat**

GSD-projekteissa tiimin jäsenet työskentelevät hajautettuna laajalle maantieteelliselle alueelle. Tämä tuo usein mukanaan työskentelyn eri kulttuureihin kuuluvien ihmisten kanssa. Tiimiläisten erilaiset taustat ja lähtökohdat voivat monessa tilanteessa olla voimavara, mutta erilaisia kulttuureita edustavien henkilöiden välisessä työskentelyssä saattaa myös syntyä hankausta, jolloin työn tekeminen vaikeutuu.

#### **3.4.1 Kulttuurien väliset erot**

Ebertin ja De Neven (2001) mukaan kulttuurierot ovat yksi keskeinen GSD:n ongelma-kohta. Erilaiset yrityskulttuurit voivat johtaa muun muassa erilaisiin työtapoihin, normeihin, arvoihin ja viestintään. Kulttuurieroja syntyy erilaisten valtioiden ja kansallisuuksien välille. Usein ei kuitenkaan tiedosteta, että myös kansallisten organisaatioiden tai erilaisten ammattikuntien välille voi helposti syntyä kulttuurieroja (Smith, 2001). Esimerkiksi insinöörit ja juristit saattavat painottaa argumentoinnissaan erilaisia asioita ja arvottaa asioita eri tavoilla. Erilaiset kansalliset- ja yrityskulttuurit johtavat siihen, että myös projektinhallinnassa pitää ottaa huomioon kulttuurierot eri maiden ja organisaatioiden välillä (Niazi ja muut, 2016a).

Niazi ja muut (2016b) tulivat tutkimuksessaan siihen tulokseen, että GSD:n yleisin projektin johtamiseen liittyvä heikkous oli tiimien huono tietämys muiden tiimiläisten kulttuureista. Tiimiläisten hajautuminen eri maihin ja eri kulttuureihin vaikeuttaa GSD-projektin johtamista. Ilman riittävää ymmärrystä eri kulttuureista saattaa syntyä paljon väärinkäsityksiä. Huono vieraiden kulttuurien ymmärtäminen vaikeuttaa projektin työskentelyä muun muassa heikentämällä kommunikaatiota ja vähentämällä luottamusta.

### 3.4.2 Kommunikaatio

Ohjelmistokehityksessä projektitiimin ja muiden sidosryhmien välisellä yhteistyöllä ja kommunikaatiolla on suuri merkitys. Ohjelmistokehitys ei ole yleensä mahdollista ilman laajan joukon tietoja ja asiantuntemusta. Layzell ja muut (2000) mukaan kommunikaation onnistuminen vaikuttaa suoraan hajautetusti toteutetun projektin lopputuloksen laatuun. Hajautetussa projektissa jatkuva viestintä on välttämätöntä. Liian vähäinen viestintä vaikeuttaa projektin johtamista. Monien ohjelmiston suunnittelussa tai toteutuksessa tehtyjen huonojen ratkaisujen tai jopa suoranaisten virheiden taustalta löytyy kommunikaation epäonnistumisesta aiheutunut väärinkäsitys.

Hajautettua sovelluskehitystä tehtäessä on pakko käyttää kommunikaatiovälineitä, koska henkilökohtaiset tapaamiset eivät usein ole mahdollisia tai ne ovat säännöllisesti toteutettuina liian kalliita. Kommunikaatiovälineiden käyttäminen suodattaa osan viestinnän tasoista ja vähentää viestinnän määrää. Tämän takia Farias ja muut (2012) kehottavat edistämään ja lisäämään viestintää ottamalla käyttöön mahdollisimman hyviä viestintätyökaluja. Tiimiläisiä tulisi kannustaa keskenään sekä muodolliseen että epämuodolliseen kommunikaatioon. Lisäksi jos hajautetussa tiimissä on mukanaan eri äidinkieliä puhuvia jäseniä, he kehottavat myös, väärinkäsitysten minimoimiseksi, käyttämään viestinnässä selkeää yleiskieltä.

Bano ja muut (2016) mukaan keskeiset hankaluudet maantieteellisesti hajautetun tiimin työskentelyssä liittyvät työskentelyyn eri aikavyöhykkeillä sekä henkilökohtaisen, kasvokkain tapahtuvan, kommunikaation vähäisyyteen. Nämä seikat vaikeuttavat reaaliaikaista viestintää ja siten vaikeuttavat sekä hidastavat tiedon siirtoa. He kartoittivat tutkimuksessaan, mitä keinoja on käytetty maantieteellisestä hajautuksesta aiheutuneiden ongelmien kiertämiseen. Käytettyjä keinoja olivat muun muassa määrällisesti riittävä viestintä sopivia viestintävälineitä, etenkin neuvottelupuheluita, käyttäen ja kokousten laadukas valmistelu. Myös joustavaa työaikaä käytettiin, jotta eri aikavyöhykkeillä työskentelevien

henkilöiden olisi mahdollista osallistua yhteisiin kokouksiin ja viestiä keskenään reaaliaikaisesti.

Hajautetusti työskentelevän tiimin kesken on Bano ja muut (2016) mukaan vähemmän epävirallista kommunikaatiota kuin keskitetyksi työskentelevällä tiimillä. Kommunikaatioon vaikuttavat eri toimipisteiden välinen fyysinen etäisyys, aikaerot sekä se, ettei toisen toimipisteen työntekijöihin välttämättä ole vähäisemmän kanssakäymisen johdosta muodostunut henkilökohtaista sidettä. Tällöin on vaarana, että tiimiläiset jakautuvat eri leireihin ja kommunikaatio sekä tiedon jakaminen eri ryhmien välillä heikentyy entisestään. Tämän takia organisaation henkeä tulisi nostaa. Olisi tärkeää luoda tunne ja tietoisuus yhteisistä tavoitteista. Tässä auttavat liiketoimintaprosessien yhtenäistäminen sekä selkeästi määritellyt roolit ja vastuut. Teknisistä apuvälineistä voi myös tässä yhteydessä olla hyötyä. Esimerkiksi työvälineiden integraatiolla on Bano ja muut (2016) tekemän tutkimuksen mukaan saavutettu hyviä tuloksia viestinnässä koettujen ongelmien vähentämisessä.

