

Laura Kinnunen

Petri Niemi

**Loppukäyttäjien asennoituminen ohjelmiston
vaatimusmäärittelytyöhön**

Tietotekniikan
pro gradu -tutkielma
19. toukokuuta 2017



JYVÄSKYLÄN YLIOPISTO
TIETOTEKNIIKAN LAITOS

Kokkolan yliopistokeskus Chydenius

Tekijät: Laura Kinnunen ja Petri Niemi

Yhteystiedot: laura.kinnunen@pp2.inet.fi, petri.niemi@gmail.com

Puhelinnumero: +358 40 561 6200, +358 400 907 526

Ohjaaja: Risto T. Honkanen ja Joakim Klemets

Työn nimi: Loppukäyttäjien asennoituminen ohjelmiston vaatimusmäärittelytyöhön

Title in English: End users' attitude towards software requirements engineering process

Työ: Tietotekniikan pro gradu -tutkielma

Sivumäärä: 134+35

Tiivistelmä: Tämän pro gradu -tutkielman tavoitteena oli selvittää loppukäyttäjien suhtautumista ohjelmistojen vaatimusmäärittelytyöhön Lapin ammattikorkeakoulussa. Tutkielmaa ohjasi tutkimuskysymykset: "Miten loppukäyttäjät asennoituvat ohjelmistojen vaatimusmäärittelytyöhön?" ja "Kuinka loppukäyttäjien osallistamista vaatimusmäärittelytyöhön voitaisiin lisätä?". Tutkimus oli Lapin ammattikorkeakouluun päätoimiseen henkilöstöön kohdistuva tapaustutkimus, jonka aineistonhankinnassa hyödynnettiin sekä määrällistä että laadullista menetelmää. Tutkimuksen ensimmäisessä vaiheessa suoritettiin sähköinen lomakekysely, jonka jälkeen järjestettiin kaksi ryhmähaastattelua. Tulosten mukaan henkilöstö haluaisi osallistua vaatimusmäärittelytyöhön enemmän, kuin mitä tutkimushetkellä koettiin mahdolliseksi. Asenne vaatimusmäärittelytyötä kohtaan Lapin ammattikorkeakoulussa on pääasiassa myönteinen, vaikka nykyhetken tilanteeseen suhtaudutaan osittain negatiivisesti.

Avainsanat: Loppukäyttäjät, motivaatio, ohjelmistokehitys, vaatimusmäärittelyt

Abstract: The purpose of this thesis was to examine end users' attitude towards the software requirements engineering process. The research problems were: "What is the end users' attitude towards the software requirements engineering process?" and "How can end users' be more involved in the requirements engineering process?". The approach of the thesis was a case study at the Lapland University of Applied Sciences. Both quantitative and qualitative methods were used to collect data. Initially, a quantitative survey study was conducted. Afterwards, through adopting a qualitative approach group interviews were held with chosen members of Lapland University of Applied Sciences. The results demonstrate that end users' at the Lapland University of Applied Sciences are motivated to participate in software requirements engineering process more than they are capable at the moment. Attitude

towards software requirements engineering is generally mostly positive, although end users' are not totally satisfied by the current situation of software acquisition in their organisation.

Keywords: End users, motivation, requirements engineering, software development

Copyright © 2017 Laura Kinnunen ja Petri Niemi

All rights reserved.

Esipuhe

Haluamme antaa kiitoksen pitkäjänteisestä ohjauksesta ja luottamuksesta työn ohjaajille Risto Honkaselle ja Joakim Klemetsille. Kiitoksen kannustuksesta ja tuesta ansaitsevat myös Anneli Heimbürger ja Mikko Myllymäki. Muistamme lämpimin mielin Kokkolan yliopistokeskus Chydeniuksen asiantuntevaa henkilökuntaa kaikesta avusta, jota olemme vuosien varrella opinnoissamme saaneet.

Opintojen suorittaminen työn ohessa on ollut hieno mahdollisuus, mutta se on myös vaatinut paljon. Kiitos ymmärryksestä ja kannustuksesta sukulaisille ja ystäville, olette olleet ajatuksissamme.

Kiitämme Lapin ammattikorkeakoulua tutkimuksen mahdollistamisesta ja tietotekniikan maisteriopintojen arvostuksesta. Erityinen kiitos kaikille tutkimukseen osallistuneille Lapin AMKilaisille, aktiivisuutenne mahdollisti tutkimuksen onnistumisen. Kiitos kuuluu myös kaikille tentinvalvojille lähipiirissämme.

Rovaniemellä, 19. päivänä toukokuuta 2017

Laura Kinnunen ja Petri Niemi

Sanasto

Agile	Ketterä kehittäminen
Agile manifesto	Ketterän kehityksen perusmääritelmä
CHAOS	Standish Groupin vuosittain julkaisema raportti
Fischerin tarkka testi	Testillä tutkitaan kahden muuttujan välistä riippuvuutta
Frekvenssi	Kuvaa tilastotieteessä kuinka monta kertaa arvo esiintyy
IEEE	Institute of Electrical and Electronics Engineers
IT	Information Technology
Khiin neliö (χ^2)	Testimuuttuja, joka kuvaa havaitun ja hypoteettisen taulukon välistä eroa
Khiin neliö -testi	Testillä tutkitaan kahden muuttujan välistä riippuvuutta
Lapin AMK	Lapin ammattikorkeakoulu Oy
R-kieli	Tilastotieteellistä ohjelmointia varten kehitetty ohjelmointikieli
Scrum	Ketterän kehityksen menetelmä, jossa työskennellään sykleittäin ennustettavuuden optimoimiseksi ja riskien kontrolloimiseksi
S.M.A.R.T	Specific, Measurable, Agreed upon, Realistic and Time based
TKI	Tutkimus-, kehitys- ja innovaatiotoiminta
URL	A Uniform Resource Locator
Vapausaste (df)	Niiden muuttujien lukumäärä, jotka voivat vaihdella. Kuvaa ristiintaulukoinnissa tutkittavan taulukon kokoa
XP	eXtreme programming

Sisältö

Esipuhe	i
Sanasto	ii
1 Johdanto	1
2 Vaatimusmäärittelytyö ohjelmistoprojekteissa	3
2.1 Taustaa vaatimusmäärittelytyöstä	3
2.2 Vaatimusten kehittäminen	6
2.2.1 Kartoittaminen	7
2.2.2 Analysointi	14
2.2.3 Neuvottelu	15
2.2.4 Dokumentointi	17
2.2.5 Validointi	19
2.3 Vaatimusten hallinta	20
2.3.1 Muutosten hallinta	21
2.3.2 Version hallinta	22
2.3.3 Tilan seuranta ja jäljitettävyys	22
2.4 Perinteiset prosessimallit	23
2.4.1 Vesiputousmalli	24
2.4.2 V-malli	25
2.4.3 Spiraalimalli	27
2.4.4 Prototyypimalli	28
2.5 Ketterät menetelmät	30
2.5.1 Taustaa ketteristä menetelmistä	30
2.5.2 XP - eXtreme Programming	34
2.5.3 Scrum	37
2.6 Perinteisten prosessimallien ja ketterien menetelmien vertailu	40
3 Syitä ohjelmistoprojektien epäonnistumiseen	43
3.1 Tutkimuksia ohjelmistoprojektien epäonnistumisista	44

3.2	Yleisimmät loppukäyttäjään liittyvät epäonnistumisen syyt	51
3.2.1	Toimintaympäristön tuntemus on puutteellista	51
3.2.2	Yhteinen maali ja muut maalin ongelmat	54
3.2.3	Loppukäyttäjä ei ole mukana projektissa	57
3.2.4	Vaatimukset eivät ole pysyviä	63
3.3	Yhteenvedo	67
4	Loppukäyttäjän suhtautuminen vaatimusmäärittelytyöhön	68
4.1	Tutkimuksia loppukäyttäjän osallistumisesta	68
4.2	Palautteen saaminen loppukäyttäjältä	72
4.3	Loppukäyttäjän saavutettavuus	73
4.4	Loppukäyttäjän motivaatio	75
4.5	Loppukäyttäjän osallistumisen ylläpito	78
4.6	Kehittäjän motivointi	81
4.7	Käyttäjän osallistumisen negatiivinen vaikutus	82
4.8	Yhteenvedo	82
5	Tutkimus- ja analysointimenetelmät	83
5.1	Tutkimusstrategia	83
5.2	Aineiston hankintamenetelmät	84
5.3	Aineiston analyysimenetelmät	89
5.4	Tutkimuksen luotettavuus ja toistettavuus	93
5.5	Tutkimuksen toteutus	93
5.5.1	Aineistonhankintamenetelmät	95
5.5.2	Aineiston analysointi	101
6	Tutkimuksen tulokset	105
6.1	Asennoituminen vaatimusmäärittelytyötä kohtaan	105
6.2	Vaatimusmäärittelytyöhön liittyvät haasteet ja mahdollisuudet . . .	107
6.2.1	Osallistaminen ja osallistuminen	107
6.2.2	Aikaresurssin puute	108
6.2.3	Ohjelmistojen käytettävyyteen vaikuttaminen	109
6.3	Vastaajien kehitysehdotukset	110
6.3.1	Loppukäyttäjien osallistaminen vaatimusmäärittelytyöhön . .	110
6.3.2	Aikaresurssin merkitsevyys ohjelmistohankinnoissa	112
6.3.3	Ohjelmistohankintojen suunnittelu	113

7 Tulosten pohdinta	115
7.1 Asennoituminen vaatimusmäärittelytyötä kohtaan	115
7.2 Kehitysehdotukset vaatimusmäärittelytyöhön osallistamiseen	116
7.3 Tulokset ja aikaisempi tutkimus	119
7.4 Tutkimuksen luotettavuus ja toistettavuus	120
8 Yhteenveto ja loppupäätelmät	123
Lähteet	126
Liitteet	
A Sähköinen lomakekysely	
B Ryhmähaastattelut	
C Sähköisen lomakekyselyn tuloksia ja vastausten ristiintaulukointeja	

1 Johdanto

Ohjelmistoprojektin onnistumisen edellytyksiin liittyy ohjelmiston loppukäyttäjiltä kerättävät vaatimukset. Ohjelmiston loppukäyttäjät ryhmitellään sidosryhmiksi, joiden edustajilta vaatimukset eli tarpeet kehitettävää ohjelmistoa varten kerätään. Vaatimusmäärittelytyön tavoitteena on kartoittaa kaikki ne tarpeet, jotka lisäävät loppukäyttäjän tyytyväisyyttä ja sitoutumista ohjelmistoa kohtaan. Loppukäyttäjän motivaatio vaatimusmäärittelytyöhön osallistumiseen ei kuitenkaan aina ole itsestäänselvyys. Asennoitumiseen vaatimusmäärittelytyötä kohtaan voivat vaikuttaa erinäiset tekijät, kuten ajanpuute, tietämättömyys vaatimusmäärittelytyön sisällöstä, muutosvastarinta tai oman osaamisen aliarviointi.

Valitsimme tutkimuksen aiheeksi selvittää loppukäyttäjien suhtautumista vaatimusmäärittelytyöhön. Tutkielmaa ohjasi siten tutkimuskysymykset: ”Miten loppukäyttäjät asennoituvat ohjelmistojen vaatimusmäärittelytyöhön?” ja ”Kuinka loppukäyttäjien osallistamista vaatimusmäärittelytyöhön voitaisiin lisätä?”. Vaatimusmäärittelytyöllä tarkoitetaan tässä pro gradu -tutkielmassa ohjelmistohankinnan vaihetta, jonka aikana tulevat ohjelmiston loppukäyttäjät osallistuvat vaatimusten kartoittamiseen ja määrittelytyöhön.

Tutkimusstrategiana käytettiin Lapin ammattikorkeakoulun päätoimiseen henkilöstöön kohdistuvaa tapaustutkimusta. Tutkimuksen aineistonhankintamenetelminä hyödynnettiin määrällistä lomakekyselyä ja laadullista ryhmähaastattelua. Lomakekyselyn avulla pyrittiin saavuttamaan yleiskuva Lapin AMKin henkilöstön suhtautumisesta vaatimusmäärittelytyöhön. Ryhmähaastattelujen avulla syvennettiin ymmärrystä tutkittavasta ilmiöstä, eli selvitettiin syitä henkilöstön suhtautumiseen vaatimusmäärittelytyötä kohtaan.

Tutkimuksen tulokset osoittautuivat päinvastaisiksi olettamuksiimme ja taustateoriaan nähden. Ryhmähaastattelujen mukaan henkilöstö haluaisi osallistua vaatimusmäärittelytyöhön enemmän, jos heille tarjottaisiin siihen mahdollisuus. Lomakekyselyn tulosten mukaan vaatimusmäärittelytyöhön suhtaudutaan pääasiassa positiivisesti silloin, kun henkilöllä on aikaisempaa kokemusta vaatimusmäärittelytyöhön osallistumisesta tai tietoa vaatimusmäärittelytyön sisällöstä. Lomakekyselyn ja ryhmähaastattelujen tulosten perusteella asennetta voitaisiin parantaa edel-

leen perehdytysten avulla.

Vaatusmääritytyöhön osallistumisen esteeksi nähtiin aikaresurssin puuttuminen ja johdon tuen puute. Lisäksi negatiivisena asiana nykyhetkessä koettiin ohjelmistohankintojen suunnittelun puutteellisuus ja ohjelmistojen huono käytettävyys. Tutkimukseen osallistuneet kritisoivat sitä, että loppukäyttäjiä aletaan osallistaa vasta käyttöönottovaiheessa, kun uuden ohjelmiston perehdytykset järjestetään. Tutkimuksen tuloksissa yllättävää oli se, että yleisesti tutkimukseen osallistuneiden asenne vaatimusmääritytyötä kohtaan oli myönteinen, vaikka suhtautuminen nykyhetken tilanteeseen oli monella osa-alueella tyytymätön.

Luvussa 2 esitellään vaatimusmääritytyön vaiheet jaettuna vaatimusten kehittämiseen ja vaatimusten hallintaan. Luvussa esitellään myös yleisimmät perinteiset prosessimallit ja ketterät menetelmät. Luvussa 3 käsitellään kirjallisuuden ja aikaisempien tutkimusten avulla syitä ohjelmistoprojektien epäonnistumiseen. Luvussa 4 käydään läpi loppukäyttäjän roolia ohjelmistoprojektissa ja syitä miksi vaatimusmääritytyöhön osallistumista ei välttämättä koeta tärkeäksi. Luvussa 5 kuvataan työssä käytetyt tutkimus- ja analysointimenetelmät. Luvussa 6 kuvataan lomakekyselyn ja ryhmähaastattelujen tuottamat tutkimuksen tulokset. Luvussa 7 pohditaan tutkimuksen tuloksia ja tulosten luotettavuutta. Luvussa 8 esitetään yhteenveto ja loppupäätelmät.

2 Vaatimusmäärittelytyö ohjelmistoprojekteissa

Dorfmanin ja Thayerin [24, s. 41] mukaan vaatimusmäärittelytyö on ensimmäinen vaihe ohjelmiston hankinnassa. Vaatimusmäärittelytyö tarkoittaa kehitettävän ohjelmiston vaatimusten kartoittamista ja dokumentointia. Vaatimusmäärittelytyön aikana vaatimukset ja tarpeet ohjelmistoa kohtaan kerätään ohjelmiston tulevilta asiakailta ja loppukäyttäjiltä. Chemuturi [17, s. 10] korostaa vaatimusmäärittelytyön olevan prosessi, jonka aikana käyttäjien vaatimukset muokataan sellaiseen muotoon, joka mahdollistaa käyttäjien tarpeet täyttävän ohjelmiston kehittämisen. Vaatimusmäärittelytyö sisältää vaatimusten keräämisen, analysoinnin, kirjaamisen ja seurannan läpi ohjelmistokehityksen. Scharerin [72, s. 30] mukaan vaatimusmäärittelytyön laadukkaalla läpiviennillä voidaan välttää tehokkaasti epäonnistumiset ohjelmistoprojektissa.

Tässä työssä vaatimusmäärittelytyö on jaettu kahteen osaan, vaatimusten kehittämiseen ja vaatimusten hallintaan. Seuraavassa alaluvussa 2.1 käydään läpi vaatimusmäärittelytyön taustaa. Alaluvussa 2.2 esitellään vaatimusten kehittäminen ja alaluvussa 2.3 käydään läpi vaatimusten hallinta. Alaluvussa 2.4 tutustutaan perinteisiin prosessimalleihin. Alaluvussa 2.5 käydään läpi ketterien menetelmien maljeja. Alaluvussa 2.6 vertaillaan perinteistä prosessimallia ja ketteriä menetelmiä niiden eroavaisuuksien ja yhtäläisyyksien osalta.

2.1 Taustaa vaatimusmäärittelytyöstä

Vaatimusmäärittelytyö on systemaattinen lähestymistapa vaatimusten kartoittamiseen, analysointiin, tarkastamiseen, jakamiseen, seuraamiseen ja hallintaan. Selkeä yhteys vaatimusten hallinnan ja ohjelmistoprojektien onnistumisen välillä on havaittu jo 1970-luvulla [25, s. 455]. Ohjelmistokehityksen suurimmat ongelmat liittyvät vaatimusten keräämiseen ja asiakkaan muuttuvien vaatimusten hallintaan [80]. Vaatimuksista keskustelu ja yksimielisyyteen pyrkiminen on yksi vaatimusmäärittelytyön keskeisimmistä ominaisuuksista. Jotta hyväksyntä ja yksimielisyys vaatimuksista voidaan saavuttaa, täytyy loppukäyttäjiltä ensin kysyä, millaisia vaatimuksia heillä on. Ohjelmistohankinnoissa ei tulisi sortua ajattelemaan, että vaati-

mukset voitaisiin tietää ilman loppukäyttäjien kuulemista [55]. Yu [89] korostaa kysymään vaatimusmäärittelytyön menetelmästä riippumatta enemmän ”*miksi*” kuin ”*mitä*”. Vastausten saaminen ”*miksi*”-kysymyksiin edesauttaa onnistuneiden ohjelmistojen kehittämistä ja yhteensopivuutta muiden ohjelmistojen kanssa. Onnistuneiden ohjelmistoprojektien edellytyksenä on aina tehokas vaatimusmäärittelyprosessi [8].