Kommunikaatiosuunnitelman tekeminen on tärkeää, jotta kyetään varmistamaan tehokas viestintä hajautetussa tiimissä (Denhere ja muut, 2015). Viestintäsuunnitelmalla varmistetaan, että riittävä informaatio toimitetaan oikeassa muodossa, oikeaan aikaan, oikeille henkilöille. Asioita, joita viestintäsuunnitelmassa tulisi ottaa huomioon ovat ainakin seuraavat:

- Käytettävä kieli
- Kuinka tiimiä tiedotetaan edistymisestä
- Milloin voidaan järjestää henkilökohtaisia tapaamisia
- Viestintäteknologian käyttöä varten tarvittava koulutus
- Mitä viestintävälineitä käytetään
- Milloin käytetään synkronista ja milloin asynkronista kommunikaatiota

## 4 Virtuaalitiimit

Nykyinen kommunikaatioteknologia on mahdollistanut tiimien toimimisen huomattavasti laajemmalla keinovalikoimalla kuin aikaisemmin. Kommunikaatioteknologia mahdollistaa tiimien toimimisen hajautetusti, ilman fyysistä vuorovaikutusta tiimiläisten kesken. Projektien ja tiimien hajauttaminen aiheuttaa kuitenkin vaikeuksia niiden johtamiseen. Sen takia monet organisaatiot ovat Casey'n ja Richardsonin (2006) mukaan muodostaneet strategian virtuaalitiimeistä. Virtuaalitiimin jäsenet ovat jakautuneet maantieteellisesti, organisatorisesti tai ajallisesti. He työskentelevät projektissa koordinoitujen toimintaansa tietoliikennetekniikkaa käyttäen. Virtuaalitiimin jäsenillä saattaa olla eri äidinkieliä, he voivat olla eri maiden kansalaisia ja kuulua eri organisaatioihin. Virtuaalitiimit ovat nykyään yleinen ja dynaaminen tapa tehdä tiimityötä ohjelmistokehityksessä. Virtuaalitiimit tarjoavat tiimille tietyissä tilanteissa mahdollisuuden työskennellä joustavammin ja taloudellisesti kannattavammin kuin perinteiset tiimit. Virtuaalitiimit eivät ole sidottuja tiettyyn maantieteelliseen alueeseen, ja niihin on usein mahdollista hankkia kattavampaa ja monipuolisempaa osaamista kuin perinteiseen tiimiin. Tiimi voi myös toimia virtuaalisesti, vaikka sen jäsenet työskentelisivätkin esimerkiksi samassa maassa ja samalla paikkakunnalla. (Casey & Richardson, 2006; Jawadi & Boukef-Charki, 2009; Jain & Suman, 2018)

Vaikka monissa lähteissä suositellaan virtuaalitiimien käyttöä, Ebert ja De Neve (2001) esittävät tekemänsä tapaustutkimuksen perusteella eriävän näkemyksen ja suosittelivat keskitettyjen tiimien käyttöä. He suosittelivat lisäksi tiimin jäsenille täyttä allokaatiota, jolloin he työskentelevät vain yhdessä projektissa, eivätkä muut tehtävät häiritse heidän työtään. Näin saadaan tiimin resurssit paremmin käyttöön ja tiimin jäsenet tehokkaammin kommunikoimaan keskenään.

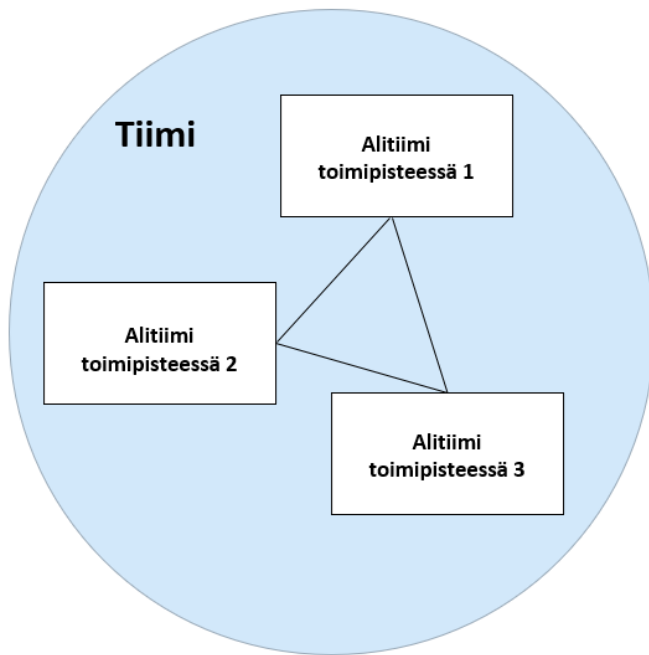
## 4.1 Kommunikaatio

Yksi suurimmista vaikeuksista virtuaalitiimien toiminnassa on kommunikaation toimivuus. Kommunikaatio voidaan jakaa synkroniseen ja asynkroniseen kommunikaatioon. Synkronisessa kommunikaatiossa sanomien välittäminen tapahtuu reaaliajassa. Asynkroninen viestintä ei ole reaaliaikaista, eivätkä viestin lähettäjä tai vastaanottaja ole ajallisesti sidottuja toisiinsa. Niazi ja muut (2016a) tarkoittavat synkronisella kommunikaatiolla pääasiassa henkilökohtaista keskustelua. Sen määrä hajautetussa tiimissä voi olla hyvin vähäistä, elleivät jäsenet matkusta toistensa luokse. Synkronisena kommunikaationa voi pitää myös televiestintäsovelluksien tai puhelimen avulla tapahtuvaa viestintää. Dubé ja Robey (2008) ehdottavat yhtenä ratkaisuna kommunikaation parantamiseen sitä, että virtuaalitiimit mahdollistaisivat jäsentensä fyysisen tapaamisen. Tämä voisi tapahtua esimerkiksi projektin aloituskokouksessa.

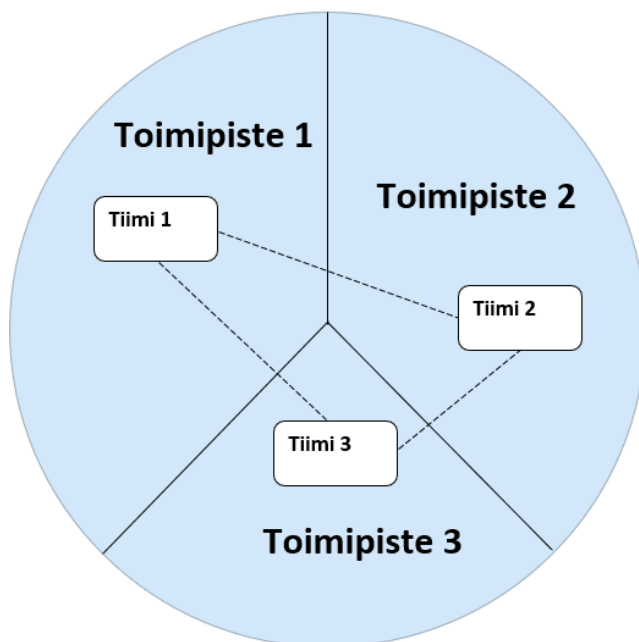
## 4.2 Hajautetun tiimin rakenne

GSD mahdollistaa työn jakamisen itsenäisesti toteutettaviin osiin, joita voidaan toteuttaa rinnakkain useammassa eri paikassa. Tätä varten Conchúir ja muut (2009b) esittävät kaksi pääasiallista tapaa hajauttaa tiimit työskentelemään eri toimipisteissä. Tavat ovat virtuaalitiimi ja löyhästi kytketyt tiimit. Eri tavat organisoida tiimityö on esitetty kuvioissa 7 ja 8. Kuvion 7 kaltaisesti organisoidussa virtuaalitiimissä työtehtävät jaetaan lähes kuten paikallisesti toimivassa tiimissä. Eri toimipisteessä työskentelevät henkilöt muodostavat kuitenkin omat alitiiminsä. Kuviossa 8 esitetyssä löyhästi kytketyssä tiimissä työt jaetaan selvästi rajatumpiin kokonaisuuksiin ja jokainen toimipiste muodostaa oman erillisen tiiminsä.





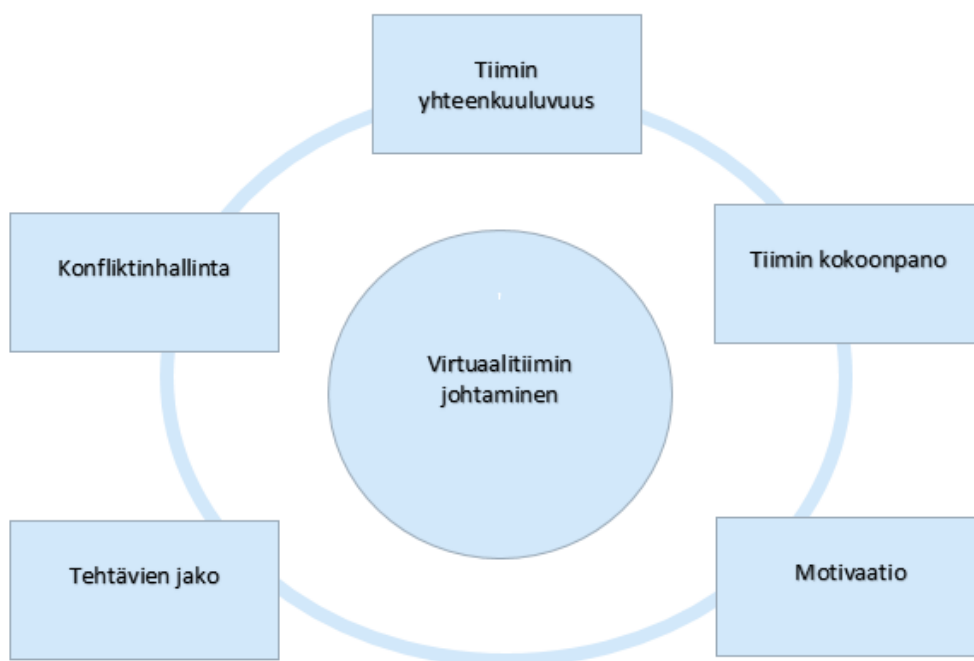
Kuvio 7. Virtuaalitiimi (Conchúir ja muut, 2009b).



Kuvio 8. Löyhästi kytketty tiimi (Conchúir ja muut, 2009b).

### 4.3 Virtuaalitiimin johtaminen

Virtuaalitiimin jäsenet tulevat erilaisista taustoista. Erilaiset taustat saattava aiheuttaa yhteentörmäyksiä tiimin sisällä. Jain ja Suman (2018) nimeävät ongelmia lisäävinä tekijöinä muun muassa riittämättömän viestinnän, huonon motivaation, tietämättömyyden erilaisten kulttuurien eroista sekä yleisesti huonon johtamisen ja erityisesti huonon konfliktien hallinnan. Virtuaalitiimin johtamisen tärkeimmät osatekijät Jainin ja Sumanin (2018) mukaan on esitetty kuviossa 9.



**Kuvio 9. Virtuaalitiimin johtamisen tärkeimmät osatekijät (Jain & Suman, 2018).**

Projektitiimin hajauttaminen saattaa heikentää ryhmän yhteenkuuluvuuden tunnetta. Fyysisillä tapaamisilla on suuri merkitys tiimin yhteenkuuluvuudentunteen muodostumisessa sekä keskinäisen luottamuksen kasvussa. Hajautetuissa tiimeissä tilannetta voi parantaa kierrättämällä projektin henkilöitä eri toimipisteiden välillä ja järjestämällä

mahdollisuuksien mukaan fyysisiä tapaamisia tiimin jäsenten välille. (Dubé & Robey, 2008; Niazi ja muut, 2016a)

Tiimin kokoonpanoon vaikuttaviin tekijöihin on GSD-projekteissa kiinnitettävä erityistä huomiota. Tiimin jäsenten tulee olla niin teknisiltä kuin kielitaidoltaankin riittävällä tasolla ja heidän roolinsa sekä vastuunsa projektissa tulee olla selkeästi määritelty. Myös ihmissuhde- ja vuorovaikutustaidot ovat tärkeitä tiimin jäsenille. Tiimin työskentelylle on eduksi, mitä useammalla jäsenellä on kokemusta kyseisestä asiakasyhteistyöstä. Kokemus parantaa tuottavuutta, vähentämällä väärinkäsityksiä ja parantamalla luottamusta. Toimivassa ryhmässä myös jäsenten vaihtuvuuden pitää pysyä kohtuullisella tasolla. Suuri vaihtuvuus heikentää ryhmän toimintaa muun muassa sen takia, että poistuvat ryhmän jäsenet vievät väistämättä mukanaan työskentelyssä tarvittavaa hiljaista tietoa. (Jain & Suman, 2018)

Tiimin jäsenten motivaatiolla on suuri merkitys tiimin työn laatuun. Beecham ja muut (2007) mukaan huono motivaatio saattaa näkyä tiimin jäsenillä muun muassa korkeampana vaihtuvuutena ja suurempana poissaolojen määränä. Korkea motivaatio näkyy maltillisempana vaihtuvuutena ja suurempana tuottavuutena. Jain ja Suman (2018) luettelevat motivaatiota kasvattavina tekijöinä esimerkiksi palkkiot, onnistuneet suoritukset, vapauden ilmaista mielipiteitään, oikeudenmukaisen johtamisen sekä säännöllisen rakentavan palautteen ja kunnioituksen.