Sommervillen [78] mukaan vaatimusmäärittelyprosessi valitaan sen mukaisesti, millaista ohjelmistoa ollaan kehittämässä huomioiden organisaation luonne ja sen koko. Isoissa organisaatioissa suurten ohjelmistojen kehityksessä hyödynnetään usein tarkasti ohjattua vaatimusmäärittelytyötä ja tarkkaan dokumentoituja vaatimuksia. Pienemmissä organisaatioissa ohjelmistojen hankinnassa useimmiten riittää vaatimusmäärittelyprosessi, joka sisältää esimerkiksi työpajataapaamisia, joiden pohjalta tuotetaan visio kehitettävästä ohjelmistosta.

Hofmannin ja Lehnerin [39] mukaan menestyvät projektiryhmät koostuvat asiantuntijoista, joilla on perusteellista tietoa sovellusalasta, IT:stä ja vaatimusmäärittelyprosessista sekä menetelmistä. Projektin onnistumista edesauttaa kokeneen projektipäällikön valitseminen ja projektiryhmän jäsenten koostaminen sopivasta yhdistelmästä osaamista. Tärkein jäsen on kuitenkin loppukäyttäjä, jonka osallistaminen on kriittistä koko vaatimusmäärittelyprosessin alusta loppuun saakka. Onnistuneissa ohjelmistoprojekteissa loppukäyttäjää osallistetaan ja vuorovaikutuksesta pidetään hyvin huolta. Goguen ja Linde [32] toteavat loppukäyttäjillä olevan sellaista tietoa ja kokemusta, joka edesauttaa suuresti ohjelmiston kehityksessä. Käyttäjiltä kerättävät vaatimukset voivat olla toiminnallisten vaatimusten lisäksi myös turvallisuuden, luotettavuuteen tai muokattavuuteen liittyviä vaatimuksia. Yhteistyöllä pyritään varmistamaan vaatimusten oikein tulkitseminen, käsittelemään muuttuvat vaatimukset ja välttämään mahdolliset tietokatkokset.

Vaatimukset ovat IEEE standardin 610.12-1990 [43] mukaan ehtoja tai toimintoja, jotka mahdollistavat ohjelmiston käytettävyyden jonkin ongelman ratkaisemiseksi tai tavoitteen saavuttamiseksi. Vaatimukset jaetaan toiminnallisiin ja ei-toiminnallisiin vaatimuksiin. Toiminnalliset vaatimukset ovat vaatimuksia toiminnoista, jotka ohjelmiston on kyettävä toteuttamaan. Berander ja Andrews [11, s. 84] ja Machado ja muut [56, ss. 58–59] mukaan toiminnalliset vaatimukset liittyvät johonkin tiettyyn ohjelmiston toimintoon, kun taas ei-toiminnalliset vaatimukset liittyvät useisiin toiminnallisuuksiin. Ei-toiminnalliset vaatimukset asettavat rajoitteita ohjelmiston suunnitteluun, koska ne tyypillisesti määrittävät kehityksen alkuvai-

heessa suunnittelun ja toteutuksen ratkaisuja. Machadon ja muiden [56, ss. 58–59] mukaan ei-toiminnalliset vaatimukset voidaan jakaa kolmeen luokkaan: suunnittelutavoitteisiin, suunnittelupäätöksiin ja suunnittelurajoituksiin. Suunnittelutavoitteet liittyvät ohjelmiston laadullisiin vaatimuksiin. Laadullisia vaatimuksia voivat siten olla esimerkiksi vaatimus ohjelmiston nopeasta toiminnasta, helposta käyttönotosta tai edullisesta toteutuksesta. Suunnittelupäätökset voivat liittyä esimerkiksi tulevan ohjelmiston liittämiseen kaupallisiin ohjelmistoihin tai organisaatiossa olemassa olevaan suurempaan ohjelmistoon. Tämänkaltaiset ei-toiminnalliset vaatimukset voivat vaikuttaa teknologiavalintoihin tai ohjelmiston toiminnallisuuksiin. Suunnittelurajoituksiin voi sisältyä vaatimukset suorituskyvystä, luotettavuudesta, ohjelmiston laajuudesta ja hinnasta. Ei-toiminnallisten vaatimusten tunnistaminen on haastavampaa kuin toiminnallisten vaatimusten, koska niiden vaikutusta ei voi havaita jossakin ohjelmiston osassa, vaan ne ovat läpileikkaavia [46, s. 130].

Aurumin ja Wohlinin [5, s. 4] mukaan vaatimukset on tärkeää kerätä kaikilta mahdollisilta sidosryhmiltä sen sijaan, että kerättäisiin vain loppukäyttäjien vaatimukset. Tavoitetilassa vaatimukset ovat erillään ohjelmiston suunnittelusta, kertoen mitä ohjelmiston pitäisi tehdä sen sijaan, että vaatimukset kertoisivat, miten tulee toimia. Rachevan ja muiden [67] mukaan vaatimusten merkitys liiketoiminnalle on myös huomioitava. Arviointi voidaan tehdä miettimällä tilannetta kahdesta eri näkökulmasta. Arviointi voidaan toteuttaa arvioimalla toteutetun vaatimuksen tuomaa hyötyä ohjelmistolle tai sidosryhmälle tai toteuttamatta jätetyn vaatimuksen tuomaa haittaa. Arviointia tulisi johtaa mahdollisimman neutraalisti, sillä käyttäjät voivat myös yrittää manipuloida vain itselleen tärkeitä vaatimuksia kehitykseen mukaan.

Sidosryhmiksi nimitetään loppukäyttäjiä, kehittäjiä ja organisaation ulkopuolisia käyttäjiä kuten asiakkaita. Sidosryhmiä ovat kaikki käyttäjät, jotka ovat jollakin tavalla vuorovaikutuksessa ohjelmiston kanssa [5, s. 6], [19, s. 11], [74], [79]. IEEE standardin 1220-2005 [41] mukaan sidosryhmä on osapuoli, jolla on oikeus jakaa tai vaatia ominaisuuksia, jotka täyttävät kyseisen osapuolen tarpeet ja odotukset. Sidosryhmä sisältää myös asiakkaan ja loppukäyttäjän roolit. Asiakas on organisaatio tai henkilö, joka vastaanottaa ohjelmiston. Loppukäyttäjä puolestaan on henkilö tai ryhmä, joka hyötyy ohjelmiston käytöstä. Hull [40, s. 7] kuvaa sidosryhmän kiinnostuksen ohjelmistoa kohtaan syntyvän ohjelmiston käyttämisestä, siitä hyötymisestä tai vastuusta. Suhde voi olla myös päinvastainen. Ohjelmiston voidaan kokea tuottavan haittaa kuten kustannuksia. Cotterellin ja Hughesin [19, s. 13] mukaan

on tärkeää, että ohjelmiston sidosryhmät tunnistetaan mahdollisimman aikaisessa vaiheessa yhteydenpidon ja vuorovaikutuksen varmistamiseksi.

Aurumin ja Wohlinin [5, s. 5] mukaan tyypillisiä vaatimusmäärittelytyön osia ovat vaatimusten kerääminen, tulkitseminen, analysointi, dokumentointi, validointi ja vaatimuksista neuvottelu. Vaatimusmäärittelytyön suorittamiseen on olemassa erilaisia prosessimalleja, jotka tässä työssä on jaettu perinteisiin prosessimalleihin ja ketteriin menetelmiin. Stephensin [81, ss. 267–270] mukaan perinteiset prosessimallit ovat ennustavia, ne etenevät vaiheittain eteenpäin vaatimusten keräämisestä aina ohjelmiston käyttöönottoon. Perinteisessä prosessimallissa seuraavaan vaiheeseen siirrytään vasta, kun edellinen vaihe on saatu päätökseen. Ohjelmistoprojektissa perinteisen prosessimallin hyödyntäminen tarkoittaa, että ohjelmointia ei voida aloittaa ennen kuin vaatimusmäärittelytyö ja dokumentaatio on tuotettu valmiiksi. Ohjelmiston tavoitteet täytyy siten osaltaan ennustaa projektin alkuvaiheessa. Ketterät menetelmät antavat mahdollisuuden kehittää ja muuttaa vaatimuksia läpi projektin. Ketterät menetelmät etenevät sykleittäin, jotka sisältävät aina samat toimenpiteet vaatimusten priorisoinnista testaamiseen. Jokaisen syklin alussa vaatimukset priorisoidaan ja vaatimuksia on mahdollista lisätä sekä muuttaa.

Regnellin ja muiden [68] mukaan markkinavetoisia ohjelmistoja ja valmisohjelmistoja kehitettäessä vaatimukset eli tarpeet ja mahdollisuudet kerätään suoraan markkinoilta. Sidosryhmiä ovat tällaisissa tapauksissa markkinointiyritykset maailmanlaajuisesti. Markkinavetoisten ohjelmistojen vaatimusmäärittelytyö eroaa suuresti tässä työssä esitetyistä sopimus pohjaisista vaatimusmäärittelytyöistä, joissa kehittäjä on suorassa vuorovaikutuksessa asiakkaan kanssa. Tässä työssä ei käsitellä markkinavetoisten ohjelmistojen vaatimusmäärittelytyötä.

2.2 Vaatimusten kehittäminen

Sommervillen ja Sawyerin [80, s. 11] mukaan vaatimusten kehittämiseen sisältyy vaatimusten kartoittaminen, vaatimusten analysointi ja neuvottelu sekä vaatimusten validointi. Youngin [88, s. 120] mukaan loppukäyttäjien tarpeiden tunnistaminen ja täyttäminen voidaan ymmärtää kolmena vaiheena: sidosryhmien tarpeiden kartoittamisena, ohjelmiston vaatimusten muodostamisena, sekä vaatimusten analysointina ja validointina. Seuraavissa alaluvuissa käsitellään vaatimusten kehittämiseen liittyvät toimet.

2.2.1 Kartoittaminen

Hullin [40, s. 25] mukaan todellinen tarve ohjelmistolle pitää selvittää ennen kuin vaatimusten keräämistä tai projektia voidaan aloittaa. Jos ohjelmiston käyttötarkoitus ei ole selvä, on mahdotonta kehittää ohjelmistoa, joka täyttäisi käyttäjien vaatimukset. Nuseibehin ja Easterbrookin [62] mukaan ohjelmiston rajat määräytyvät samalla kun ongelma määritetään. Ohjelmiston rajojen tunnistaminen vaikuttaa myös sidosryhmien tunnistamiseen, tavoitteisiin ja tehtäviin sekä suunnitelmiin ja käyttötapauskuvauksiin. Zowghi ja Coulin [91, s. 21] korostavat tarpeen selvittämisen jälkeen vielä tilanteen tutkimista todellisessa maailmassa. Nykytilanteen tutkiminen antaa selkeämmän kuvan siitä, miten havaitun ongelman kanssa tullaan toimeen ennen ohjelmiston kehittämistä, ja mitä mahdollisia rajoitteita toimintaympäristöstä voi löytyä. Käytössä olevien toimintaprosessien tunteminen ja niihin liittyvät ongelmat täytyy olla tunnistettuna ja kuvattuna ennen vaatimusten keräämistä. Sommervillen ja Sawyerin [80, s. 81] mukaan kehitettävän ohjelmiston tulee tukea organisaation korkean tason tavoitteiden saavuttamista. Organisaation tavoitteet toimivat suunnannäyttäjinä vaatimusten kartoittamisessa ja auttavat sidosryhmiä päätöksenteossa vaatimuksista neuvoteltaessa.

Sidosryhmien tunnistaminen tulee tehdä ennen kuin vaatimuksia voidaan alkaa kartoittaa. Nuseibehin ja Easterbrookin [62] kuvailemana sidosryhmät ovat yksilöitä tai organisaatioita, jotka kokevat hyötyä tai haittaa ohjelmiston menestyksestä tai epäonnistumisesta. Sidosryhmiin kuuluvat asiakkaat, jotka vastaavat ohjelmiston kustannuksista, kehittäjät, jotka suunnittelevat, kehittävät ja ylläpitävät ohjelmistoa ja loppukäyttäjät, jotka ovat vuorovaikutuksessa ohjelmiston kanssa saadakseen tehtävänsä hoidettua. Loppukäyttäjien rooli korostuu etenkin vuorovaikutteisten ohjelmistojen suunnittelussa, koska vaatimukset käytettävyyteen voidaan kerätä vain loppukäyttäjiltä. Loppukäyttäjät sidosryhmänä sisältää ohjelmiston luonteesta riippuen erilaisia loppukäyttäjärooleja, kuten vähän kokeneita käyttäjiä, kokeneita käyttäjiä, satunnaisia käyttäjiä tai käyttäjiä, joilla on erityistarpeita.

Hull [40, ss. 97–98] esittää listan sidosryhmäkategorioiden, jonka läpikäymällä projektipäällikkö voi arvioida, onko vaatimusmäärittelytyöhön valittu edustajat kaikista sidosryhmistä.

- Päätäjät: Henkilöt, jotka päättävät projektin budjetista ja resursseista sekä arvioivat ohjelmiston soveltuvuutta organisaation arvoihin ja strategiaan.
- Sijoittajat: Henkilöitä tai organisaatioita, jotka rahoittavat ohjelmiston hankin-

taa.

- **Käyttäjät:** Ohjelmiston tulevat käyttäjät ovat tärkein sidosryhmä. Käyttäjiä ovat kaikki henkilöt, joilla on kiinnostus ohjelmistoa kohtaan ja jotka ovat suorassa tai epäsuorassa vuorovaikutuksessa ohjelmiston kanssa. Käyttäjiä voivat olla myös organisaation asiakkaat, joille ohjelmiston käyttöä tarjotaan. Olemassa olevan ohjelmiston käyttäjiltä voidaan kerätä paljon arvokasta tietoa nykyhetken ongelmista ja näkemyksiä tulevan ohjelmiston kehityksen.
- **Tukipalvelut:** Henkilöt, jotka vastaavat ohjelmiston ylläpidosta käyttöönoton jälkeen.
- **Kouluttajat:** Henkilöt, jotka ovat kiinnostuneita ohjelmiston käytettävyydestä ja perehdyttämisestä muille käyttäjille.
- **Toimintaympäristön asiantuntijat:** Henkilöt, jotka tuntevat olemassa olevan kokonaisarkkitehtuurin ja ymmärtävät, kuinka uusi ohjelmisto sijoittuu siihen ja mitä integraatioita on mahdollista toteuttaa.

Sidosryhmäkartoituksen jälkeen valitaan sidosryhmien edustajat. Valinnassa tulee huomioida, että edustajaksi valittu henkilö pystyy tuomaan mahdollisimman laajasti ja selkeästi oman sidosryhmänsä vaatimukset esiin. [40, s. 98]

Sidosryhmäedustajien lisäksi vaatimusten kartoittamisessa tärkeässä roolissa on henkilö, joka vastaa vaatimusmäärittelyprosessista. Zowghi ja Coulin [91, s. 24] kuvaavat tämän roolin tehtävien olevan riippuvainen projektin luonteesta. Vaatimusten kartoittajalla täytyy olla ymmärrys toimintaympäristöstä ja ongelmasta, jonka ratkaisemiseen ohjelmistoa ollaan kehittämässä. Vaatimusten kartoittaja ohjaa tilanteita, joissa sidosryhmien edustajat ovat läsnä, kuten ryhmäkeskusteluita. Haastattelutilanteissa pyritään ohjaamaan keskustelua hedelmälliseen suuntaan ja tekemään sidosryhmien edustajille myönteinen kuva heidän tarpeellisuudestaan sen sijaan, että keskityttäisiin vain kysymään vastauksia ja kirjaamaan vastauksia ylös. Vaatimusten kartoittaja toimii myös sovittelijan roolissa vaatimuksia priorisoitaessa, sidosryhmien välisissä erimielisyyksissä pyritään saavuttamaan yhteinen tahtotila tai kompromissi.

Sommervillen ja Sawyerin [80, ss. 38–42] mukaan vaatimusten kartoittaminen tarkoittaa vaatimusmäärittelytyön osaa, jossa selvitetään millaisia vaatimuksia ohjelmiston pitäisi toteuttaa. Vaatimusten kartoittaminen vaatii ymmärrystä asiakkaan

toimintaympäristöstä ja organisaatiosta, prosesseista sekä ongelmasta, jota ohjelmistolla ollaan ratkaisemassa. Ohjelmiston käytettävyyden on suoraan riippuvainen siitä, kuinka hyvin vaatimukset on kartoitettu. Zowghin ja Coulinin [91, s. 22] mukaan vaatimukset tulevat ensisijassa sidosryhmiltä. Käyttäjät ja asiantuntijat ovat parhaita lähteitä esittämään ongelmia ja käyttäjän tarpeita. Olemassa olevat ohjelmistot, prosessit ja dokumentaatio ovat myös vaatimusten lähteitä. Dokumentaatio olemassa olevista ohjelmistoista ja toimintaprosesseista voi tuottaa hyödyllistä tietoa organisaatiosta ja sen toimintaympäristöstä vaatimusmäärittelytyöhön.

Sommervillen ja Sawyerin [80, ss. 38–42] mukaan vaatimusten kartoittamisen kriittinen ongelma on yhteisen kielen puuttuminen loppukäyttäjän ja toimittajan välillä. Loppukäyttäjät eivät osaa aina ilmaista vaatimuksiaan riittävän yksityiskohteisella tasolla tai he eivät tiedä riittävän tarkasti mitä vaatimuksia ohjelmistolle pitäisi asettaa. Käyttäjät voivat tehdä myös epärealistisia vaatimuksia, koska he eivät ole välttämättä tietoisia toteutuksen kustannuksista. Eri sidosryhmillä voi olla poikkeavia vaatimuksia tai he voivat ilmaista samoja vaatimuksia eri tavoin, jolloin on tärkeää, että vaatimuksia keräävä henkilö osaa tulkita tilanteita oikein.