Tehtävien jakaminen tiimin jäsenille on yksi keskeinen asia projektin johtamisessa. Kun projektiryhmä työskentelee hajautetusti, tehtävien jakamiseen liittyy erityisiä vaatimuksia. Tehtäviä jaettaessa tulee ottaa huomioon useita seikkoja, kuten esimerkiksi eri toimipaikoissa tehdyn työn kustannusvaikutus, tehtävien väliset riippuvuussuhteet ja tuotteen arkkitehtuuri, aikavyöhykkeiden vaikutus työn organisointiin ja kulttuuriin liittyvät erot. Huomioivia seikkoja ovat vielä muun muassa saatavilla oleva tieto-taito, asiantuntijoiden kielitaito ja haluttujen asiantuntijoiden käytettävyys, etäisyys asiakkaaseen ja

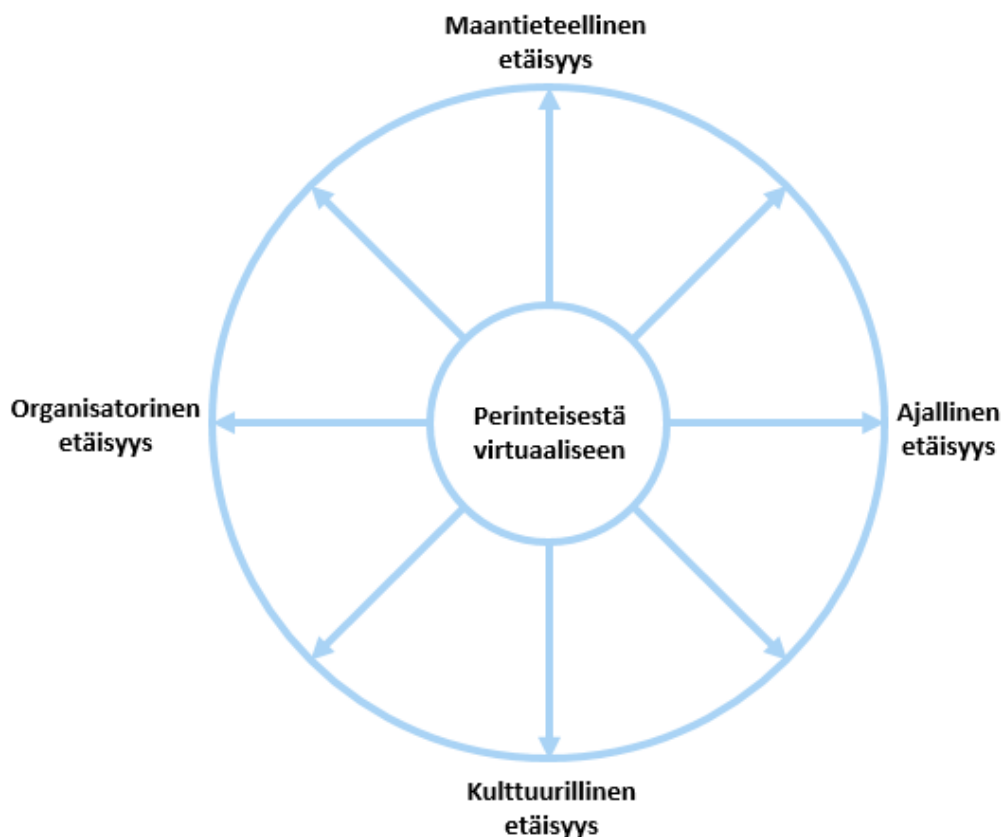
markkinoihin, poliittiset tekijät, kuten esimerkiksi paikalliset lait ja säädökset sekä henkilökohtaiset kontaktit. (Lamersdorf ja muut, 2009)

Yhtenä ratkaisevan tärkeänä osana virtuaalitiimin johtamisessa Jain ja Suman (2018) näkevät konfliktien hallinnan. Projektityöskentelyssä ja varsinkin virtuaalitiimin työssä, törmätään melkein väistämättä jonkinlaisiin konflikteihin. Konflikteja saattaa syntyä niin hajautetun tiimin sisällä kuin hajautetun tiimin ja jonkin sidosryhmän, esimerkiksi asiakkaan välillä. Nämä konfliktitilanteet on syytä saada käsiteltyä nopeasti ja niin, etteivät ne jää enää rasittamaan tiimin tulevaa toimintaa. Konfliktien hallinnassa projektipäälliköllä tai virtuaalitiimin vetäjällä on ratkaiseva rooli.

#### **4.4 Virtuaalisuuden aste ja etäisyys**

Virtuaalitiimejä on hyvin erilaisia. Ne voivat erota toisistaan esimerkiksi jäsenyysskriteerien, tavoitteiden tai kulttuurillisten ominaispiirteiden osalta. Myös tiimin virtuaalisuudella voi olla eri asteita. Zigurs (2003) esittää, että tiimi voi olla hajautettu monella eri tavalla. Yleisimmät virtuaalisuuden asteeseen vaikuttavat ulottuvuudet on esitetty kuviossa 10. Ne ovat:

- Maantieteellinen etäisyys
- Ajallinen etäisyys
- Kulttuurillinen etäisyys
- Organisatorinen etäisyys



**Kuvio 10. Virtuaalitiimin yleisimmät ulottuvuudet (Zigurs, 2003).**

Tiimin virtuaalisuuden aste on eri virtuaalisuuden ulottuvuuksien kokonaismäärä. Jokaisen ulottuvuuden kasvu lisää tiimin virtuaalisuuden kokonaismäärää. Mitä enemmän tiimin virtuaalisuus kasvaa, sitä monimutkaisemmiksi muodostuvat myös tarpeet, joita tiimi tulee työskennellessään kohtaamaan (Zigurs, 2003). Tiivistäen voisi todeta, että matalamman virtuaalisuuden asteen tiimin jäsenten on helpompi tavata toisensa henkilökohtaisesti, työskennellä samaan aikaan ja ymmärtää toistensa kulttuuria.

Maantieteellinen etäisyys tarkoittaa käytännössä tiimiläisten fyysistä etäisyyttä toisiinsa. Tässä etäisyydellä tarkoitetaan kuitenkin sitä, miten helppoa jäsenten on fyysisesti tavata toisiaan (Holmström ja muut, 2006). Lyhytkin fyysinen etäisyys voi tässä tarkoittaa suurta etäisyyttä, jos välissä on este, jota ei voida ylittää. Esimerkiksi toimipisteiden välillä ei ole sopivia kulkuyhteyksiä tai esimerkiksi valtakunnan raja on suljettu. Toisaalta

myös pitkä matka voidaan katsoa lyhyeksi etäisyydeksi, jos se on vaivaton kulkea esimerkiksi hyvien juna- tai lentoyhteyksien ansiosta. Maantieteellinen etäisyys johtaa siihen, että henkilökohtaiset tapaamiset vähenevät, joka puolestaan johtaa vähäisempään luonnolliseen ja epäviralliseen kommunikaatioon (Bano ja muut, 2016). Kommunikaation vähentymiseen liittyvänä vaarana on, että se saattaa heikentää luottamusta ja yhteenkuuluvuuden tunnetta (Daim ja muut, 2012). Maantieteellisestä etäisyydestä aiheutuneita ongelmia voi pyrkiä vähentämään lisäämällä viestintää erilaisten kommunikaatiovälineiden avulla sekä järjestämällä tiimin jäsenille fyysisiä tapaamisia.

Khan ja muut (2014) mukaan kulttuurinen etäisyys kuvaa sitä, miten paljon henkilöt eroavat toisistaan kielen, uskonnon, sosiaalisen tai taloudellisen aseman, poliittisten näkemysten tai muiden vastaavien tekijöiden perusteella. Kaikki edellä mainitut asiat vaikuttavat tiimin jäsenten käytökseen ja asenteisiin sekä voivat hankaloittaa heidän keskinäistä yhteisymmärrystään ja heikentää luottamusta. Yhdeksi tiimiläisten aivan keskeiseksi eroiksi voi luonnollisesti muodostua kielitaito (Jain & Suman, 2018). Kulttuurillisen etäisyyden tuomia vaikeuksia voi lievittää esimerkiksi kielitaitoon ja erilaisiin kulttuureihin liittyvillä koulutuksilla.

Ajallinen etäisyys tarkoittaa sitä yhteistä työaikaa, joka tiimiläisillä on käytössään, jos he työskentelevät eri aikavyöhykkeillä (Ågerfalk ja muut, 2008). Ajallinen etäisyys on osittain seurausta maantieteellisestä etäisyydestä ja työskentelystä eri aikavyöhykkeillä. Ajallinen etäisyys voi olla hajautetussa työskentelyssä myös positiivinen asia, sillä se mahdollistaa ympärivuorokautisen työskentelyn ”follow-the-sun”-periaatteen mukaisesti. Ajallisesta etäisyydestä aiheutuu esimerkiksi viivettä kommunikaatioon. Ajallisen etäisyyden haitallisia vaikutuksia voi lievittää muun muassa käyttämällä asynkronista kommunikointia, säätämällä työaikoja ja järjestelmällä yhteiset palaverit niinä aikoina, jolloin kaikki voivat olla töissä yhtä aikaa.

Organisatorisella etäisyydellä tarkoitetaan tilannetta, jossa yhteistyötä tekevät henkilöt kuuluvat eri organisaatioihin. Tällöin heillä saattaa olla käytössä erilaisia menetelmiä sekä työtapoja ja he ovat mahdollisesti omaksuneet erilaiset organisaatiokulttuurit. Tämä voi tehdä organisaatioiden välisistä rajoista hankalia ylittää, jolloin esimerkiksi tiimien muodostaminen eri organisaatioiden välille hankaloituu ja tiedon siirto organisaatioiden välillä heikkenee. Suuri organisatorinen etäisyys voi olla tietyissä tilanteissa taroituksellinenkin asia. Voidaan esimerkiksi haluta, ettei tietoa vapaasti välitetä oman organisaation ulkopuolelle. Organisatorisen etäisyyden aiheuttamia negatiivisia vaikutuksia voidaan vaimentaa luomalla luottamusta eri osapuolten välille ja lisäämällä avoimuutta. Kun eri osapuolilla on riittävän hyvä käsitys toisten organisaatioiden rakenteesta, saavutetaan selkeä kuva siitä, kehen voi toisessa organisaatiossa voi olla yhteydessä mihinkin asiaan liittyen. (Jaanu ja muut, 2012).

## 5 Johtopäätökset

Tietotekniikan kehitys ja halpeneminen sekä tietoliikenneyhteyksien nopeutuminen ja luotettavuuden parantuminen ovat mahdollistaneet ohjelmistokehityksen hajautumisen ja samalla kansainvälistymisen. Tässä ohjelmistokehitys on seurannut yleisesti vallalla olutta globalisaation trendiä. GSD on kasvanut ohjelmistoalan merkittäväksi suuntaukseksi kaikkialla maailmassa. Ohjelmistokehityksen kansainvälistymisellä on saavutettu merkittäviä hyötyjä. Havaittavissa on kuitenkin selvästi se, että liian monet kansainvälisesti toteutetut ohjelmistoprojektit jäävät tavoitteistaan. Tämä kertoo vahvasti siitä, ettei kaikkia kansainvälisen ohjelmistokehityksen vaateita osata ennakoida. Tässä työssä kävin läpi kansainvälistä sovelluskehitystä projektinhallinnan ja hajautettujen tiimien näkökulmasta sekä niihin liittyviä ongelmia ja myös ongelmiin varautumista. Työssä tunnistetut vaikeudet myötäilevät hyvin keskeisiä projektinhallinnan yleisiä osa-alueita. Kansainvälisyys ja muun muassa projektin osallistujien maantieteellinen, ajallinen, kulttuurillinen sekä organisatorinen etäisyys toisistaan tuovat vaikeuksiin vielä omat erityispiirteensä sekä monimutkaistavat projektinhallintaa. Odotetusti hajautettujen tiimien keskeiset hankaluudet liittyivät erilaisten kulttuurien kohtaamiseen, tiedonvälitykseen ja kommunikointiin.

Tässä työssä esitettyjen uhkien tunnistaminen ja niihin vastaaminen on hyödyllistä hajautettujen projektien parissa työskenteleville. Ongelmatilanteisiin oikein varautumalla, saadaan projektien onnistumistodennäköisyyttä nostettua ja mahdollisia piilokustannuksia karsittua.

Työssä nousi esiin kriittisiä tekijöitä projektin ja tiimin työskentelyn varmistamiseksi. Tällaisia asioita ovat esimerkiksi huolenpito siitä, että koko projektiryhmä jakaa saman käsityksen siitä, mitä ollaan tekemässä. Myös roolituksen ja vastuunajon on oltava selkeitä. Projektiryhmän jäsenten välille on pyrittävä rakentamaan vahva luottamuksen side. Kansainvälinen konteksti tuo mukaan vielä erilaisiin kulttuureihin ja kommunikaatioon liittyvät hankaluudet. Tiimin osalta on ensiarvoisen tärkeää varmistaa, että jäsenillä on yhteinen kieli ja että kaikki puhuvat sitä riittävän hyvin. Pelkkä kielen tekninen puhuminen ei



silti välttämättä riitä, vaan tiimin jäsenten välillä on oltava yhteinen käsitys kielen semantiikasta. Parhaan mahdollisen kommunikaation varmistamiseksi tulee kiinnittää huomiota ryhmän käytössä olevien kommunikaatiotyökalujen ja –ohjelmistojen tarkoituksenmukaisuuteen.