Nuseibehin ja Easterbrookin [62] mukaan vaatimusten keräämiseen voidaan hyödyntää erilaisia tapoja riippuen käytettävissä olevista resursseista ja siitä, millaista tietoa halutaan saada kerättyä. Vaatimusten kartoittamisessa voi hyödyntää haastatteluja, kyselyitä, ryhmätyöskentelyä, prototyypitystä ja käyttötapauskuvauksia. Mohapatra [59, s. 70] kehottaa valitsemaan sopivan menetelmän projektin luonteesta riippuen. Vaatimusten kartoittamisen menetelmän valintaan vaikuttaa sidosryhmien edustajien kyky ilmaista vaatimuksia ja toisaalta myös vaatimusten kartoittajan osaaminen vaatimusten keräämiseen ja arviointiin.

Haastattelu

Zowghi ja Coulin [91, s. 22] esittävät haastattelun olevan perinteisin ja käytetyin vaatimusten kartoittamisen tapa. Haastattelun sosiaalisesta perusluonteesta johtuen se on erittäin tehokas tapa kerätä aineistoa lyhyessä ajassa. Aineiston käytettävyyden ja hyödyllisyyden riippuu haastattelijan eli vaatimusten kartoittajan taidokkuudesta. Vaatimusten kartoittajalla tulee olla riittävä toimintaympäristön ja ongelman tuntemus, kokemusta vaatimusmäärittelyprosesseista ja hyvät sosiaaliset taidot. Paetschin ja muiden [63] ja Goguen ja Linden [32] artikkeleissa kuvataan kaksi erilaista tapaa haastattelujen läpiviemiseen. Haastattelu voi olla suljettu, jolloin haastattelijalla on etukäteen laaditut kysymykset, tai haastattelu voi olla avoin. Avoimessa haastat-

telussa haastattelijat ja sidosryhmien edustajat keskustelevat vapaamuotoisesti odotuksistaan tulevaa ohjelmistoa kohtaan. Zowghi ja Coulin [91, s. 25] toteavat, että keskustelussa jokin tärkeä osa-alue voi jäädä huomiotta, tai keskustelu voi rajautua vain johonkin tiettyyn aiheeseen. Avoin haastattelu sopii parhaiten tilanteisiin, jossa toimintaympäristön tuntemus on vähäistä, tai tilanteisiin, joissa strukturoidumpi haastattelu on järjestetty jo aikaisemmin. Gogue ja Linde [32] toteavat, että käyttäjillä on yleensä paras tieto omista työtehtävistään ja he tietävät miten toimintaprosessit käytännössä toimivat. Käytännön työtä on kuitenkin hankala kuvailla sanallisesti, jonka vuoksi työtehtävien seuraaminen voi olla tutkijalle tai vaatimusten kartoittajalle antoisampaa. Esimerkiksi, jos henkilöltä kysytään kuinka hän solmii kengännauhansa, on kysymykseen vaikea vastata, koska luontaisempaa on näyttää, kuinka solmiminen tapahtuu. Haastatteluissa ei siis kannata kysyä mitään, mitä ei normaalisti kuvailla sanallisesti [32].

Zowghin ja Coulinin [91, s. 25] mukaan haastattelu voidaan suorittaa myös strukturoituna haastatteluna, jolloin keskustelussa hyödynnetään ennalta määrättyjä kysymyksiä. Strukturoidun haastattelun onnistuminen edellyttää tietämystä ja kokemusta kysymysten muotoilusta, esittämisestä ja siitä, kuka kysymyksiin pystyy vastaamaan. Strukturoituja haastatteluja pidetään täsmällisinä ja tehokkaina, vaikka niillä on taipumus rajoittaa uusien ideoiden tai huomioiden syntymistä.

Goguen ja Linden [32] ja Zowghin ja Coulinin [91, s. 28] mukaan ryhmähaastattelujen avulla saadaan enemmän luontaista vuorovaikutusta käyttäjien välille kuin hyödyntämällä kyselyitä tai haastatteluja. Ryhmähaastattelu kokoaa yhteen erilaisia ryhmiä keskustelemaan jostakin aiheesta yhdessä vaatimuksia keräävän henkilön kanssa. Ryhmähaastattelussa tulee huomioida aihe, josta keskustellaan. Mukaan valitut ryhmät tulee koostua kyseisen aihealueen asiantuntijoista. Tärkein tekijä ryhmähaastattelun onnistumisessa on luottamuksellisen ja miellyttävän keskustelutilanteen mahdollistaminen. Hull [40, s. 109] korostaa, että ryhmähaastattelun tulee olla suunniteltu ja sitä tulisi johtaa keskustelun kautta. Sidoryhmien tulee saada käsitys, että vaatimusten antaminen ei ole vaikeaa tai aikaa vievää. Siksi perehdytys vaatimusmäärittelytyön tarkoituksesta auttaa ymmärtämään mitä heiltä odotetaan.

Keskustelu ja kysely

Hull [40, s. 106] ja Goguen ja Linde [32] toteavat, että haastattelun tai ryhmähaastattelun sijasta voidaan hyödyntää myös tavanomaista keskustelua. Keskustelu vaatii

osaamista vaatimusten kartoittajalta. Hänen tulee pystyä ilmaisemaan kysymyksensä ja ohjaamaan keskustelua siten, että käyttäjä ymmärtää, mistä puhutaan ja mitä häneltä odotetaan. Vaikeita ammattisanastoon liittyviä termejä tulisi välttää ja pyrkiä käyttäjälle tuttuun kieleen.

Zowghin ja Coulinin [91, s. 26] mukaan kyselyjä hyödynnetään pääasiassa vaatimusten kartoittamisen alkuvaiheessa. Kyselyt voivat sisältää avoimia ja suljettuja vastausvaihtoehtoja, joiden avulla voidaan kerätä nopeasti aineistoa useilta eri sidosryhmiltä. Kyselyjen muodostaminen vaatii hyvää asiantuntemusta kyselyn laatijalta. Toimintaympäristön tuntemus sekä ymmärrettävän kielen käyttäminen kyselyn laatimisessa on ehdotonta. Sidoryhmien tulee ymmärtää kysymykset ja vastausvaihtoehdot, jotta tulokset eivät vääristy. Hyvästäkin suunnittelusta riippumatta kyselyillä ei välttämättä saavuteta kovin syvällistä ymmärrystä sidoryhmien tarpeista ja vaatimuksista. Kyselyitä voidaan hyödyntää kuitenkin epävirallisina tarkistuslistoina, joiden avulla voidaan varmistua jo aikaisessa vaiheessa että ohjelmiston peruselementit on huomioitu oikein. Tarkistuslistat toimivat myös pohjan luomiseen tulevia tarkempia haastatteluja varten.

Aivoriihi ja havainnointi

Zowghi ja Coulin [91, s. 28] esittävät aivoriihen nopeaksi vaatimusten kartoittamisen tavaksi. Aivoriiheen kerätään sidoryhmien edustajia kehittämään ideoita mahdollisimman nopeasti keskittymättä mihinkään tiettyyn ongelmaan tai aihealueeseen. Aivoriihessä tulee pyrkiä välttämään ideoiden rajoittamista tai kritisointia. Aivoriihissä ei yleensä keskitytä hakemaan ratkaisua isoihin ongelmiin tai tekemään päätöksiä ratkaisusta. Tekniikan hyödyntäminen soveltuu parhaiten projektin alkuvaiheeseen, jossa määritellään ohjelmistohankinnan suuntaa. Aivoriihen hyödyntäminen edesauttaa vapaata ajattelua ja ideoiden ilmaisemista sekä antaa mahdollisuuden uusien innovatiivisten ratkaisujen löytämiseen.

Zowghi ja Coulin [91, s. 29] kuvaavat havainnoinnin mahdollistavan nykytilanteeseen ja toimintaprosesseihin tutustumisen. Vaatimusten kartoittamisesta vastaava henkilö voi tutustua organisaation työskentelytapoihin havainnoimalla, jonka avulla saavutetaan ymmärrys ongelmasta ja siitä, kuinka tilanne on ratkaistu ennen ohjelmiston kehittämistä. Havainnointia hyödynnetään yleensä tarkentavana menetelmänä jonkin toisen vaatimusten kartoittamisen menetelmän rinnalla.

Käyttötapausten kuvaaminen

Käyttötapausten kuvaaminen on usein hedelmällistä sekä tulevalle ohjelmiston käyttäjälle että haastattelijalle. Hullin [40, s. 100] ja Paetsch ja muiden [63] mukaan käyttötapaukset kuvaavat käyttäjän ja ohjelmiston välistä vuorovaikutusta erityisesti käyttäjän näkökulmasta. Käyttötapauksissa käyttäjä miettii jotakin hänelle tuttua työtehtävää, miten tehtävä nyt suoritetaan ja miten sen voisi suorittaa ohjelmiston avulla. Käyttötapausten kuvaaminen auttaa käyttäjää ja kehittäjää ymmärtämään vaatimukset selkeämmin. Tämän menetelmän hyödyntämisessä riittää, että käyttäjä hallitsee toimintaprosessin, jonka suorittamista halutaan ohjelmiston avulla tehostaa. Sommervillen ja Sawyerin [80, s. 100] mukaan käyttötapauskuvaukset voivat olla rakenteeltaan erilaisia, mutta niiden tulisi sisältää vähintään kuvaus ohjelmiston tilasta ennen käyttötapausten aloittamista, käyttötapausten aikana tapahtuvat toimet, normaalista toiminnasta poikkeavat toimet, tieto muista tehtävistä, joita voi olla käynnissä samaan aikaan ja kuvaus ohjelmiston tilasta käyttötapausten jälkeen.

Käyttötapausten kuvaaminen toteutetaan yhteistyössä vaatimusten kartoittajan ja loppukäyttäjän kanssa. Loppukäyttäjä kuvailee käyttötapausta ja vaatimusten kartoittaja kirjaa ylös käyttäjän kommentteja, ongelmia ja ehdotuksia. Loppukäyttäjä kuvailee käyttötapausten alusta loppuun ja vaatimusten kartoittaja kysyy tarkentavia kysymyksiä. Tällaisia voivat olla, kuinka käyttötapausta suoritetaan ohjelmiston avulla, keitä muita käyttäjiä tapauksen läpiviemiseen liittyy ja mitä tapahtuisi, jos tapaus hoidettaisiin jollakin toisella tavalla. Käyttötapausten esittämiseen voidaan hyödyntää myös prototyyppiä, jos sellainen on käytettävissä. Käyttötapausten läpikävely loppukäyttäjän kanssa auttaa vaatimusten kartoittajaa ymmärtämään, mitä vaatimuksia loppukäyttäjällä on ohjelmistoa kohtaan. Lopuksi käyttötapaukset täytyy purkaa toiminnallisiksi ja ei-toiminnallisiksi vaatimuksiksi. [80, s. 100]

Zowghi ja Coulin [91, s. 30] ja Hull [40, s. 119] esittävät prototyyppityksen hyödyllisenä silloin, kun kehitetään uutta ohjelmistoa, jonka käyttöliittymää loppukäyttäjien on vielä vaikea hahmottaa. Ohjelmiston kehittäjät luovat pienellä resurssilla prototyyppin ohjelmistosta, jota voidaan esitellä sidosryhmien edustajille vaatimusten keräämisen helpottamiseksi. Prototyyppitys on erityisen hyödyllinen silloin, kun teknologiaratkaisu tai sen tarjoamat mahdollisuudet eivät ole sidosryhmille tuttuja. Prototyyppin hyödyntäminen aktivoi sidosryhmiä ja erityisesti loppukäyttäjää osallistumaan vaatimusten kartoittamiseen. Prototyyppityksessä tulee varmistaa, että sidosryhmien edustajat ymmärtävät kyseessä olevan testiversio ohjelmistosta, jota ei oteta tuotantokäyttöön tai että sen kehittämiseen ei käytetä liikaa projektin resurs-

seja. Denger ja Olsson [22, s. 174] korostavat prototyypin pienentävän vaatimusten kuvausten ja käytännön välistä kuilua. Prototyypityksen avulla toteutetut kokeilut antavat selkeämmän kuvan vaatimusten toteutuskelpoisuudesta, joka osaltaan voi pienentää projektin todellisia kustannuksia.

Yhteenveto

Vaatimusten kartoittamiseen liittyvät menetelmät tuovat monia vaihtoehtoja vaatimusten kartoittamiseen erilaisissa projekteissa ja tilanteissa. Hyvästä vaatimusten kartoittamisen suunnittelusta huolimatta vaatimusten saavuttaminen voi olla haastavaa, mikäli sidosryhmät eivät ole halukkaita osallistumaan vaatimusmäärittelytyöhön. Mohapatran [59, s. 67] mukaan sidosryhmien kiinnostuksen puute vaatimusmäärittelytyötä kohtaan voi johtua heikosta vuorovaikutuksesta vaatimusten kartoittajan ja sidosryhmien välillä. Vaatimusten kartoittaminen vaatii hyviä ihmishuuhdetaitoja sekä ongelman ja sovellusalan tuntemusta. Vaatimusten kartoittaja voi tahtomattaan antaa itsestään kuvan soveltumattomana avuntarjoajana, mikäli hän ei tunne organisaation ongelmaa riittävän hyvin. Sidosryhmien edustajat voivat myös kokea projektin kuluttavan heidän aikaansa ja vaatimusten kartoittajan sekaantuvan heidän työhönsä.

Mohapatra [59, s. 67] toteaa tiedon olevan valtaa. Loppukäyttäjät eivät välttämättä halua siksi jakaa osaamistaan muille. He voivat kokea uuden ohjelmiston ja toimintaprosessin uudistamisen vähentävän tai aliarvioivan heidän työtään. Usein loppukäyttäjä ei ole vakuuttunut uuden ohjelmiston tai toimintatavan tarpeellisuudesta eikä siksi ole sitoutunut vaatimusmäärittelytyöhön. Useimmat vaatimusmäärittelytyöhön haluttomat osallistujat voivat osoittautua kuitenkin erinomaisiksi jäseniksi vaatimusten kartoittamisessa, mikäli vaatimusten kartoittaja pystyy löytämään ratkaisun ongelmallisen tilanteen korjaamiseksi.

Pahinta, mitä vaatimusmäärittelytyössä voidaan tehdä, on sellaisten vaatimusten muodostaminen, jotka eivät vastaa loppukäyttäjien tarpeita. Vaatimusmäärittelytyössä tärkeintä on pyrkiä tuottamaan toimiva ja käyttäjiä palveleva ohjelmisto. Jos vaatimuksia ei saada määritettyä riittävän tarkalla tasolla, tulee ohjelmistohan- kinta epäonnistumaan, vaikka projekti toteutettaisiin muilta osin onnistuneesti. [55]

2.2.2 Analysointi

Sommervillen ja Sawyerin [80, s. 111] mukaan vaatimusten kartoittamisen jälkeen vaatimukset tulee analysoida, jotta mahdolliset ristiriidat, päällekkäisyydet, puutteet ja epä johdonmukaisuudet voidaan tunnistaa. Kun analysointi on valmis ja sen tuottama tieto on jaettavissa, sidosryhmät voivat neuvotella ristiriidoista ja priorisoida vaatimukset. Batool ja muiden [8] ja Paetsch ja muiden [63] mukaan vaatimusten analysoinnissa arvioidaan vaatimusten todellinen tarpeellisuus, johdonmukaisuus, täydellisyys ja toteutettavuus. Sidosryhmät analysoivat vaatimukset ja määrittävät ne tärkeysjärjestykseen. Priorisoinnin avulla saadaan selville kriittisimmät vaatimukset. Analysointivaiheessa jokaisen sidosryhmän edustajan läsnäolo on tärkeää, jotta jokaiselle käyttäjäryhmälle tärkeät vaatimukset voidaan huomioida mahdollisimman kattavasti. Priorisointi auttaa sidosryhmiä myös ymmärtämään, mitä vaatimuksia voidaan jättää toteuttamatta tiukan aikataulun ja resurssien puitteissa. Ohjelmistokehittäjät voivat tarvittaessa ohjata keskustelua ilmaisemalla vaatimusten teknisistä riskeistä, kustannuksista tai vaikeista toteutustavoista [63].

Sommervillen ja Sawyerin [80, s. 117] mukaan vaatimusten analysoinnin tavoitteena on löytää ongelmia vaatimuksissa. Analysointi voidaan toteuttaa tarkistuslistan avulla, jolloin organisaatio kehittää aikaisemmin havaittuihin ongelmiin perustuvan listan ja vertaa jokaista vaatimusta listaan. Jos ongelmia havaitaan, ne tulee kirjata vaatimuksen yhteyteen. Tarkistuslistan avulla analysointivaihe voidaan toteuttaa systemaattisesti virheiden määrää ja epä johdonmukaisuutta vähentäen. Jokaisen organisaation tulisi kehittää oma tarkistuslista, joka pohjautuu organisaatioon aikaisempaan tietoaaineistoon ohjelmistoprojektien vaatimusmäärittelytyöistä ja ongelmista.

Esimerkki tarkistuslistasta on esitetty taulukossa 2.1. Jokainen kartoitetuista vaatimuksista tulee verrata tarkistuslistan kysymyksiin. Kun mahdollisia ongelmia havaitaan, ne tulisi kirjata vaatimusdokumenttiin tai erilliseen analysointilistaan. Tarkistuslistaan valittavien kysymysten tulisi olla paremmin yleisiä kuin liian rajattuja, jotta listaa voi hyödyntää mahdollisimman monipuolisesti erilaisten ohjelmistojen vaatimusmäärittelytyössä. Kysymysten ei tulisi sisältää laatuun liittyviä huomioita. Jotta analysointi tarkistuslistan avulla on hyödyllistä, kysymysten määrä kannattaa rajata korkeintaan kymmeneen. Tarkistuslista kehittyy ideaalitalanteessa organisaation ymmärryksen kasvaessa vaatimusmäärittelytyöstä. [80, ss. 118–119]

Taulukko 2.1: Sommervillen ja Sawyerin esittämä malli tarkistuslistasta [80, s. 118]

Tarkistuslistan kohta	Kuvaus
Ennenaikainen suunnittelu	Sisältääkö vaatimus ennenaikaista suunnittelua?
Yhdistetyt vaatimukset	Voiko vaatimuksen jakaa useammaksi vaatimukseksi?
Tarpeettomat vaatimukset	Onko vaatimus ohjelmiston käytölle kriittinen?
Organisaation tavoitteet	Edistääkö vaatimus organisaation tavoitteiden saavuttamista?
Vaatimusten epäselvyys	Voiko vaatimusta tulkita eri tavoin?
Vaatimusten realistisuus	Onko vaatimus realistinen toteuttaa valitulla teknologialla?
Vaatimusten testattavuus	Pystyykö vaatimuksen testaamaan toteutetussa ohjelmistossa?