Hajautetussa ohjelmistokehityksessä saavutettujen hyötyjen ja piilevien kustannusten vuoksi hajautetun ja keskitetyn työskentelyn kustannustehokkuuden vertailu on vaikeaa. On yksinkertaista verrata pelkästään hajauttamisella tai ulkoistamisella saatua halvempaa työn tuntihintaa. On kuitenkin selviä viitteitä siitä, että hajauttaminen tuo mukanaan muun muassa tehottomuutta, kommunikaation heikentymistä sekä työn hidastumista. Tämä kaikki vaikeuttaa kokonaishyödyn laskemista ja saattaa jopa tehdä siitä mahdotonta.

## **5.1 Tutkimuksen luotettavuus**

Tutkimuksen luotettavuutta mietittäessä mieleen tulee ensimmäisenä se, miten toistettava se on. Eli millä todennäköisyydellä toinen henkilö päätyisi samalla metodilla, samasta aiheesta, vastaavaan lopputulemaan. Tämä työ on opinnäytetyö, joten se asettaa selvät rajat työhön käytetylle ajalle. Se asettaa myös jossain määrin rajat sille mihin aineistoihin on mahdollista päästä käsiksi.

Aiheiston hakuun käytetyt hakuehdot ja käytössä olleet hakutietokannat saattavat tuoda vaihtelua hakuvaiheen tuloksena saatavaan aineistoon. Samasta aiheesta olisi saattanut hieman erilaisilla hakuehdoilla päätyä osittain erilaiseen aineistoon ja saada siten alkuvaiheessa käyttöönsä erilaisen lähdeaineiston. En kuitenkaan usko, että ero voisi olla merkittävä.

## 5.2 Jatkotutkimuksen aiheita

Tässä työssä viitatuissa lähteissä on aihetta käsitelty yleisellä tasolla, ja niissä sivutut esimerkit ovat pääsääntöisesti Euraasiasta ja Pohjois-Amerikasta. Mielestäni on kuitenkin viitteitä siitä, että hajautus ulottuu jatkossa myös yhä enemmän tämän alueen ulkopuolelle. Olisi kiinnostavaa tietää tuovatko esimerkiksi Afrikka tai Etelä-Amerikka jotain omia lisäpiirteitään projektityöskentelyyn tai hajautettujen tiimien toimintaan.

Jainin ja Sumanin (2015) mukaan ohjelmiston toteutus ja testaus ovat helpoimpia ulkoistaa ja sen takia niitä hyödynnetään eniten GSD:ssä. Jatkotutkimuksen aiheena voisi olla myös se, miten muiden ohjelmistokehitysvaiheiden, kuten määrittelyn tai suunnittelun, osuutta GSD:ssä voisi lisätä.

Monet organisaatiot osallistuvat globaaliin ohjelmistokehitykseen niin, ettei niillä ole etukäteen käsitystä oman projektijohtamisensa kyvystä selviytyä kansainvälisen toiminnan vaatimuksista (Khan ja muut, 2011). Olisi hyödyllistä tietää, millä seikoilla voi varmistaa, että organisaatio on valmis globaaliin ohjelmistokehitykseen. Voidaanko määritellä mittareita, joista asian näkisi objektiivisesti?

## Lähteet

- Agile Alliance (2020). *Agile 101*. Noudettu 20.3.2020 osoitteesta <https://www.agilealliance.org/agile101/>
- Alzoubi, Y.I., Gill, A. Q. & Al-Ani, A. (2016). *Empirical studies of geographically distributed agile development communication challenges: A systematic review*. Information & Management Volume 53, Issue 1, January 2016, Pages 22-37. <https://doi.org/10.1016/j.im.2015.08.003>
- Anantatmula, V. & Thomas, M. (2010). *Managing Global Projects: A Structured Approach for Better Performance*. Project Management Journal. <https://doi.org/10.1002/pmj.20168>
- Anwar, R., Rehman, M., Wang, K. S. & Hashmani, M. A. (2019). *Systematic Literature Review of Knowledge Sharing Barriers and Facilitators in Global Software Development Organizations Using Concept Maps*. Project Management Journal. <https://doi.org/10.1109/ACCESS.2019.2895690>
- Anwer, S., Wen, L., Wang, Z. & Mahmood, S. (2019). *Comparative Analysis of Requirement Change Management Challenges Between in-House and Global Software Development: Findings of Literature and Industry Survey*. IEEE Access, vol. 7, pp. 116585-116611. <https://doi.org/10.1109/ACCESS.2019.2936664>
- Avdoshin, S. M. & Pesotskaya, E. Y. (2011). *Software risk management*. 2011 7th Central and Eastern European Software Engineering Conference (CEE-SECR), Moscow, 2011, pp. 1-6. <https://doi.org/10.1109/CEE-SECR.2011.6188471>
- Bannerman, P. L., Hossain, E. & Jeffery, R. (2012). *Scrum Practice Mitigation of Global Software Development Coordination Challenges: A Distinctive Advantage?* 45th Hawaii International Conference on System Sciences, Maui, HI, 2012, pp. 5309-5318. <https://doi.org/10.1109/HICSS.2012.512>
- Bano, M., Zowghi, D. & Sarkissian, N. (2016). *Empirical study of communication structures and barriers in geographically distributed teams*. IET Software, vol. 10, no. 5, pp. 147-153. <https://doi.org/10.1049/iet-sen.2015.0112>

- Beecham, S., Baddoo, N., Hall, T., Robinson H. & Sharp, H. (2007). *Motivation in Software Engineering: A systematic literature review*. Information and Software Technology 50(9–10), 860-878. <https://doi.org/10.1016/j.infsof.2007.09.004>
- Binder, J. (2007). *Global Project Management: Communication, Collaboration and Management Across Borders*. Gower Publishing Limited. ISBN: 9780566087066
- Boehm, B. (1989). *Software risk management*. In: Ghezzi C., McDermid J.A. (eds) ESEC '89. ESEC 1989. Lecture Notes in Computer Science, vol 387. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/3-540-51635-2\\_29](https://doi.org/10.1007/3-540-51635-2_29)
- Casey, V. (2010). *Virtual software team project management*. J Braz Comput Soc 16, 83–96. <https://doi.org/10.1007/s13173-010-0013-3>
- Casey, V. & Richardson, I. (2006). *Uncovering the reality within virtual software teams*. GSD '06: Proceedings of the 2006 international workshop on Global software development for the practitioner. Pages 66–72. <https://doi.org/10.1145/1138506.1138523>
- Chadli, S. Y., Idri, A., Fernández-Alemán, J. L. & Ros, J. N. (2015). *Frameworks for risk management in GSD projects: A survey*. 2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA), Rabat, 2015, pp. 1-6. <https://doi.org/10.1109/SITA.2015.7358381>
- Conchúir, E. Ó., Olsson, H. H., Ågerfalk, P. J. & Fitzgerald, B. (2009a). *Benefits of Global Software Development: Exploring the Unexplored*. Softw. Process Improve. Pract. 2009; 14: 201–212. <https://doi.org/10.1002/spip.417>
- Conchúir, E. Ó., Ågerfalk, P. J., Olsson, H. H. & Fitzgerald, B. (2009b). *Global software development: where are the benefits?* Communications of the ACM. <https://doi.org/10.1145/1536616.1536648>
- Cusumano, M. (2008). *Managing software development in globally distributed teams*. Communications of the ACM. <https://doi.org/10.1145/1314215.1314218>
- Daim, T. U., Ha, A., Reutiman, S., Hughes, B., Pathak, U., Bynum, W. & Bhatla, A. (2012). *Exploring the communication breakdown in global virtual teams*. International Journal of Project Management Volume 30, Issue 2, February 2012, Pages 199-212. <https://doi.org/10.1016/j.ijproman.2011.06.004>