2.2.3 Neuvottelu

Grünbacher ja Seyff [34, s. 143] kuvaavat ristiriitoja syntyvän lähes kaikissa ohjelmistoprojekteissa, kun sidosryhmät pyrkivät sovittamaan yhteen vaatimuksiaan. Ristiriitoja jätetään usein huomioimatta tai ne hoidetaan riittämättömällä tarkkuudella. Sidosryhmillä voi olla hyvin erilaisia näkökulmia ohjelmiston kehittämiseen. Loppukäyttäjät ovat usein kiinnostuneita ominaisuuksista, hyvästä käytettävyydestä ja nopeasta käyttöönnotosta. Organisaation johto voi olla kiinnostunut kustannustehokkuudesta, standardien noudattamisesta ja resurssirajoitteista. Vaikka ristiriitojen aiheutuminen on enemmän sääntö kuin poikkeus, vain harva ohjelmistoprojektin prosessimalli huomioi, kuinka ristiriidat käsitellään ja ratkaistaan. Grünbacher ja Seyff [34, s. 143] toteavat, että ohjelmistokehitys on yhteistyötä vaativa prosessi, joka edellyttää mielipiteiden esiintuomista, olivatpa ne yhdenmukaisia tai vastakkaisia. Kaikkien sidosryhmien näkökulmat täytyy ymmärtää ja sovittaa yhteen, jotta yhteisesti hyväksyttävät vaatimukset voidaan määrittää. Neuvottelun tavoitteena ei ole se, että kaikki sidosryhmät olisivat samaa mieltä, vaan neuvottelun avulla pyritään saavuttamaan ymmärrys, miksi sidosryhmät ovat eri mieltä. Tunnistetut erimielisyydet ovat riskejä ohjelmistoprojektille, jonka vuoksi projektin johdon tulee puuttua niihin.

Grünbacherin ja Seyffin [34, ss. 143–145] mukaan vaatimuksista neuvottelu ei ole vain kertaluontoinen osa projektia, vaan sitä tulisi hyödyntää jo projektin alussa ja toistaa myöhemmissä vaiheissa vaatimusmäärittelyprosessia. Neuvottelu auttaa sidosryhmien edustajia ymmärtämään ohjelmistoprojektin rajoitteita kuten esimerkiksi budjettia ja aikataulua. Ilman neuvottelua projektissa voidaan pyrkiä hyödyntämään olemassa olevaa ratkaisua tai kuviteltua ratkaisua sen sijaan, että pyrittäisiin kehittämään uusia ratkaisuja, jotka hyödyttävät kaikkia osapuolia. Sidosryhmillä ei aina ole tarkkaa ymmärrystä omista vaatimuksista, joka ilmenee usein epävarmuutena. Neuvottelu vähentää epävarmuutta korostamalla vaatimuksia, jotka ovat tärkeitä ja edistää siten sidosryhmien yhteistä näkemystä. Neuvottelujen aloittaminen jo aikaisessa vaiheessa ohjelmistoprojektia auttaa sidosryhmiä sopeutumaan myös mahdollisiin projektin aikana kohdattaviin muutoksiin. Kun sidosryhmät ovat tietoisia vaatimusten ongelmista ja toisaalta mahdollisuuksista, on heidän helpompi hyväksyä teknologioihin, ihmisiin ja sopimukseen liittyvät muutokset.

Grünbacherin ja Seyffin [34, ss. 143–145] mukaan vaatimuksista neuvottelu on aina oppimisprosessi sekä sidosryhmien edustajille, että projektiryhmälle ja vaatimusten kartoittajalle. Sidosryhmien edustajat ovat projektissa mukana erityisesti oman osaamisensa, taustansa ja tavoitteidensa puolesta. Sidosryhmien edustajat tuovat esiin omia näkemyksiään ja tavoitteitaan ja neuvottelevat niistä. Vaatimusten kartoittaja oppii runsaasti sidosryhmien edustajien työstä ja heidän arjestaan, kun sidosryhmien edustajat puolestaan oppivat lisää muiden sidosryhmien edustajien työstä sekä siitä, mitä valittu teknologia mahdollistaa ja mitä on realistista resursien puitteissa toteuttaa. Hiljaisen tiedon siirtäminen on myös neuvottelun tavoite. Keskustelun avulla voidaan saada tietoon aikaisemmin havaitsematta jääneitä odotuksia ja toiveita ohjelmistoa kohtaan.

Sommerville ja Sawyer [80, s. 125] toteavat, että useat organisaatiot eivät varaa projektissa riittävästi aikaa vaatimuksista neuvotteluun ja ristiriitojen selvittämiseen. Organisaatiot voivat pitää ristiriitojen ilmenemistä myös projektin epäonnistumisena ja sulkea ristiriidan siksi pois projektista. Ongelmat ovat kuitenkin jopa luonnollinen osa projektia. Ne kertovat erilaisten sidosryhmien erilaisista tarpeista ohjelmistoa kohtaan. Neuvottelun toteuttaminen on yksinkertaisinta järjestää tapaamisena, joka sisältää seuraavat kolme vaihetta [80, s. 125]:

1. Vaatimukseen liittyvien ongelmien selvittäminen.
2. Keskustelu sidosryhmien edustajien kesken siitä, kuinka ongelmat voitaisiin

ratkaista. Jokaisen sidosryhmän edustajan tulisi antaa oma mielipiteensä ratkaisuehdotukseksi. Keskustelun ohjaamista ja ajankäyttöä varten on jokaista ongelmaa kohti syytä varata rajallinen määrä aikaa.

3. Ratkaisuihin päättäminen. Ratkaisu voi olla ongelmallisten vaatimusten poistaminen, vaatimusten muuttaminen tai vaatimuksesta lisätiedon kartoittaminen.

Sommerville ja Sawyer [80, ss. 126–127] suosittelevat, että neuvottelua tulisi johtaa henkilön, joka ei ole minkään sidosryhmän edustaja. Ulkopuolisen edustajan on helpompi pidättäytyä neutraalissa roolissa vaatimusten suhteen. Neuvotteluita voidaan käydä useammin kuin kerran ja jokaisessa neuvottelussa tulisi käsitellä vain prioriteettina tärkeimpiä ongelmallisia vaatimuksia. Kasvokkain pidettävät tapaamiset ovat usein tehokkaampia ja vähemmän kuormittavia kuin etäyhteyksin järjestettävät tapaamiset.

2.2.4 Dokumentointi

Sommervillen ja Sawyerin [80, ss. 38–42] mukaan vaatimusdokumentti on selvitys ohjelmiston vaatimuksista asiakkaita, loppukäyttäjiä ja kehittäjiä varten. Vaatimusdokumentin tulisi sisältää vaatimukset johdonmukaisesti ja mahdollisimman täydellisesti kuvattuina siten, että dokumentti on ymmärrettävästi luettavissa eri käyttäjäryhmillä. Vaatimusdokumentti kattaa sekä toiminnalliset että ei-toiminnalliset vaatimukset. Toiminnallisia vaatimuksia ovat ohjelmiston käyttöön liittyvät vaatimukset, jotka kerätään asiakkailta ja sidosryhmiltä. Ei-toiminnallisia vaatimuksia ovat esimerkiksi ohjelmiston saatavuuteen tai tietoturvaan liittyvät vaatimukset. Sommervillen [78] mukaan mitä täydellisempi ja johdonmukaisempi vaatimusdokumentti on, sitä todennäköisemmin ohjelmisto on luotettava ja se voidaan toimittaa sovitussa ajassa.

Hullin [40, ss. 77–89] mukaan vaatimusten dokumentointi ei ole teknistä kirjoittamista. Vaatimusdokumenttia laadittaessa tulee huomioida, että dokumentin tulee olla luettava, ja kirjatut vaatimukset hallittavia. Riippumatta dokumentointitavasta tulisi seuraavat ehdot täyttyä [40, s. 89]:

- Kaikki vaatimukset on kirjattu yksilöidysti
- Vaatimus on mahdollista toteuttaa laillisesti ja teknisesti resurssien puitteissa

- Vaatimukset eivät ole ristiriidassa keskenään
- Jokainen vaatimus on esitetty vain kerran
- Toisiinsa liittyvät vaatimukset on esitetty lähekkäin
- Dokumentissa on selkeä rakenne
- Vaatimukset ovat jäljitettävissä

Vaatimusdokumentti voi ohjelmistosta riippuen olla hyvinkin laaja, jonka vuoksi vaatimukset ilmaistaan usein hierarkkisessa rakenteessa. Hierarkkinen rakenne sisältää vaatimukset lajiteltuina kategorisesti päälukuihin ja alalukuihin. Dokumentit voivat olla sisällöltään myös yhdistelmä teknistä ja kansankielistä tekstiä, joka auttaa sidosryhmiä ymmärtämään vaatimukset. Vaatimusdokumentissa voidaan hyödyntää jaottelua, jossa ohjelmiston tärkeimmät vaatimukset on kerätty erilleen muista vaatimuksista. Tämän esitystavan avulla on helppoa saavuttaa nopeasti ymmärrys kehitettävästä ohjelmistosta. Jokaisen listalle valitun vaatimuksen tulee vastata negatiivisesti seuraaviin kysymyksiin, *”jos ohjelmisto ei sisällä tätä ominaisuutta, ostaisinko sen silti?”* ja *”jos ohjelmisto ei toteuta tätä toimintoa, haluaisinko silti sen?”* [40, ss. 77–80]. Hullin [40, s. 90] mukaan dokumentoinnissa vaikeinta on sen aloittaminen. Siksi vaatimusdokumentin rakenne kannattaa hahmotella jo projektin alussa, ja aloittaa vaatimusten kirjaaminen mahdollisimman pian. Vaatimukset voivat olla epätäydellisiäkin. Tärkeintä on kirjoittaa selkeällä kielellä ja tuottaa alustava versio dokumentista kommentointia ja palautteen antoa varten.

Mohapatran [59, s. 213] mukaan vaatimusdokumentti kertoo, miten ohjelmisto toimii, mutta ei ota kantaa, miten ohjelmisto toteutetaan. Vaatimusdokumenttiin ei siten tule sisällyttää suunnitteluun liittyviä huomioita, kuten ohjelmiston rakennetta tai tiedonsiirtoa. Vaatimusdokumenttiin ei kirjata ohjelmistoprojektiin liittyviä tietoja kuten kustannuksia, aikataulua tai hyödynnettävää prosessimallia.

Vaatimusdokumentin luomiseen ja ylläpitoon on saatavilla erilaisia malleja ja standardeja. IEEE standardit 830-1998 [44] *”Recommended Practice for Software Requirements Specifications”* ja IEEE 1233, 1998 edition [42] *”Guide for Developing System Requirements Specifications”* tarjoavat ohjeen ja mallipohjan vaatimusdokumenttia varten. IEEE standardin 830-1998 mukainen vaatimusdokumentin sisällysluettelo voi näyttää esimerkiksi seuraavalta [44]:

1. Johdanto

- Vaatimusdokumentin tarkoitus
- Ohjelmiston tavoite
- Määritelmät ja lyhenteet
- Viittaukset
- Yleiskatsaus

2. Yleiskuvaus

- Ohjelmiston näkökulma
- Ohjelmiston toiminnot
- Kuvaus käyttäjistä
- Ohjelmiston rajoitukset
- Oletukset ja riippuvuudet

3. Vaatimukset

4. Liitteet

5. Hakemisto

Paetschin ja muiden [63] mukaan vaatimusten dokumentointi on perinteisessä vaatimusmäärittelytyössä koko prosessin läpiviennin kannalta tärkein dokumentti ja se voi kuulua myös osaksi projektin sopimusta. Dokumentaatioon palataan jokaisessa ohjelmistokehityksen vaiheessa ja sen avulla hallitaan myös vaatimusten muutoksia. Hyvän vaatimusdokumentin tulisi olla yksiselitteinen, virheetön, ymmärrettävä ja johdonmukainen. Hullin [40, s. 90] mukaan dokumentoinnissa tulee välttää rönsyilevää kieltä, spekulointia, epämääräisiä sanoja kuten usein, yleensä, tyypillisesti ja sellaisten vaatimusten kirjaamista, jotka eivät ole vaatimuksia.

2.2.5 Validointi

Hofmannin ja Lehnerin [39] mukaan validoinnin tarkoituksena on varmistaa, että vaatimukset vastaavat sidosryhmien ja kehittäjien tarpeita. Validoinnissa voidaan hyödyntää katselmoiteja, läpikävelyjä ja skenaarioita vaatimuksista. Tärkeintä on, että sidosryhmät ja kehittäjät ymmärtävät validoinnin jälkeen kehittävänsä oikeanlaista ohjelmistoa, joka vastaa käyttäjien tarpeisiin. Mohapatran [59, s. 223] mukaan

asiakkaan näkökulmasta validoinnissa varmistetaan vaatimusten tarpeellisuus, johdonmukaisuus, täydellisyys ja toteutuskelpoisuus. Kehittäjän näkökulmasta varmistetaan todennettavuus, ymmärrettävyys, jäljitettävyys ja hyödynnettävyys. Baatoolin ja muiden [8] ja Paetschin ja muiden [63] mukaan validoinnin lopputuotoksena saadaan lista, joka sisältää validoinnissa raportoidut huomiot vaatimusdokumentissa sekä niiden korjausehdotukset. Validoinnissa listatut ongelmat on korjattava ennen ohjelmistokehityksen aloittamista.

Sommerville ja Sawyer [80, ss. 190–191] korostavat sidosryhmien roolia validoinnin aikana. Validointiprosessiin tulee ottaa mukaan sidosryhmät, vaatimusten kirjoittajat sekä ohjelmiston kehittäjät. Vaatimusten validoinnin haasteellisuus liittyy siihen, että validoinnin aikana ei ole olemassa vielä ohjelmistoa, jota vasten vaatimuksia voitaisiin verrata. Vaatimuksia ei voida havainnollistaa siten, että voitaisiin varmistua vaatimusten oikeellisuudesta. Vaatimusten validoinnista saatava hyöty on erityisesti varmuus vaatimusdokumentin selkeästä esitystavasta ja sen kelpoisuudesta ohjelmiston kehittämistä varten.

Sommervillen ja Sawyerin [80, ss. 195–200] mukaan vaatimusten validointiin osallistuvat samat sidosryhmien edustajat, jotka osallistuvat vaatimusneuvotteluihin ja jotka ovat tietoisia vaatimusten kehittymisestä projektin aikana. Validoinnin apuna voidaan hyödyntää samanlaisia tarkistuslistoja kuten vaatimusten analysoinnissa. Vaatimusten analysointivaiheessa tarkistuslistan kohteet eivät sisällä vielä laatuun liittyviä seikkoja, mutta validoinnissa dokumentin ollessa valmis myös laatu huomioidaan tarkistuslistassa. Validointivaiheen tarkistuslista voi sisältää yksittäisiä vaatimuksia koskevien kysymysten lisäksi myös vaatimusdokumenttiin liittyviä kysymyksiä sekä vaatimusten välisiä kysymyksiä. Tarkistuslistat tulee kirjoittaa selkeällä yleiskielellä, jotta se on laajasti ymmärrettävissä eri sidosryhmissä. Hyvä tarkistuslista sisältää korkeintaan 10 tarkkaan suunniteltua kysymystä. Tarkistuslista toimii parhaassa tapauksessa muistilistana sille, mitä lukijan tulisi vaatimusdokumentissa huomioida sitä tarkastaessaan.

2.3 Vaatimusten hallinta

Hullin [40, s. 160] mukaan vaatimusten hallinta on haastava osa vaatimusmäärittelytyötä, sillä hyvin harvoilla ihmisillä on osaamista vaatimusten hallinnasta ohjelmistoprojekteissa. Osaamisen puute ja vaatimusten hallinnan haasteellisuus johtuu suurelta osin siitä, että harvat organisaatiot ovat määrittäneet vaatimusten hal-

linnan osaksi ohjelmistoprojekteja ja vaatimusmäärittelytyötä. Jos vaatimusten kartoittaminen sidosryhmiltä ja tärkeimmältä käyttäjäryhmältä loppukäyttäjiltä on jätetty tekemättä, ei vaatimuksia voida hallita. Young [88, s. 120] jakaa vaatimusten hallinnan kahteen osaan, joiden avulla varmistetaan, että loppukäyttäjän tarpeet on täytetty: Vaatimuksia ja niiden muutoksia tulee hallita ja epä johdonmukaisuudet ohjelmistokehityksen ja vaatimusten välillä tulee tunnistaa.

Seuraavissa alaluvuissa käsitellään vaatimusten hallintaan liittyvät toimenpiteet.