- Denhere, N., Hörne, T. & van der Poll, J. A. (2015). *Managing Globally Distributed Software Development Projects using Virtual Teams: A Middle East Case Study*. SA-ICSIT '15: Proceedings of the 2015 Annual Research Conference on South African Institute of Computer Scientists and Information Technologists. <https://doi.org/10.1145/2815782.2815786>
- Dingsøyr, T. & Smite, D. (2014). *Managing Knowledge in Global Software Development Projects*. IT Professional, vol. 16, no. 1, pp. 22-29. <https://doi.org/10.1109/MITP.2013.19>
- Dubé, L. & Robey, D. (2008). *Surviving the paradoxes of virtual teamwork*. Journal compilation © 2008 Blackwell Publishing Ltd. <https://doi.org/10.1111/j.1365-2575.2008.00313.x>
- Earl, M. (2001). *Knowledge Management Strategies: Toward a Taxonomy*. Journal of Management Information Systems, 18:1, 215-233. <https://doi.org/10.1080/07421222.2001.11045670>
- Ebert, C. (2011). *Global Software and IT: A Guide to distributed development, projects, and outsourcing*. The Institute of Electrical and Electronics Engineers, Inc. ISBN: 9780470636190
- Ebert, C. & De Neve, P. (2001). *Surviving global software development*. IEEE Software 18(2), 62–69. <https://doi.org/10.1109/52.914748>
- Ebert, C., Kuhrmann, M. & Prikladnicki, R. (2016). *Global Software Engineering: Evolution and Trends*. 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE), Irvi-ne, CA, 2016, pp. 144-153. <https://doi.org/10.1109/ICGSE.2016.19>
- Ebert, C. & Paasivaara, M. (2017). *Scaling Agile*. IEEE Software, vol. 34, no. 6, pp. 98-103, November/December 2017. <https://doi.org/10.1109/MS.2017.4121226>
- El Bajta, M., Idri, A., Ros, J. N., Fernández-Alemán, J. L., de Gea, J. M. C., García, F. & Toval, A. (2018). *Software project management approaches for global software development: a systematic mapping study*. Tsinghua Science and Technology 23(6), 690–714. <https://doi.org/10.26599/TST.2018.9010029>

- Espinosa, J.A. & Carmel, E. (2004). *The Effect of Time Separation on Coordination Costs in Global Software Teams: A Dyad Model*. 37th Annual Hawaii International Conference on System Sciences, 2004. <https://doi.org/10.1109/HICSS.2004.1265158>
- Farias, I., Azevedo, R. R. d., Moura H. P. d. & Silva, D. S. M. d. (2012). *Elicitation of Communication Inherent Risks in Distributed Software Development*. 2012 IEEE Seventh International Conference on Global Software Engineering Workshops, Porto Alegre, 2012, pp. 37-42. <https://doi.org/10.1109/ICGSEW.2012.18>
- Gopal, A., Mukhopadhyay, T. & Krishnan, M.S. (2002). *The role of software processes and communication in offshore software development*. Communications of the ACM. <https://doi.org/10.1145/505248.506008>
- Gumm, D. C. (2006). *Distribution Dimensions in Software Development Projects: A Taxonomy*. IEEE Software 23(5), 45–51. <https://doi.org/10.1109/MS.2006.122>
- Herbsleb, J. D. & Mockus, A. (2003). *An empirical study of speed and communication in globally distributed software development*. IEEE Transactions on Software Engineering, vol. 29, no. 6, pp. 481-494. <https://doi.org/10.1109/TSE.2003.1205177>
- Herbsleb, J. D. & Moitra, D. (2001). *Global software development*. IEEE Software, vol. 18, no. 2, pp. 16-20. <https://doi.org/10.1109/52.914732>
- Holmström, H., Conchuir, E. O., Ågerfalk, P. J. & Fitzgerald, B. (2006). *Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance*. IEEE International Conference on Global Software Engineering (ICGSE'06), Florianopolis, 2006, pp. 3-11. <https://doi.org/10.1109/ICGSE.2006.261210>
- Jaanu, T., Paasivaara, M. & Lassenius, C. (2012). *Effects of four distances on communication processes in global software projects*. Proceedings of the 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, Lund, 2012, pp. 231-234. <https://doi.org/10.1145/2372251.2372293>
- Jain, R. & Suman, U. (2015). *A Systematic Literature Review on Global Software Development Life Cycle*. ACM SIGSOFT Software Engineering Notes. <https://doi.org/10.1145/2735399.2735408>

- Jain, R. & Suman, U. (2018). *A Project Management Framework for Global Software Development*. ACM SIGSOFT Software Engineering Notes. <https://doi.org/10.1145/3178315.3178329>
- Jawadi, N. & Boukef-Charki, N. (2009). *An Exploratory Study Of The Effects Of Virtuality On Team Effectiveness*. MCIS 2009 Proceedings. Noudettu 17.2.2020 osoitteesta <https://pdfs.semanticscholar.org/af43/39ac9b86caf00c81bc0cdf4e1a0ea60a5d49.pdf>
- Jiménez, M., Piattini, M. & Vizcaíno, A. (2009). *Challenges and Improvements in Distributed Software Development: A Systematic Review*. Adv. Soft. Eng. 2009. <https://doi.org/10.1155/2009/710971>
- Kangasniemi, M., Utriainen, K., Ahonen, S-M., Pietilä, A-M., Jääskeläinen, P. & Liikanen, E. (2013). *Kuvaileva kirjallisuuskatsaus: Eteneminen tutkimuskysymyksestä jäsenettyyn tietoon*. Hoitotiede 25(4), 291–301. <https://urn.fi/URN:NBN:fi:ELE-1614408>
- Kaur, P. & Sharma, S. (2014). *Agile Software Development in Global Software Engineering*. International Journal of Computer Applications, 97(4), 39-43. <https://doi.org/10.5120/16999-7181>
- Khan, S.U., Niazi, M. & Ahmad, R. (2010). *Factors Influencing Clients in the Selection of Offshore Software Outsourcing Vendors: An Exploratory Study Using a Systematic Literature Review*. Journal of Systems and Software Volume 84, Issue 4, April 2011, Pages 686-699. <https://doi.org/10.1016/j.jss.2010.12.010>
- Khan, S.U., Niazi, M. & Ahmad, R. (2011). *Barriers in the selection of offshore software development outsourcing vendors: An exploratory study using a systematic literature review*. Information and Software Technology Volume 53, Issue 7, July 2011, Pages 693-706. <https://doi.org/10.1016/j.infsof.2010.08.003>
- Khan, A.A., Basri, S. & Dominc, P. D. D. (2014). *A Proposed Framework for Communication Risks During RCM in GSD*. Procedia - Social and Behavioral Sciences Volume 129, 15 May 2014, Pages 496-503. <https://doi.org/10.1016/j.sbspro.2014.03.706>
- Lamersdorf, A., Munch, J. & Rombach, D. (2009). *A Survey on the State of the Practice in Distributed Software Development: Criteria for Task Allocation*. 2009 Fourth IEEE