2.3.1 Muutosten hallinta

Hullin [40, ss. 165–166] ja Berenbachin [12, s. 195] mukaan vaatimusten kartoittamisen aikana sidosryhmiltä kerättävissä vaatimuksissa esiintyy runsaasti muutoksia. Vaatimusten kartoittamisen aikana muutoksia ei ole tarpeen erityisesti hallita, vaan tilanne saa elää prosessin vaiheen mukaisesti. Vaatimusten analysoinnin, dokumentoinnin ja validoinnin jälkeen vaatimusdokumenttiin ei kuitenkaan voi tehdä enää vapaasti muutoksia tai lisäyksiä. Vaatimuksen muuttuessa seuraavat toimenpiteet tulee huomioida [40, s. 166]:

- Ehdotettu muutos kirjataan
- Muutoksen aiheuttamat vaikutukset tunnistetaan
- Muutoksen hyväksymisestä tai hylkäämisestä tehdään päätös
- Muutoksen toimeenpanon ajankohta päätetään

Ehdotetun muutoksen lisäksi tulee antaa selvitys perusteluista, miksi muutos on tärkeä ja tunnistaa sidosryhmien vaatimukset, joita muuttaminen, lisääminen tai poistaminen koskee. Muutosta pyytänyt käyttäjä tai organisaatio tulee myös kirjata. Muutoksen aiheuttamat vaikutukset riippuvat projektin vaiheesta, jossa muutospyyntö tuodaan ilmi. Muutoksen kohteena olevien vaatimusten vaikutus ohjelmiston suunnitteluun, kehittämiseen ja toimintoihin tulee selvittää. Päätöksen muutos ehdotuksen hyväksymisestä tai hylkäämisestä tekee muutosten hallinnan työryhmä. Vain nimetyt henkilöt voivat päättää muutosten hyväksymisestä tai hylkäämisestä. Päätöstä valmisteltaessa tulee huomioida organisaation tavoitteet, ohjelmiston laajuus, ohjelmistokehityksen tilanne sekä ohjelmiston käyttötarkoitus. Jos

muutosehdotus hyväksytään, tehdään vielä päätös muutoksen toimeenpanon ajankohdasta. Jos muutos on kriittinen, sen toimeenpano huomioidaan välittömästi ohjelmistokehityksessä. Mikäli muutos on tärkeä, mutta ei kriittinen lisättäväksi kehitettävään ohjelmistoversioon, voidaan muutoksen huomioiminen siirtää ohjelmiston seuraavaan versiokehitykseen. [40, s. 166]

2.3.2 Version hallinta

Pohlin ja Ruppin [66, ss. 128–129] mukaan vaatimusten kehittämisen aikana uusia vaatimuksia syntyy, vaatimukseen tulee muutoksia ja joitakin vaatimuksia voidaan poistaa. Muutokset vaatimuksissa johtuvat eri syistä. Usein esimerkiksi sidosryhmät ymmärtävät ohjelmistoprojektin edetessä vaatimukset ohjelmistoa kohtaan selkeämmin ja muutoksia tehdään ymmärryksen kasvaessa. Vaatimusten versiointi pyrkii yksilöimään jokaisen vaatimuksen ja pitämään yllä muutostietoja vaatimuksen kehittämisestä.

Vaatimusten versioinnissa voidaan hyödyntää pääversioita ja luonnosversioita. Versiointi aloitetaan luvusta 0.1, jonka jälkeen jokaisen pienen muutoksen jälkeen versionumero kasvaa aina yhdellä desimaalilla. Kolmen muutoksen jälkeen versionumero olisi näin ollen 0.4. Kun vaatimukset validoidaan, vaatimuksesta voidaan julkaista pääversio 1.0, joka ilmaisee vaatimuksen olevan sillä hetkellä parhaan tietämyksen mukaan valmis. Jos muutoksia aiheutuu vielä myöhemmässä vaiheessa, voidaan luonnosversiointia jatkaa, jolloin versionumero muuttuu 1.1. [66, ss. 128–129]

2.3.3 Tilan seuranta ja jäljitettävyys

Sommervillen ja Sawyerin [80, ss. 224–225] mukaan vaatimusten jäljitettävyys kuuluu omana osanaan vaatimusten hallintaan. Vaatimusten jäljitettävyys auttaa ymmärtämään vaatimusten välisiä riippuvuuksia ja vaatimusten sekä ohjelmistokehityksen välisiä riippuvuuksia. Vaatimusten muutoksista vastaavat henkilöt vastaavat myös vaatimusten jäljitettävyyteen liittyvän tiedon ylläpitämisestä, koska heillä on paras ymmärrys, miten vaatimukset ovat kehittyneet projektin aikana ja miten tiedot tulisi esittää. Jokainen organisaatio määrittää vaatimusten jäljitettävyyteen liittyvät periaatteet ja dokumentoinnin laajuuden. Jäljitettävyys kuuluu olennaisena osana projektin resurssien ja laadunhallintaan. Sen avulla voidaan luoda järkevissä mittakaavassa toteutettava ja ylläpidettävä toimintamalli.

Vaatimusten jäljittäminen voidaan toteuttaa yksinkertaisimmillaan taulukkolaskennan avulla, jolloin hyödynnetään rivejä ja sarakkeita tiedon esittämiseen. Taulukon ylläpitäminen voi kuitenkin käydä hankalaksi vaatimusmäärän kasvaessa satoihin ja tuhansiin vaatimuksiin. Suurissa ohjelmistoprojekteissa vaatimusten hallinnan apuna voidaan hyödyntää erilaisia kaupallisia ohjelmistotuotteita. [80, s. 229]

2.4 Perinteiset prosessimallit

Vaiheittain etenevän vesiputousmallin kehitti Winston Royce 1970-luvulla, joskaan Royce ei nimennyt prosessimallia vesiputousmalliksi [70]. Mohapatran [59, s. 18] mukaan vesiputousmallista tuli laajasti tunnettu ja käyttöön otettu ohjelmistokehityksen prosessimalli. Malli oli kaivattu apu ohjelmistokehityksen tueksi. Se tarjosi käytännön neuvoja ja toimenpiteiden suoritusjärjestyksen ohjelmistokehitykselle.

Paetsch ja muut [63] kuvaavat perinteisen vaatimusmäärittelytyön tavanomaisesti ohjelmistotuotannon prosessiksi, jonka tavoitteena on tunnistaa, analysoida, dokumentoida ja vahvistaa kehitettävän ohjelmiston vaatimukset. Vaatimusmäärittely auttaa sekä asiakasta että kehittäjää ymmärtämään, mitä halutaan ja mitä on tarkoituksena alkaa kehittää ennen kuin varsinainen ohjelmistokehitys aloitetaan. Vaatimusmäärittelytyössä syntyvä dokumentaatio esittää tulevaan ohjelmistoon kohdistuvat erilaiset vaatimukset ja käyttötapaukset. Dokumentaation tarkoitus ei ole kertoa, kuinka suunnitelmat toteutetaan.

Sommerville ja Sawyer [80, ss. 4–5] määrittelevät vaatimusten kartoittamisen tapahtuvan ohjelmistoprojektin alkuvaiheessa kuvauksena siitä, mitä aiotaan kehittää. Vaatimukset ovat kuvauksia siitä, kuinka ohjelmiston tulisi käyttäytyä, tai kuvauksia järjestelmän ominaisuuksista. Vaatimukset voivat olla myös ohjelmiston rajoituksia. Vaatimusmäärittelytyö puolestaan on prosessi, joka sisältää vaatimusten löytämisen, dokumentoinnin ja hallinnan.

Paetsch ja muut [63] toteavat, että hyvällä etukäteen tehdyllä suunnittelulla pyritään saavuttamaan asiakasta tyydyttävä lopputulos siten, että kalliita muutoksia ei enää jälkikäteen tarvitse toteuttaa. Perinteisen vaatimusmäärittelytyön tavoite saavutetaan mahdollisimman täydellisen vaatimusdokumentaation luomisella ennen ohjelmistokehityksen alkua, ja mahdollisten virheiden sekä puuttuvien vaatimusten tunnistamisella aikaisessa vaiheessa projektia. Paetschin ja muiden [63] ja Batoolin ja muiden [8] mukaan perinteinen vaatimusmäärittelytyö koostuu viidestä osaluokasta, jotka ovat vaatimusten kartoittaminen, analysointi, dokumentointi, vali-

dointi ja hallinta.

2.4.1 Vesiputousmalli

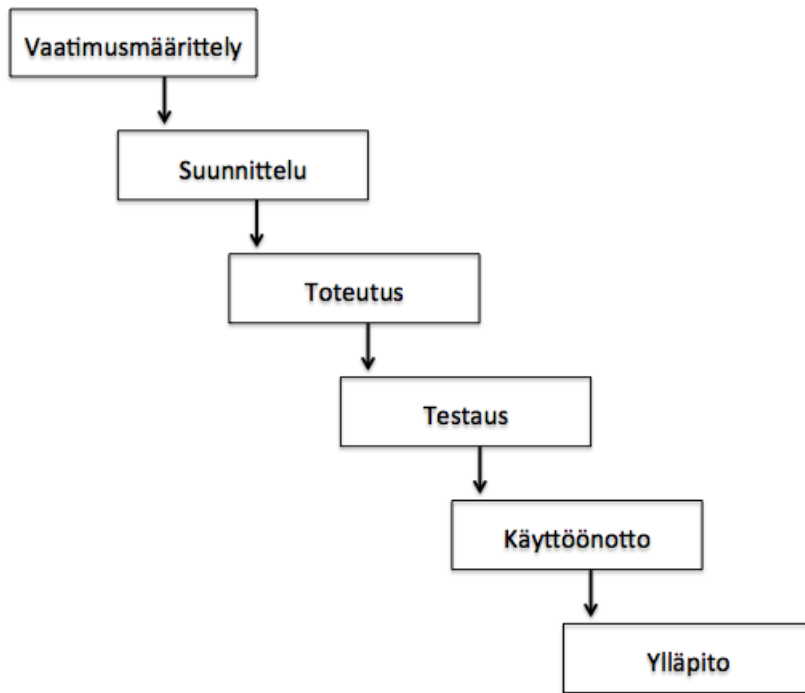
Stephensin [81, s. 270] mukaan vesiputousmalli on perinteinen ohjelmistotuotannon prosessimalli, joka sisältää vaatimusmäärittelyn, suunnittelun, toteutuksen, testauksen, käyttöönoton ja ylläpidon. Vesiputousmalli etenee vaiheittain edellyttäen, että edellinen vaihe on täysin valmis ennen kuin seuraavaan vaiheeseen voidaan siirtyä. Mallin toimintaperiaatetta voidaan hahmottaa valuvan veden avulla. Vesi valuu mallissa alaspäin ja vaiheistukset toimivat säiliöinä. Kun ensimmäinen säiliö on täyttynyt (vaihe on edennyt loppuun ja säiliö on täynnä tietoa), sen sisältö valuu seuraavaan säiliöön, joka hyötyy siten edellisen vaiheen työstä.

Cotterellin ja Hughesin [19, s. 65] mukaan vesiputousmallissa voidaan tarvittaessa nousta vaiheissa myös ylöspäin täydentämään edellisen vaiheen sisältöä, mutta sen pitäisi olla enemmän poikkeus kuin sääntö. Vesiputousmallin pitäisi siis pääasiassa edetä alaspäin ja vain tarvittaessa palata lyhyesti aikaisempaan vaiheeseen.

Vesiputousmalli toimii parhaiten projekteissa, joissa vaatimukset tiedetään tarkasti ennen ohjelmoinnin aloittamista. Vaatimukseen ei sisälly korkeita riskejä ja vaatimukset eivät muutu paljon projektin aikana. Projektiryhmällä tulee olla kokemusta vastaavista projekteista ja aikaresurssia tulee olla riittävästi vaiheiden edistämiseen yksi kerrallaan. [81, s. 271] Cotterellin ja Hughesin [19, s. 65] mukaan malli voi olla mieluisa myös laajoissa ohjelmistoprojekteissa, joissa halutaan välttää prosessissa takaisinpäin palaaminen ja lisätyön tekeminen.

Mohapatran [59, s. 25] mukaan vesiputousmalli on tuonut aikanaan ohjelmistokehitykselle tarvitun prosessimallin ja kurinalaisuutta. Vesiputousmallia on kuitenkin kritisoitu sen jäykkyydestä erityisesti prosessivaiheiden välillä siirryttäessä. Edellisen vaiheen tulokset tulee lukita ennen kuin seuraavaan vaiheeseen on mahdollista siirtyä. Vesiputousmallissa vaatimusten kartoittaminen on kriittisessä roolissa ohjelmiston onnistumisen kannalta. Loppukäyttäjä näkee ohjelmiston yleensä ensimmäistä kertaa vasta, kun ohjelmistoa ollaan ottamassa käyttöön. Vesiputousmalli nojautuu vahvasti dokumentaation tärkeyteen, joka tekee mallista myös haastavan ylläpitää etenkin laajoissa ohjelmistoprojekteissa.

Kuvassa 2.1 esitetään vesiputousmalli Stephensin [81, s. 270] mukaan alaspäin etenevänä prosessina. Ensimmäisessä vaiheessa suoritetaan vaatimusmäärittely, joka voidaan toteuttaa aikaisemmin tässä työssä esitetyn alaluvun 2.2.1 vaatimusten kartoittamisen menetelmiä hyödyntäen. Ohjelmiston loppukäyttäjät osallistuvat

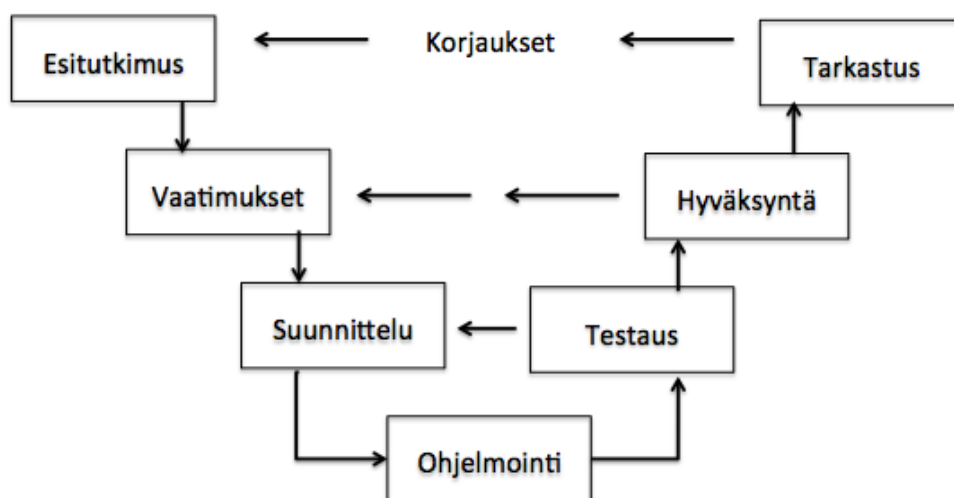


Kuva 2.1: Vesiputousmallissa edetään vaiheittain [81, s. 270].

vesiputousmallissa alkuvaiheessa vain vaatimusmäärittelytyöhön. Vesiputousmalli etenee kuvan 2.1 mukaisesti vaatimusmäärittelyn jälkeen suunnitteluun ja toteutukseen, joissa osallisena on ohjelmiston kehitystiimi. Ohjelmisto toteutetaan kerralla valmiiksi, jonka jälkeen mallissa siirrytään neljänteen vaiheeseen, testaukseen. Ohjelmiston loppukäyttäjät suorittavat testauksen, jonka jälkeen mallin viimeiset vaiheet ovat käyttöönotto ja ylläpito. Ohjelmisto käynnistetään sen toimintaympäristössä, jolloin kaikki sidosryhmät pääsevät viimeistään käyttämään uutta ohjelmistoa. Ohjelmiston ylläpitovastuu voi siirtyä asiakkaalle, tai tapauksesta riippuen ylläpitosisopimus voidaan solmia myös toimittajan ja asiakkaan välille.

2.4.2 V-malli

Cotterellin ja Hughesin [19, ss. 66–67] mukaan V-malli on kehittynyt vesiputousmallista ja korostaa erityisesti validoinnin tärkeyttä. V-malli ikään kuin laajentaa vesiputousmallin testausvaihetta siten, että jokaisesta prosessin vaiheesta voidaan palata prosessissa taaksepäin suorittamaan uudelleen edellisiä vaiheita. Prosessissa takaisin päin pitäisi siirtyä vain niissä tapauksissa, kun projektin etenemisessä huomataan ristiriitoja. Esimerkiksi kehittäjä on voinut tulkita vaatimuksen väärin ja oh-



Kuva 2.2: Vesiputousmallista kehittynyt V-malli [19, s. 66].

jelmiston ominaisuus on toteutettu virheellisesti. Testausvaiheessa, kun virheellinen toteutus havaitaan, voidaan siirtyä V-mallin mukaisesti mallin toisella reunalla sijaitsevaan suunnitteluvaiheeseen takaisin. Prosessimallissa ei ole sallittua tehdä siirtymiä takaisin päin siksi, että olisi mukava suunnitella ohjelmistoa uudelleen, koska silloin prosessimalli ei ole enää V-malli vaan evoluutiokehitysmalli.

Kuvassa 2.2 havainnollistetaan Cotterellin ja Hughesin [19, s. 66] mukaan V-mallin vaiheet. V-malli etenee V-kirjaimen muotoisesti ensin ylhäältä vasemmalta alas ja sen jälkeen ylös oikealle. Kuvan osoittamien vaakasuunnassa kulkevien nuolien mukaisesti jokaisesta oikean reunan vaiheesta voidaan palata takaisin samalla tasolla olevaan vasemman puoleiseen vaiheeseen. Mallin mukaan ohjelmistoprojekti aloitetaan esitutkimuksella, jonka jälkeen edetään vaatimusmäärittelytyöhön. Vaatimusmäärittelytyö voidaan toteuttaa aikaisemmin tässä työssä esitetyn alaluvun 2.2.1 vaatimusten kartoittamisen menetelmiä hyödyntäen.

Cotterellin ja Hughesin [19, s. 66] mukaan vaatimusten kartoittamisen jälkeen ohjelmisto suunnitellaan ja toteutetaan. Toteutuksen eli ohjelmoinnin jälkeen mallissa aletaan nousta alhaalta takaisin ylös mallin oikeaa reunaa pitkin. Ensimmäisenä oikeassa reunassa suoritetaan ohjelmiston testaus yhdessä loppukäyttäjien kanssa. Testauksen jälkeen V-mallissa voidaan keskustella tarpeesta palata prosessissa edellisiin vaiheisiin, mikäli puutteita vaatimuksissa tai ohjelmistossa havaitaan. Testauksesta on mahdollista siirtyä mallin vasemmassa reunassa samalla tasolla olevaan vaiheeseen eli suunnitteluun. Testauksen jälkeen toiseksi viimeisenä vaiheena

on hyväksyntä, josta voidaan tarvittaessa palata uudelleen vaatimusten kartoittamiseen. V-mallin viimeinen vaihe on tarkastus, josta voidaan palata prosessin ensimmäiseen vaiheeseen, esitutkimukseen.

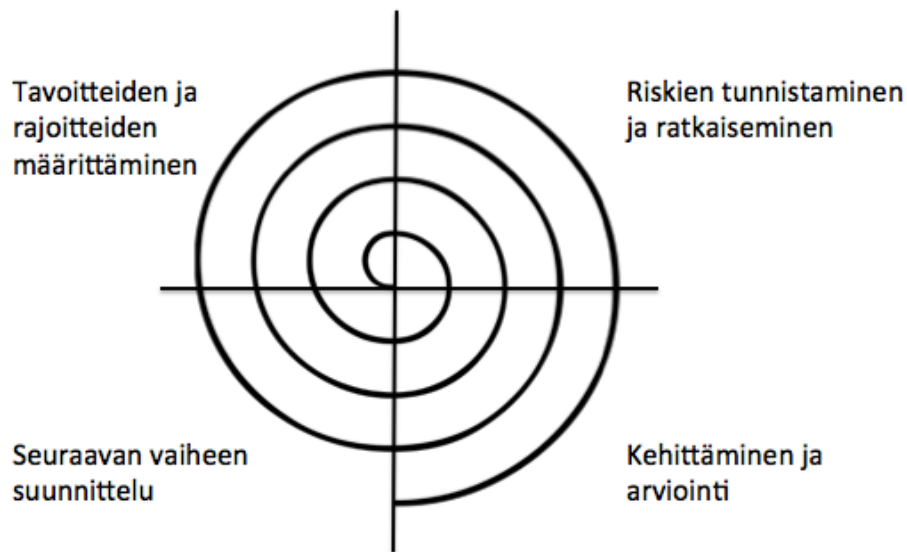
Cotterell ja Hughes [19, s. 66] toteavat, että jos V-mallissa palataan aikaisempiin vaiheisiin, tulee sen jälkeen kulkea kaikki vaiheet uudelleen läpi. V-mallissa siirtymiä takaisin vasemman reunan vaiheisiin tulisi sallia vain erityistapauksissa. Ohjelmistokehitys päättyy mallin mukaan, kun tarkastusvaihe on suoritettu hyväksytysti.

2.4.3 Spiraalimalli

Mohapatran [59, ss. 31–32] mukaan spiraalimallin on esitelty ensimmäistä kertaa Boehm [13] vuonna 1988. Spiraalimalli on saanut vaikutteita vesiputousmallista, inkrementaalista mallista ja kehittyvästä prototyypimallista. Spiraalimallin ominaisuuksiin kuuluu prosessin muodostaminen spiraalin muotoon, joka etenee itsensä ympäri myötäpäivään kulkien. Spiraali jakautuu neljään osaan, jokainen spiraalin neljäsosa sisältää osan ohjelmistoprosessista. Neljä kierroksittain toistuvaa vaihetta ovat tavoitteiden ja rajoitteiden määrittäminen, riskien tunnistaminen ja ratkaiseminen, kehittäminen ja arviointi ja seuraavan vaiheen suunnittelu. Spiraalimalli ei tunne prosessissa takaisin päin palaamista, vaan spiraali kehittyy ja kasvaa jatkuvasti uusina kierroksina. Kierrosten määrä vaihtelee projektin tyypistä ja organisaatiosta riippuen. Cotterellin ja Hughesin [19, s. 67] mukaan ohjelmisto tarkentuu jokaisella kierroksella ja varmuus ohjelmiston onnistumisesta täsmentyy.

Spiraalin säde ilmaisee projektin kustannuksia ja kierrosten lukumäärä edustaa ohjelmistokehityksen vaihetta. Spiraalimallin erityisominaisuus on riskien tunnistaminen ja ratkaiseminen. Ohjelmiston kehittämiseen tarvittavien kierrosten määrä riippuu myös riskien hallinnasta. Ohjelmistoprosessi on mahdollista viedä läpi vain yhdellä spiraalin kierteellä, jos kehitettävä ohjelmisto on helposti ymmärrettävä, vaatimukset eivät muutu ja riskien määrä on pieni. [59, s. 32]

Kuvassa 2.3 havainnollistetaan Mohapatran [59, s. 31] mukaan spiraalimallin vaihteita. Mallin mukainen spiraali käynnistyy kuvion keskeltä, jolloin ensimmäisenä vaiheena suoritetaan ohjelmiston tavoitteiden ja rajoitteiden määrittäminen. Spiraalin seuraavassa vaiheessa suoritetaan riskienhallintaa, mahdolliset riskit tunnistetaan ja niihin tuotetaan ratkaisut. Spiraali etenee riskienhallinnan jälkeen ohjelmiston kehittämiseen. Spiraalin viimeisessä neljänneksessä arvioidaan, jatketaanko ohjelmiston kehittämistä edelleen ja suunnitellaan seuraavaan vaiheeseen siirtymi-



Kuva 2.3: Spiraalimalli sisältää samat vaiheet kierroksittain [59, s. 31].

nen. Spiraali sisältää vähintään yhden täyden kierroksen, mutta kierroksia on yleensä enemmän mallin nimen mukaisesti. Jokainen kierros lisää yhden kierroksen spiraalin ulkoreunaan ja sisältää aina edellä mainitut neljä vaihetta peräkkäisessä järjestyksessä.

2.4.4 Prototyypimalli

Mohapatran [59, s. 27] ja Dorfmanin ja Thayerin [24, s. 9] mukaan prototyypimalli perustuu kokeelliseen prosessiin, jossa toimiva ohjelmiston prototyyppi annetaan loppukäyttäjälle testattavaksi ja kommentoitavaksi. Prototyypin hyödyntäminen antaa käyttäjälle mahdollisuuden ymmärtää ohjelmiston tarjoamia toiminnallisuuksia ja siten helpottaa vaatimusten ilmaisemista konkreettisemmin. Dorfman ja Thayer [24, s. 9] toteavat, että toimivia prototyyppijä tehdään yleensä useampi kappaale. Prototyypin avulla kerätyt vaatimukset dokumentoidaan ja analysoidaan. Dokumentoinnin jälkeen voidaan päättää jatketaanko ohjelmistokehitystä edelleen prototyypimallin kehittämisellä kohti lopullista tuotantoversiota.

Mohapatran [59, s. 27] mukaan prototyypit voivat olla kahden tyyppisiä, nopeasti pois heitettäviä prototyyppijä tai kehittyviä prototyyppijä. Pois heitettävän mallin ajatuksena on tuottaa nopea ja väliaikainen malli vaatimusten kartoittamista varten. Prototyypin kehittämistä tai käyttöä ei jatketa lainkaan enää vaatimus-

ten kartoittamisen jälkeen, vaan prosessimalli etenee sen jälkeen vesiputousmallin kaltaisesti lopullisen ohjelmiston kehitykseen. Kehittyvän prototyypin tapauksessa mallia ei hylätä vaatimusten kartoittamisen jälkeen, vaan sen kehittämistä jatketaan kohti ohjelmiston lopullista versiota.

Molempien prototyypin menetelmien tavoitteena on tunnistaa vaatimukset, jotka muuten saattaisivat jäädä kartoittamatta. Pois heitettävän prototyypin haasteena on sen soveltumattomuus ei-toiminnallisten vaatimusten testaamiseen. Prototyyppi on aina arvaus lopullisesta ohjelmistosta, jonka vuoksi loppukäyttäjät voivat saada myös vääristyneen kuvan siitä, mitä projektissa ollaan kehittämässä. Kehittyvän prototyypin haasteena on erilaisten versioiden testaaminen, joka johtaa usein huonoon rakenteeseen ja ylläpidettävyyteen ohjelmistossa. [59, s. 28] Cotterell ja Hughes [19, s. 68] esittävät kehittyvän prototyypin olevan prototyypimallin toinen suuntaus, mutta he eivät pidä mallia todellisena prototyypinä. Perusajatus prototyypistä suunnittelun ja vaatimusten kartoittamisen apuna ei pidä enää paikkansa, kun prototyyppiä kehitetään kohti lopullista tuotantoversiota ohjelmistosta.

Mohapatran [59, ss. 29–30] mukaan prototyypin kehittämiseen voi hyödyntää seuraavia ohjeita:

- Prototyypin tavoitteiden tulee olla selkeitä, jotta loppukäyttäjät ovat tietoisia niistä. Tavoitteita voivat olla käyttöliittymän kehittäminen, toiminnallisten vaatimusten validointi tai jonkin vastaavan yksityiskohtaisen tiedon kerääminen.
- Prototyypin kehittäminen aiheuttaa kustannuksia. Kustannusten hallitsemisen vuoksi prototyyppiin kannattaa kehittää samoja ominaisuuksia, joita lopulliseen ohjelmistoon kehitetään. Prototyypin kehittämisessä ei ole tarpeellista huomioida ei-toiminnallisia vaatimuksia tai virheiden käsittelyä.
- Kehittäjien tulee hyödyntää ohjelmointikieliä ja ohjelmistoja, jotka mahdollistavat prototyypin kehittämisen nopeasti ja pienillä kustannuksilla.

Cotterellin ja Hughesin [19, s. 68] mukaan prototyyppien hyödyntäminen lisää vuorovaikutusta sidosryhmien ja vaatimusten kartoittajan välillä. Sidosryhmien edustajat saavat prototyypin näkemisestä paremman kuvan ohjelmiston tarjoamista mahdollisuuksista kuin lukemalla projektin dokumentaatiota. Prototyyppien hyödyntäminen edistää tekemällä oppimista. Prototyyppien virheistä voidaan oppia jo saman projektin aikana ja samalla lisätä tietämystä organisaation tietopankkiin.

2.5 Ketterät menetelmät

Larman ja Basilin [54] mukaan ketterien menetelmien mukaista ajattelua on ollut olemassa jo ainakin 1970-luvulta alkaen. Aikaisimmat viitteet iteratiivisen ja inkrementaalisen projektimallin käytöstä on vuodelta 1957. Ketteriä menetelmiä tutkittiin 70-luvulla perinteisen vesiputousmallin vastakohtana. Vaihtoehtoisia ohjelmistokehityksen malleja ei otettu vielä silloin riittävällä vakavuudella vastaan, jonka vuoksi perinteinen vesiputousmalli säilytti asemansa vielä vuosituhanen loppuun saakka. Abbasin ja muiden [2] mukaan ketterät menetelmät yleistyivät merkittävästi 2000-luvun alusta lähtien.

2.5.1 Taustaa ketteristä menetelmistä

Abbasin ja muiden [2] mukaan ketteristä menetelmistä käytetään myös nimitystä agile. Agile on määritelty tarkoittavan muun muassa tehokasta ja ketterää. Agilen ohjelmistokehityksen tulee olla kevyt ja riittävä. Keveydellä tarkoitetaan prosessimallin kykyä pysyä ketteränä. Riittävyys puolestaan tarkoittaa kykyä tuottaa riittävä lopputulos, joka ei ole alle tai yli tavoitteiden. Ketteristä menetelmistä on vaikea luoda tarkkaa kuvausta, koska se sisältää useita hieman toisistaan poikkeavia ohjelmistokehityksen prosessimalleja. Abbasin ja muiden [2] mukaan ketterille prosessimalleille yleisesti tyypillinen toimintatapa on kevyt iteratiivinen lyhyissä aikasykleissä tapahtuva kehittäminen. Aktiivinen käyttäjien osallistaminen, vaatimusten priorisointi ja vaatimusten tarkastaminen kuuluu toimintatapaan. Ketterät menetelmät korostavat tiimin osaamista ja tietotaitoa enemmän kuin dokumentaation tärkeyttä.

Sillitti ja Succi [75, s. 310] kuvaavat ketterien menetelmien olevan joukko ohjelmistokehityksen prosesseja, jotka pyrkivät tuottamaan toimivan ohjelmiston nopeasti, laadukkaasti ja asiakkaan tarpeet täyttäen. Batoolin ja muiden [8] mukaan menetelmät pyrkivät tuottamaan toimivan ohjelmiston lyhyessä ajassa tekemällä läheistä yhteistyötä kehitystiimin ja sidosryhmien edustajien kanssa. Kasvokkain tapahtuva yhteistyö koetaan arvokkaaksi tavaksi tiedonvälitykseen asiakkaalta ohjelmistokehittäjille. Projektin onnistumista mitataan ohjelmiston toimivuudella.

Rachevan ja muiden [67] mukaan vaatimusten muuttuminen on tiedostettua ja jopa toivottua ketterissä menetelmissä. Vaatimusten sallitaan muuttuvan projektin aikana, koska se yleensä pienentää kustannuksia verrattuna käyttöönnoton jälkeisiin muutoksiin. Vaatimusten kehittämisessä otetaan aina huomioon asiakastyty-

väisyys. Ketterät menetelmät pyrkivät aina kaikessa toiminnassa kohti asiakkaan ongelman ratkaisemista. Tärkeintä ketterissä menetelmissä on jatkuva vaatimusten priorisointi asiakkaan näkökulmasta. Tämän avulla asiakkaalle tuotettava hyöty voidaan maksimoida. Ketterät menetelmät eroavat perinteisestä prosessimallista vaatimusten priorisoinnin osalta siten, että priorisointi tehdään uudelleen jokaisen iteraatiokierroksen alussa. Priorisointi perustuu aina asiakkaan liiketoimintatavoitteiden saavuttamiseen. Tärkeimmät ominaisuudet toteutetaan ensin huomioiden kuitenkin riskien pitäminen mahdollisimman alhaisina.

Rachevan ja muiden [67] mukaan ketterissä menetelmissä riskien hallinnassa keskitytään vähentämään epävarmuutta ja jo toteutettujen vaatimusten muuttamista myöhemmin. Riskejä hallitaan uudelleenpriorisoinnin avulla jokaisen iteraatiokierroksen alussa. Epävarmojen vaatimusten toteuttamista pyritään välttämään mahdollisimman pitkään, jotta muutoksia toteutukseen ei myöhemmässä vaiheessa tarvitsisi tehdä.

Sillittin ja Succin [75, s. 310] mukaan ketteriin menetelmiin kuuluu useita hyvin erilaisia prosessimalleja, joista suosituimpia ovat:

- eXtreme Programming (XP)
- Scrum
- Dynamic Systems Development Method (DSDM)
- Adaptive Software Development (ASD)
- The Crystal Family

Kaikissa ketterien menetelmien prosessimalleissa yhdistyy joukko perusarvoja, jotka on määritelty Agile manifestossa [75, s. 310]:

1. Yksilöitä ja vuorovaikutusta enemmän kuin prosesseja ja työkaluja: Ketterät menetelmät korostavat enemmän ihmisten välistä vuorovaikutusta prosessien ja työkalujen sijaan.
2. Asiakasyhteistyötä enemmän kuin sopimuksia: Kehitystiimin ja asiakkaan välistä suhdetta ylläpidetään osallistamalla asiakasta kehitystyöhön sen sijaan, että suhdetta hallittaisiin sopimuksilla. Yleensä ketterissä ohjelmistoprojekteissa projektin hinta määräytyy joustavasti kiinteän sopimushinnan sijasta.

3. Toimivaa sovellusta enemmän kuin dokumentaatiota: Kehitystiimin tavoitteena on tuottaa ja toimittaa asiakkaalle läpi projektin toimivaa koodia ja toimivia ohjelmiston osia. Hyvin kirjoitettu koodi on jo itsessään dokumentoitu, jolloin muodollinen dokumentaatio vähenee.
4. Muutokseen vastaamista enemmän kuin suunnittelua: Kehitystiimin on vastattava nopeasti muuttuviin vaatimuksiin ja rajoittavien päätösten tekemistä pyritään välttämään mahdollisimman pitkään. Suunnitteluun käytettävä aika pyritään rajaamaan minimiin ja tulevaisuuden tarpeiden pohdintaa pyritään välttämään.

Sillittin ja Succin [75, s. 311] mukaan ketterän ohjelmistokehityksen manifestin arvoista on tunnistettu yhteisiä käytänteitä, jotka pohjautuvat onnistuneisiin ja epäonnistuneisiin ketteriä menetelmiä hyödyntäneisiin ohjelmistoprojekteihin. Ketterän ohjelmistokehityksen prosessimalli tulee valita projektin luonteen mukaan, siten, että malli sopii sekä asiakkaalle että kehitystiimille. Kaikki ketterän kehityksen mallit toteuttavat iteratiivista sykleittäin tapahtuvaa kehitystä. Jokainen sykli sisältää vaiheina analysoinnin, suunnittelun, ohjelmoinnin ja testaamisen. Syklit kestävät ajallisesti kahdesta viikosta kahteen kuukauteen, joiden aikana keskitytään vain valittujen kehittämiskohteiden toteuttamiseen. Ohjelmiston vaatimukset priorisoidaan jokaisen syklin alussa, jolloin vaatimuslistalta valitaan tärkeimmät vaatimukset toteutettavaksi seuraavan syklin ajaksi. Jokaisen syklin päätyttyä valmis ohjelmiston osa toimitetaan asiakkaalle, joka testaa ja antaa palautteen toteutuksesta. Tämä menetelmä tuottaa hyötyä erityisesti asiakkaalle, joka pääsee testaamaan ohjelmistoa jo varhaisessa vaiheessa projektia ja tuomaan esiin mahdolliset ongelmat ja uudet vaatimukset. Projektin resurssien hallinta pysyy paremmin hallinnassa, kun muutospyynnöt tuodaan kehitystiimin tietoon projektin aikana. Näkyvän työn ansiosta asiakas saa myös tunteen projektin etenemisestä ja siitä, että hänen tarpeisiinsa vastataan. Jatkuva vuorovaikutus kehitystiimin ja asiakkaan välillä lisää luottamusta molempien osapuolten välillä. Asiakas voi olla myös osa kehitystiimiä, jolloin asiakasta voidaan konsultoida nopeasti aina tarvittaessa.