- International Conference on Global Software Engineering, Limerick, 2009, pp. 41-50. <https://doi.org/10.1109/ICGSE.2009.12>
- Layman, L., Williams, L., Damian, D. & Bures, H. (2006). *Essential communication practices for Extreme Programming in a global software development team*. Information and Software Technology Volume 48, Issue 9, September 2006, Pages 781-794. <https://doi.org/10.1016/j.infsof.2006.01.004>
- Layzell, P., Brereton, O. P. & French, A. (2000). *Supporting collaboration in distributed software engineering team*. Proceedings Seventh Asia-Pacific Software Engineering Conference. APSEC 2000, Singapore, 2000, pp. 38-45. <https://doi.org/10.1109/APSEC.2000.896681>
- Niazi, M., Mahmood, S., Alshayeb, M., Qureshi, A. M., Faisal, K. & Cerpa, N. (2016a). *Toward successful project management in global software development*. International Journal of Project Management Volume 34, Issue 8, November 2016, Pages 1553-1567. <https://doi.org/10.1016/j.ijproman.2016.08.008>
- Niazi, M., Mahmood, S., Alshayeb, M., Riaz, M., Faisal, K., Cerpa, N., Khan, S. & Richardson, I. (2016b). *Challenges of Project Management in Global Software Development: A Client-Vendor Analysis*. Information and Software Technology Volume 80, December 2016, Pages 1-19. <https://doi.org/10.1016/j.infsof.2016.08.002>
- Noll, J., Beecham, S., Richardson, I. & Nic Canna, C. (2016). *A Global Teaming Model for Global Software Development Governance: A Case Study*. 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE). <https://doi.org/10.1109/ICGSE.2016.33>
- Prikladnicki, R., Audy, J. L. N. & Evaristo, R. (2006). *A Reference Model for Global Software Development: Findings from a Case Study*. IEEE International Conference on Global Software Engineering (ICGSE'06), Florianopolis, 2006, pp. 18-28. <https://doi.org/10.1109/ICGSE.2006.261212>
- Prikladnicki, R. & Audy, J. L. N. (2010). *Process models in the practice of distributed software development: A systematic review of the literature*. Information and Software Technology Volume 52, Issue 8, August 2010, Pages 779-791. <https://doi.org/10.1016/j.infsof.2010.03.009>



- Project Management Institute. (2013). *A guide to the project management body of knowledge (PMBOK guide)*. 5<sup>th</sup> edition. Pennsylvania: Project Management Institute. ISBN: 978-1-62825-184-5.
- Razzak, M. A. & Smite, D. (2015). *Knowledge Management in Globally Distributed Agile Projects—Lesson Learned*. 2015 IEEE 10th International Conference on Global Software Engineering, Ciudad Real, 2015, pp. 81-89. <https://doi.org/10.1109/ICGSE.2015.22>
- Robinson, P. T. (2019). *Communication Network in an Agile Distributed Software Development Team*. 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE), Montreal, QC, Canada, 2019, pp. 100-104. <https://doi.org/10.1109/ICGSE.2019.00015>
- Saleem, N. (2019). *Empirical Analysis of Critical Success Factors for Project Management in Global Software Development*. 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE), Montreal, QC, Canada, 2019, pp. 68-71. <https://doi.org/10.1109/ICGSE.2019.00025>
- Saleem, N., Mathrani, S. & Taskin, N. (2019). *Understanding the Different Levels of Challenges in Global Software Development*. 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE). <https://doi.org/10.1109/ICGSE.2019.00027>
- Sangwan, R., Bass, M., Mullick, N., Paulish, D. J. & Kazmeier, J. (2006). *Global Software Development Handbook*. Auerbach Publications. ISBN: 978-0849393846
- Sharma, S., Kaur, P. & Kaur, U. (2015). *Communication Understandability Enhancement in GSD*. International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), Noida, 2015, pp. 28-33. <https://doi.org/10.1109/ABLAZE.2015.7154962>
- Smith, P. G. (2001). *Communication holds global teams together*. 008 International Seminar on Future Information Machine Design. 7/26/2001, Vol. 73 Issue 14, p70-74. ISSN: 0024-9114
- Tao, Y. (2008). *A Study of Software Development Project Risk Management*. 008 International Seminar on Future Information Technology and Management Engineering,

- Leicestershire, United Kingdom, 2008, pp. 309-312.  
<https://doi.org/10.1109/FITME.2008.125>
- ul Haq, S., Raza, M., Zia, A. & Khan, M. N. A. (2011). *Issues in Global Software Development: A Critical Review*. J. Software Engineering & Applications, 2011, 4, 590-595.  
<https://doi.org/10.4236/jsea.2011.410069>
- Vallon, R., da Silva Estácio, F. W B, Prikladnicki, R. & Grechenig, T. (2018). *Systematic literature review on agile practices in global software development*. Information and Software Technology 96 (2018) 161–180.  
<https://doi.org/10.1016/j.infsof.2017.12.004>
- Zigurs, I. (2003). *Leadership in Virtual Teams: Oxymoron or Opportunity?* Organizational Dynamics, 31(4), 339–351. [https://doi.org/10.1016/S0090-2616\(02\)00132-8](https://doi.org/10.1016/S0090-2616(02)00132-8)
- Ågerfalk, P. J., Fitzgerald, B., Holmström, Olsson, H. & Ó Conchúir, E. (2008). *Benefits of Global Software Development: The Known and Unknown*. In: Wang Q., Pfahl D., Raffo D.M. (eds) Making Globally Distributed Software Development a Success Story. ICSP 2008. Lecture Notes in Computer Science, vol 5007. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-540-79588-9\\_1](https://doi.org/10.1007/978-3-540-79588-9_1)