Ketterän ohjelmistokehityksen manifesti [9] toteaa asiakkaan aktiivisen osallistumisen olevan ehdotonta projektin onnistumisen kannalta. Ketterän ohjelmistoprojektin toimittaja tarjoaa yleensä projektiryhmälle sähköisen työskentelyalustan projektin yhteydenpidon sujuvuuden tukemiseksi. Sykliä aikana tapahtuva ohjelmiston testaaminen ja palautteen kerääminen pohjautuu sitoutuneisuuteen projektin

päämäärää ja toisia projektiryhmän jäseniä kohtaan. Luottamuksen puute tai sanomattomat erimielisyydet voivat aiheuttaa konflikteja projektiryhmän jäsenten välillä.

Sillittin ja Succin [75, ss. 313–314] mukaan ketterissä menetelmissä kehitystiimien koko pyritään pitämään pienenä. Projektin ketteryys on usein sidoksissa kehitystiimin kokoon. Suora vuorovaikutus ja dokumentaation minimoiminen onnistuu vain pienissä tiimeissä. Tiimin kasvaessa dokumentaation määrä kasvaa ja vuorovaikutus vähenee. Dokumentaation rooli kasvaa, kun kehitystiimin suora vuorovaikutus vähenee. Sillitti ja Succi [75, s. 314] ovat sitä mieltä, että ketterät menetelmät eivät sovellu laajoihin ohjelmistoprojekteihin, vaan niitä tulisi hyödyntää rajallisen skaalautuvuuden vuoksi ainoastaan pienissä ja keskisuurissa ohjelmistoprojekteissa. Ketterät menetelmät soveltuvat erityisesti ohjelmistoprojekteihin, joiden tavoitetilaa ei ole rajattu liian tiukasti, ja jotka ovat kooltaan riittävän pieniä.

Paetsch ja muut [63] toteavat artikkelissaan, että ketterä ohjelmistokehitys ja sen erilaiset suuntautumiset ovat kasvattaneet suosiotaan asiakaslähtöisen periaatteen ansiosta. Sillitti ja Succi [75, s. 314] toteavat myös ketterien menetelmien tavoitteena olevan ihmisten pitäminen keskiössä ongelmien ratkaisemisessa ja tiedon jakamisessa. Ketterien menetelmien hyödyntäminen tuloksellisesti vaatii kehitystiimiin vankkaa osaamista tiimityöskentelystä ja vuorovaikutustaidoista kehitystiimin ja asiakkaan välillä.

Sillittin ja Succin [75, s. 314] mukaan ketterät menetelmät keskittyvät tuottamaan vain sellaista lopputulosta, joka tuottaa hyötyä asiakkaalle. Ketterien menetelmien tavoitteena ei ole tuottaa esimerkiksi uudelleen hyödynnettäviä ominaisuuksia, koska silloin projektiin jouduttaisiin sisällyttämään vaatimuksia, jotka eivät palvele kyseisen projektin tavoitteita. Ketterät menetelmät eivät siis sovellu kaikkiin ohjelmistoprojekteihin. Soveltumattomuus korostuu erityisesti turvallisuuden kannalta kriittisissä tai laajoissa ohjelmistoprojekteissa.

Sillittin ja Succin [75, s. 315] mukaan vaatimukset ovat kaikkien ohjelmistojen perusta. Vaatimusten kartoittaminen, hallinta ja ymmärtäminen ovat yleisesti tunnistettuja ongelmia kaikissa ohjelmistokehityksen prosessimalleissa. Perinteinen prosessimalli hyödyntää standardeja IEEE Standard 1233: ”Guide for Developing System Requirements Specifications” [42] ja IEEE Standard 830: ”Recommended Practice for Software Requirements Specifications” [44].

Ketterät menetelmät eivät nojaudu edellä mainittuihin standardeihin vaatimusten kartoittamisen ja hallinnan osalta, mutta hyödyntävät kyllä standardien perusa-

jatusta. Ketterissä menetelmissä esimerkiksi koko kehitystiimi osallistuu vaatimusten kartoittamiseen, kun perinteisessä prosessimallissa vaatimusten kartoittamiseen osallistuu vain rajattu osa projektiryhmästä.

Paetschin ja muiden [63] mukaan vaatimukset kartoitetaan ketterässä ohjelmistokehityksen mallissa samoilla tavoilla kuin perinteisessä vaatimusmäärittelytyössä. Apuna vaatimusten kartoittamisessa käytetään haastatteluja, aivoriisiä, priorisointia ja käyttötapauskuvausten kaltaisia käyttäjäkertomuksia (user story), jotka ovat kuvauksia jostakin ohjelmiston ominaisuudesta. Kertomukset kuvaavat erityisesti ominaisuuksia, jotka tuovat asiakkaalle lisäarvoa. Perinteiset käyttötapauskuvaukset ovat sen sijaan vain kuvauksia käyttäjän ja ohjelmiston välisestä vuorovaikutuksesta. Ennen käyttäjäkertomusten luontia asiakkaan on mietittävä omia odotuksiaan tulevaa ohjelmistoa kohtaan. Odotusten pohtiminen voi olla myös aivoriisiä, jonka tuloksena tarinakortit syntyvät. Lopuksi tarinakortit analysoidaan, priorisoidaan ja validoidaan.

Paetschin ja muiden [63] mukaan kaikki vaatimukset kirjataan priorisoituina ohjelmiston kehitysdokumenttiin. Dokumentti sisältää tulevan ohjelmiston kaikki vaaditut ominaisuudet, toiminnallisuudet ja kehityksen aikana havaitut puutteet. Kehitysdokumentista valitaan tärkeimmäksi priorisoidut vaatimukset seuraavaan kehityssykliin. Vaatimukset pysyvät muuttumattomina syklin ajan, mutta niitä voidaan muokata jälleen seuraavaa sykliä varten. Syklin päätyttyä toteutettu ohjelmiston osa esitellään projektiryhmälle, joka antaa kommentit seuraavaa kierrosta varten. Menetelmän toimivuuden edellytyksenä on avoin vuorovaikutus projektin jäsenten ja kehittäjien kesken. Avaintekijöitä on luottamus, läpinäkyvyys ja rehellisyys. Ketterän ohjelmistokehityksen manifestin [9] mukaan asiakkaan aktiivinen osallistuminen läpi projektin on välttämätöntä sen onnistumisen kannalta. Asiakkaan on helppo antaa välitöntä palautetta heti, kun ohjelmistosta on toimitettu toimiva versio. Saadun palautteen ja muutospyyntöjen perusteella ohjelmistoa kehitetään edelleen, jonka jälkeen toimitetaan uusi versio kommentoitavaksi.

2.5.2 XP - eXtreme Programming

Measeyn [57, s. 125] mukaan eXtreme programming (XP) kehitettiin 1990-luvun lopussa uutena prosessimallina helpottamaan muuttuvien vaatimusten hallitsemista jättyneiden mallien rinnalle. XP kehittyi nopeasti hallitsevaksi ketterän menetelmän malliksi, kunnes Scrum otti sen aseman 2000-luvun alussa. XP:n arvoissa, periaatteissa ja käytännöissä on paljon yhtäläisyyksiä ketterän ohjelmistokehityksen mani-

festiin. XP:n arvoja ovat vuorovaikutus, palaute, yksinkertaisuus, kunnioitus ja rohkeus. Lankhorst [53, s. 19] korostaa XP:ssä asiakastyytyväisyyttä ja tiimityöskentelyä. Asiakastyytyväisyys saavutetaan iteratiivisen työskentelytavan avulla, XP tuottaa toimivia ohjelmiston osia tiheästi jo hyvin aikaisessa vaiheessa projektia.

Kochin [50, s. 245] ja Measeyn [57, ss. 128–129] mukaan XP-projekti aloitetaan suunnittelulla, jossa asiakas muuntaa vaatimuksensa käyttäjäkertomuksiksi. Käyttäjäkertomukset ovat XP:ssä määritelty hyvin lyhyiksi, nyrkkisääntönä voidaan pitää, että kertomuksen pitäisi mahtua kirjoitettuna 8×13 cm kokoiselle kortille. Kehitystiimi arvioi käyttäjäkertomusten kehittämisaika-arvion, jonka jälkeen asiakas suhteuttaa resurssiarviot omiin liiketoimintatavoitteisiin ja määrittää seuraavan iteraatiokierroksen tavoitteet. XP:n tavoite on tuottaa toimiva ohjelmiston osa jokaisen iteraatiokierroksen päätyttyä. Jokaisesta iteraatiokierroksesta saatu palaute ohjaa sekä kehitystiimiä että asiakasta tekemään tarvittavia muutoksia seuraavaan iteraatiokierrokseen. Asiakas voi määrittää iteraatiokierrosten välillä myös uusia käyttäjäkertomuksia, joille arvioidaan kehittämisaika. Kochin [50, s. 246] mukaan XP:ssä käytetään metaforaa eli kuvausta kehitettävästä ohjelmistosta. Metafora pyrkii kuvaamaan kehitettävän ohjelmiston helposti ymmärrettävällä kielellä, jota sekä asiakas että kehittäjät pystyvät hyödyntämään. Kuvaus on korkean tason visio ohjelmiston kaikista vaatimuksista, ja sen tulisi pysyä suhteellisen muuttumattomana koko projektin ajan.

Measeyn [57, s. 130] mukaan käyttäjäkertomukset toimivat iteraatioiden päätyttyä myös testauksen tukena. Asiakas voi testata kuvailemiensa kertomusten vastaavuutta kehitetyssä ohjelmiston osassa. Asiakas saa testauksen myötä varmuuden ohjelmistoprojektin etenemisestä aikataulussa, mikäli testaus todetaan onnistuneeksi. Kochin [50, s. 247] mukaan ulkoisen testauksen lisäksi myös kehittäjät testaavat koodia ja kehittävät sitä iteraatioiden myötä. Kehittäjät refaktoroivat koodia eli parantavat ja yksinkertaistavat koodia siten, että sen sisäinen rakenne paranee, mutta toiminnallisuudet säilyvät ennallaan.

Measeyn [57, s. 131] mukaan XP pyrkii edistämään kehittäjien luovuutta ja tuottavuutta. Yleensä viikkotyöaika XP-projekteissa on rajattu 40 tuntiin. Ylitöiden tekemistä ei suositella, joskaan ylitöitä ei myös kielletä. XP:ssä on kuitenkin ohjeistus, jonka mukaan ylitöitä ei saisi tehdä viikkoa pidempään. Ylitöiden tarve kertoo myös iteraatiokierroksen työmääräarvion pettämisestä tai muista työtä haittaavista ongelmista. Ongelmista viestiminen muulle kehitystiimille iteraatiokierroksen aikana pitäisi tehdä mahdollisimman aikaisessa vaiheessa. Kochin [50, s. 245] mukaan

jokainen iteraatiokierros kestää muutaman viikon.

Ally ja muut [4] ja Canfora ja muut [16] esittävät pariohjelmoinnin olevan XP:n erikoisuus. Pariohjelmointi tarkoittaa kahden ohjelmoijan työskentelyä tiiviissä yhteistyössä yhteisen ongelman ratkaisemiseksi. Tavanomaiseen yksilöohjelmointiin verrattuna XP:ssä tuotettu koodi on tehty aina yhteistyössä kahden ohjelmoijan kesken, jopa samaa työasemaa ja tietokonetta hyödyntäen. Canfora ja muut [16] esittävät pariohjelmoinnin olevan tehokasta, koska parityöskentelyssä kehittäjät joutuvat tulemaan pois omalta mukavuusalueeltaan ja pohtimaan ratkaisuja yhdessä. Jatkuva koodin tarkastaminen vähentää virheiden määrää ja laskee siten ohjelmointityön kustannuksia. Parityöskentely parantaa lisäksi osaamisen jakamista kehittäjien kesken ja korostaa vuorovaikutuksen tärkeyttä.

Pareittain työskentelevillä kehittäjillä on erilaiset, mutta samanarvoiset roolit, jotka vaihtuvat sopivin väliajoin. Ensimmäinen kehittäjä hallitsee tietokoneen syöttölaitteita, hiirtä, näppäimistöä ja muita laitteita kirjoittaakseen koodia ja testataksena sitä. Toinen kehittäjä seuraa ohjelmoijan työtä ja havainnoi kehittyvää koodia, kysyy tarkentavia kysymyksiä, miettii vaihtoehtoisia toteutustapoja ja suhtautuu työhön kriittisesti. Hyvän yhteistyön onnistumiseksi kehittäjäparien tulee olla tietyllä tapaa samankaltaisia ajattelijoina ja sosiaalisesti virittäytyneitä. [4]

Kochin [50, ss. 247–248] mukaan XP:n toimintatapaan liittyy jatkuva integrointi ohjelmiston kehittyessä. Jokaisen iteraatiokierroksen päätteeksi kehittäjäparien tehtävänä on liittää iteraatiossa kehitetty käyttäjäkertomus osaksi ohjelmistoa. Integraation jälkeen ohjelmistoa voidaan testata kokonaisuudessaan ja viimeksi kehitetyn käyttäjäkertomuksen osalta. Jos testauksessa ilmenee ongelmia, kehitysparien tulee ratkaista havaittu ongelma ennen kuin seuraavan käyttäjäkertomuksen toteuttamiseen voidaan edetä.

Racheva ja muut [67] esittävät asiakkaan olevan XP:ssä vastuullinen osapuoli kaikkiin projektiin liittyviin päätöksiin ja siten myös vaatimusten priorisointiin. Kehitystiimi tukee kuitenkin asiakasta priorisoinnissa arvioimalla vaatimusten kustannuksia ja teknisiä riskejä. Vaatimusten tärkeyttä asiakkaan liiketoiminnalle voi melko suoraviivaisesti arvioida miettimällä asiaa kahden kysymyksen avulla: Millaista hyötyä sidosryhmät saavuttavat, mikäli vaatimus toteutetaan? Millaisen menetyksen vaatimuksen toteuttamatta jättäminen aiheuttaa? Tietoturvaan liittyvät vaatimukset tulisi myös käsitellä toteuttamatta jättämisen arvioinnin avulla. Tietoturvaan liittyvän vaatimuksen toteuttamatta jättämisen seurauksia voidaan pohtia, jonka jälkeen pystytään päättämään aiheutuuko toteuttamatta jättämisestä liian suuri

riski ja halutaanko se hyväksyä vai välttää.

2.5.3 Scrum

Lankhorstin [53, s. 20] mukaan Scrum on 2000-luvun alussa kehitetty ketterän ohjelmistokehityksen prosessimalli, joka kasvatti nopeasti suosiotaan käytetyimmäksi ketterän kehityksen malliksi. Scrumissa työskennellään ketterän kehityksen arvojen ja periaatteiden mukaisesti tavoitellen suurinta mahdollista hyötyä asiakkaalle. Scrumissa ohjelmisto toteutetaan iteraatiokierrosten avulla, joita kutsutaan sprinteiksi. Saddington [71, ss. 20–21] toteaa Scrumin olevan projektin sidosryhmille hyödyllinen prosessimalli, koska he saavat jatkuvasti ajantasaista tietoa projektin tilanteesta ja voivat vaikuttaa vaatimusten priorisointiin ja projektin resursseihin.

Schwaber ja Sutherland [73, s. 57] kuvaavat Scrumin yksinkertaiseksi prosessimalliksi, joka sisältää vain kolme erilaista käyttäjäroolia, kolme toimintaan liittyvää ominaisuutta ja viisi toistuvaa tapahtumaa.

- Käyttäjäroolit
 - Tuoteomistaja
 - Scrum-master
 - Kehitystiimi
- Prosessin ominaisuudet
 - Sprinteissä tapahtuva kehitys
 - Tuotteen kehitysjono
 - Sprintin tehtävälista
- Tapahtumat
 - Sprintti
 - Sprintin suunnittelu
 - Päiväpalaveri
 - Sprinttikatselmus
 - Retrospektiivi

Scrumin kehitystiimiin eli Scrum-tiimiin kuuluu henkilö, joka haluaa ohjelmiston kehitettäväksi eli tuoteomistaja (product owner). Tiimin työskentelystä ja johtamisesta vastaa Scrum-master. Loput Scrum-tiimin jäsenistä ovat kehittäjiä. Tuoteomistaja päättää, mitä ohjelmiston ominaisuuksia toteutetaan projektin iteraatioissa eli Scrumin termien mukaisesti sprinteissä. Tuoteomistaja vastaa viime kädessä vaatimusten priorisoinnista ja hän arvioi myös sprinttien päättyessä toteutetun ohjelmistokokonaisuuden osan. Tuoteomistajia voi olla samassa projektissa vain yksi sekaannusten välttämiseksi. Scrum-master vastaa projektin johtamisesta Scrum-mallin mukaisesti. Scrum-master voi olla sertifioitu osaaja, tai hänellä voi olla erityistä osaamista ja kokemusta Scrumin hyödyntämisestä ohjelmistoprojekteissa [73, s. 57]. Saddingtonin [71, s. 18] mukaan Scrum-masterin rooliin kuuluu erityisesti kehittäjien työrauhan takaaminen sprinttien aikana, Scrum-master pyrkii motivoimaan kehittäjiä ja näyttämään hyvää esimerkkiä Scrum-menetelmän hyödyntämisestä. Saddington [71, s. 57] listaa Scrumiin liittyviin rooleihin myös sidosryhmät ja muut ryhmät, jotka liittyvät kehitettävään ohjelmistoon. Saddingtonin mainitsemat ryhmät eivät kuitenkaan varsinaisesti ole Scrum-tiimiä ajatellen aktiivisessa roolissa, vaan Scrum-tiimiin kuuluva tuoteomistaja välittää sidosryhmien vaatimukset tiimin tietoon, ja toisaalta tuoteomistaja välittää myös Scrum-tiimin havaintoja sidosryhmille.

Asiakkaan ja loppukäyttäjien äänenä Scrumissa toimii tuoteomistaja. Tuoteomistaja on jatkuvasti askeleen edellä projektissa, koska hän on vastannut projektin tavoitteen asettamisesta ja vaatimusten kartoittamisesta. Tuoteomistaja vastaa viime kädessä sprinttiin valittavien tehtävien priorisoinnista ja hänellä on kuva seuraavien sprinttien tehtävistä. Koska tuoteomistajan roolissa voi toimia vain yksi henkilö, on rooliin valittavalla henkilöllä oltava selkeä ymmärrys siitä, mitä projektissa ollaan kehittämässä, vahva sitoutuneisuus ketteriä menetelmiä kohtaan, hyvä yhteys sidosryhmiin ja riittävästi luottamusta liiketoimintaan liittyvien ratkaisujen tekemiseen. [71, s. 57]

Projektin alkaessa Scrum-masterin tehtävänä on koota kehitystiimi, jonka osaaminen mahdollistaa ohjelmiston kehitysjonon mukaisten tehtävien toteuttamisen toimivaksi ohjelmistoksi. Scrum-tiimi kokoontuu yhteen sopimaan tiimin työskentelytavasta ja luomaan yhteisen vision ohjelmiston lopputuloksesta. Projektin aloituspalaveri kestää Scrumin nopean työskentelymallin mukaisesti korkeintaan päivän. Järjestäytymisen ja projektin selkeyttämisen jälkeen tiimi priorisoi tuoteomistajan johdolla ensimmäiseen sprinttiin valittavat tärkeimmät vaatimukset ja jakaa ne

sopivan kokoisiksi tehtäviksi. Scrum-tiimin työskentely alkaa välittömästi suunnittelupalaveria seuraavana työpäivänä. Jokainen sprintti tuottaa toimivan ja testattavan osan ohjelmistoa, jonka tulee sisältää vähintään ne vaatimukset, jotka sprintin alussa arvioitiin sprintissä toteutettavan. Mikäli sprintissä edetään nopeasti valittujen tehtävien kanssa, voidaan jäljelle jäänyt aika hyödyntää myös lisätehtävien toteuttamiseen. [73, s. 58]

Saddington [71, ss. 27–28] korostaa vuorovaikutuksen tärkeyttä ketterissä menetelmissä ja Scrumissa. Scrum-masterin tulee huolehtia että kehitystiimissä on avoin ilmapiiri, jossa jokainen kehitystiimin jäsen on yhdenvertainen. Saddington esittää kehitystiimin työskentelyn olevan tehokkainta, kun kehittäjät työskentelevät fyysisesti samassa paikassa, parhaassa tilanteessa jopa kasvokkain. Jokaisella kehittäjällä tulisi olla myös mahdollisuus työskennellä tarvittaessa itsenäisesti rauhallisessa työympäristössä.

Kehitystiimi keskustelee päivittäin päiväpalaverissa sprintin etenemisestä ja mahdollisista havaituista haasteista. Päiväpalaverien tulee olla lyhyitä ja ytimekkäitä, noin 15 minuutin mittaisia tapaamisia. Päiväpalaverissa jokaisen kehittäjän tulisi vastata seuraaviin kysymyksiin: 1. Mitä tein eilen ja mitä eilen valmistui? 2. Mitä teen tänään ja mitä pyrin saamaan valmiiksi? 3. Millaiset esteet voivat rajoittaa työtäni tänään? Jokaisen kehitystiimin jäsenen läsnäolo päiväpalaverissa on tärkeää, jotta tiimissä säilyy ajantasainen ymmärrys sprintin tilanteesta. Sprinttien aikana toteutetaan vain tehtäviä, jotka kehitystiimi ja tuoteomistaja on hyväksynyt sprintin alussa tehtäväksi. Häiriöiden pitäminen mahdollisimman vähäisenä kasvattaa kehittäjien tuottavuutta. [73, ss. 58–59]

Jokaisen sprintin päättyessä tuoteomistaja, scrum-master ja kehittäjät kokoontuvat pitämään sprinttikatselmuksen. Katselmuksessa käydään läpi, miten sprintti onnistui, kuinka paljon valmista saatiin aikaiseksi ja kuinka hyödyllinen sprintin tulos on. Jos kaikkia sprinttiin valittuja tehtäviä ei ehditä saattaa valmiiksi sprintin aikana, ne siirretään takaisin tuotteen kehitysjonoon. Katselmuksessa voi syntyä usein uusia vaatimuksia ja huomioita, jotka kirjataan kehitysjonoon priorisoitavaksi. Sprinttikatselmuksessa voidaan sopia seuraavassa sprintissä kehitettävät tehtävät ja valmistella sprintti. Katselmuksessa voidaan myös todeta, että ohjelmistokehitys lopetetaan. Katselmuksset ovat Scrumille tyypilliseen tapaan nopeita, sprinttikatselmuksen tulisi kestää ajallisesti enintään 4 tuntia. [73, s. 59]

Sprintin retrospektiiveissä Scrum-tiimin kaikki jäsenet pyrkivät kehittämään toimintaansa sprintti sprintiltä tehokkaammaksi. Retrospektiivit järjestetään sprinttien

välillä, jolloin kehitystiimillä on mahdollisuus keskustella mahdollisista aikaisemmin havaituista tiimityöskentelyyn liittyvistä ongelmista, ja erityisesti hakea yhteistä ratkaisua, jonka avulla tilannetta voidaan kehittää seuraavan sprintin aikana. Keskustelussa voidaan keskustella esimerkiksi kehitystiimin osaamisesta, työmääristä, vaatimusten ymmärrettävyydestä, yhteistyöskentelystä ja yleisesti ajatuksista Scrum-mallin hyödyntämisestä. Keskustelun päätteeksi Scrum-tiimi päättää mil-lä tavoin tiimityöskentelyä voidaan seuraavassa sprintissä tehostaa. Scrum etenee sprinteissä, kunnes projektille asetetut tavoitteet on saavutettu. [73, s. 60]

2.6 Perinteisten prosessimallien ja ketterien menetelmien vertailu

Batoolin ja muiden [8] mukaan perinteinen vaatimusmäärittelytyö ja ketterä ohjelmistokehitys pyrkivät samaan lopputulokseen lähes samoilla ominaisuuksilla, mutta erilaisella lähestymistavalla. Perinteinen vaatimusmäärittelytyö luottaa dokumentaation täydellisyyteen koko ohjelmistokehityksen ajan, vaikka usein asiakas haluaisi edetä projektissa joustavasti ja nopealla aikataululla. Ketterä ohjelmistokehitys on kasvattanut suosiotaan asiakasorientoituneella ja joustavalla toimintatavalla. Ketterä menetelmä mahdollistaa ohjelmistopalasten nopean julkaisun ja vaatimusten muuttamisen ja lisäämisen projektin aikana. Lawrencen ja muiden [55] mukaan on helpompaa olettaa, että vaatimukset tulevat muuttumaan ja kehittymään projektin edetessä, kuin että ne pysyisivät muuttumattomina. Sommerville [78] pitää myös vaatimusten muuttumista väistämättömänä, etenkin toimintaympäristöjen ja liiketoiminnan muuttuessa sekä reagoinnissa uusiin mahdollisuuksiin.

Paetschin ja muiden [63] näkemyksen mukaan perinteisen vaatimusmäärittelytyön perustana ja tiedonjakajana on dokumentaatio, kun ketterässä ohjelmistokehityksessä pyritään keskittymään oikeassa elämässä tapahtuvaan keskusteluun ja yhteydenpitoon asiakkaan ja kehittäjien välillä samanlaisten tavoitteiden vahvistamiseksi. Ketterä ohjelmistokehitys on ohjelmointiorientoitunut sen sijaan, että se keskittyisi luomaan runsaasti dokumentaatiota ja pitämään sitä menetelmänsä keskiössä. Ketterä ohjelmistokehitys on mukautuva, mutta ei niin ennustettava kuin perinteinen vaatimusmäärittely. Perinteisessä vaatimusmäärittelytyössä projekti suunnitellaan jo alkuvaiheessa pitkälle aikavälille. Tämä malli sopii projekteihin, joissa vaatimukset pystytään kuvaamaan jo alkuvaiheessa riittävän tarkasti. Projektille asetetun aikataulun ja suunnitelman toteutumisen ehtona on myös se, että ohjelmistoteknologiat ja ohjelmiston käyttötarkoitus on kehittäjillä hyvin tiedossa.

Goguen ja Linden [32] mukaan vaatimusmäärittelyprosessi ei aina ole helposti suoritettavissa. Loppukäyttäjät voivat muuttaa mieltään ymmärtäessään mahdollisuudet paremmin tai he voivat määrittää lisää vaatimuksia vaatimusten kartoittamisen jälkeisissä vaiheissa. Ei ole tavatonta, että käyttäjät eivät tiedä, eivät halua tai eivät pysty ilmaisemaan vaatimuksiaan. Usein ongelmat johtuvat hyvin järkiperaisista syistä. Käyttäjät haluaisivat nähdä demoesimerkkejä siitä, mitä ollaan kehittämässä, tutkia muita mahdollisuuksia tai kehittää uusia ratkaisuja olemassa olevaan ongelmaan. Batoolin ja muiden [8] mukaan ketterä ohjelmistokehitys on osoittautunut menestyksekkääksi, koska hyvän kommunikaation avulla vaatimukset saadaan tuotettua selkeämmin, ja asiakas tietää läpi projektin mitä tehdään ja millainen ohjelmistosta on muokkautumassa. Vaatimusten sallitaan kertyvän projektin aikana ja ne voivat myös muuttua. Ketterät menetelmät tuovat loppukäyttäjälle suurta hyötyä iteratiivisen toimintatavan ansiosta, käyttäjä näkee jo hyvin varhaisessa vaiheessa ensimmäisiä versioista ohjelmistosta, ja pystyy määrittämään sen jälkeen vaatimuksiaan tarkemmin.

Rachevan ja muiden [67] mukaan asiakkaat ja käyttäjät ymmärtävät ketterissä menetelmissä paremmin toteutettavan vaatimuksen laajuuden, vaikuttavuuden ja kustannukset. Hyöty saavutetaan iteratiivisen työskentelyn ja jatkuvan vaatimusten priorisoinnin avulla. Perinteisessä vaatimusmäärittelytyössä priorisointi toteutetaan vain projektin alkuvaiheessa, jonka vuoksi vaatimuksista johtuvia rajoituksia ei välttämättä ymmärretä ja osata ottaa huomioon.

Perinteinen vaatimusmäärittelytyö näyttäytyy työmäärältään asiakkaalle työläänä projektin alkuvaiheessa vaatimusten kartoittamisessa ja analysoinnissa, sekä projektin loppuvaiheessa ohjelmiston testauksessa ja käyttöönotossa. Ketterä menetelmä arvostaa enemmän yhteydenpitoa asiakkaan kanssa, kasvokkain tapahtuvia palavereita pyritään järjestämään usein. Avoin kommunikaatio projektiryhmän ja kehittäjän välillä vähentää väärinymmärryksiä ja edistää ohjelmiston kehitystä. Yhteydenpidon ansiosta asiakas voi olla vakuuttunut, että kehitettävä ohjelmisto on vaatimusten mukainen.

Perinteisen vaatimusmäärittelytyön lopputuloksena syntyvä ohjelmisto voidaan julkaista hyvinkin pitkän ajan jälkeen projektin aloittamisesta. Asiakas on voinut unohtaa projektin alussa validoidut vaatimukset, tai liiketoimintatarpeet ovat voineet muuttua ajan kuluessa. Lopullinen ohjelmisto ei välttämättä vastaa enää julkaisuhetken tarpeita, vaikka ohjelmisto olisi kehitetty täsmällisesti vaatimusdokumentin mukaisesti. Ketterä ohjelmistokehitys elää asiakkaan tarpeiden ja muuttu-

vien olosuhteiden mukana, vaatimuksia voidaan kehittää jokaisen kehittämissyklin välillä. Ketterää menetelmää hyödyntäen asiakas on jatkuvasti tietoinen siitä, millainen ohjelmistosta on kehittymässä.

3 Syitä ohjelmistoprojektien epäonnistumiseen

Ohjelmistoprojektien onnistumiseen liittyvistä tutkimuksista tunnetuksi tullut Standish Group [1] luokittelee ohjelmistoprojektit CHAOS-raportissaan vuodelle 2014 kolmeen ryhmään: onnistuneet, haasteelliset ja epäonnistuneet. Onnistuneet projektit ovat projekteja, jotka valmistuvat ajallaan, pysyvät budjetissa ja sisältävät kaikki alunperin määritellyt ominaisuudet ja toiminnot. Haasteelliset projektit valmistuvat, mutta eivät pysy aikataulussa, eivät pysy budjetissa tai eivät sisällä kaikkia alunperin määriteltyjä ominaisuuksia tai toimintoja. Epäonnistuneet projektit ovat projekteja, jotka keskeytetään jossain vaiheessa. Vuoden 2015 raporttiin Standish Group [45] kuitenkin muutti onnistuneen projektin määritelmää. Uuden määritelmän mukaan projekti on onnistunut, jos se valmistuu ajallaan, pysyy budjetissa ja lopputulos on asiakasta tyydyttävä ja organisaation strategian mukainen.

Yllä olevan mukaisesti Standish Group on muuttanut onnistumisen määritelmää niin, että alkuperäisen suunnitelman seuraamisen on korvannut asiakastyytyväisyys ja organisaation strategian edistäminen. Sama kehitys näkyy myös muualla ohjelmistoprojekteissa. Ketterät menetelmät yrittävät vastata muutokseen sen sijaan, että muutosta pyrittäisiin välttämään perinteisten menetelmien tavoin. Tämä näkyy myös siinä, että tutkimukset siirtyvät prosessien ymmärtämisestä ja parantamisesta ihmisten ymmärtämiseen ja vuorovaikutuksen parantumiseen. Siitä huolimatta, että tutkimusten keinot tilanteen parantumiseen ovat muuttuneet paljon viimeisen 30 vuoden aikana, itse ongelmat ovat pysyneet suhteellisen muuttumattomina.

Tämän luvun tarkoitus on esitellä laajasti aikaisemmissa tutkimuksissa havaitut yleisimmät ohjelmistoprojektin epäonnistumiseen liittyvät syyt. Luvussa käsitellään kirjallisuudesta löydettyjä riskejä ja otetaan erityisesti tarkastelun alle selkeimmät syyt, jotka liittyvät loppukäyttäjän rooliin ohjelmistoprojektissa ja ovat siten mielenkiintoisia tutkimusongelmien kannalta. Alaluvuissa käsitellyt riskit ovat osittain päällekkäisiä ja liittyvät vahvasti toisiinsa. Alaluvut ovat melkein eri näkökulma samaan ongelmaan.

Alaluvussa 3.1 käydään lyhyesti läpi aiempia tutkimuksia ohjelmistoprojektin epäonnistumisen syistä, ja luodaan yleiskuva aiheesta. Toisessa alaluvussa 3.2 kes-

kitytään joihinkin 3.1 tutkimuksista, ja niiden pohjalta käydään läpi loppukäyttäjään liittyvät epäonnistumisen syyt. Alaluku 3.3 on yhteenveto.

3.1 Tutkimuksia ohjelmistoprojektien epäonnistumisista

Morris [60] käy tutkimuksessaan läpi ohjelmistoprosessin elinkaarta ja ongelmia, ja miten näihin ongelmiin tulisi vastata. Lisäksi Morris kertoo kuinka nämä ongelmat voivat vaikuttaa projektin onnistumiseen. Lopputuloksena Morris korostaa selkeää suunnittelua ja päämäärän merkitystä. Morrisin tutkimus on yksi ensimmäisistä, joissa ohjelmistoprojektia käsitellään vuorovaikutusprosessina.

Pinto ja Slevin [65] tutkivat vuonna 1987 tekijöitä, jotka vaikuttavat onnistuneeseen ohjelmistoprojektiin. Tutkimuksessa haastateltiin 52 kokenutta projektijohtajaa. Näiden johtajien onnistuneen projektin läpiviennin kannalta tärkeäksi kokemien tekijöiden pohjalta tunnistettiin yhdeksän menestystekijää. Nämä menestystekijät liitettiin yhteen, ja muodostettiin kehikko, jonka pitäisi taata onnistunut ohjelmistoprojekti [65]:

- Selkeä Maali
- Pätevä projektipäällikkö
- Johdon tuki
- Pätevä projektiryhmä
- Riittävät resurssit
- Riittävä kommunikaatio
- Projektinjohto
- Palautteen saaminen
- Loppukäyttäjän osallistuminen

Curtis ja muut [21] tutkivat kenttätutkimuksessaan jo vuonna 1998 ohjelmistoprojektin ongelmia laajoissa järjestelmissä. Tätä tutkimusta on käytetty paljon lähteenä alan myöhemmissä tutkimuksissa. Tutkimuksessa käytiin läpi seitsemäntoista projektia yhdeksästä yrityksestä. Haastattelujen tuloksena löydettiin kolme ongelmaa, joita käsiteltiin yksilön, tiimin, projektin, organisaation ja liiketoimintaympäristön näkökulmasta. Nämä ongelmat olivat toimintaympäristön heikko tuntemus,

muuttuvat vaatimukset ja viestinnän tai yhteistoiminnan ongelmat. Tutkimuksesta voi havaita samat ongelmat, jotka toistuvat myös tämän päivän tutkimuksissa.

Bubenko [15] käsittelee julkaisussaan vuodelta 1995 vaatimusmäärittelyn haasteita. Artikkelin on lyhyt ja ytimekäs kirjallisuuteen perustuva kuva tunnetuista haasteista. Bubenko myös määrittelee hyvin taustan ja lähtökohdat vaatimusmäärittelytyölle. Hänen mielestään suurin osa vaatimusmäärittelytyön ongelmista pohjautuu siihen, miten vaatimusmäärittelytyötä arvostetaan liiketoimintajohdon ja IT-johdon tasolla. Bubenko tunnistaa haasteita johdon ja organisaation, loppukäyttäjien ja muiden sidosryhmien, toimintamallien, vaatimusmäärittelytyöhön käytettävien työkalujen ja vaatimusmäärittelytyön osaamisen osalta.

El Emam ja Madhavji [28] käsittelevät kenttätutkimuksessaan vaatimusmäärittelytyöhön liittyviä ongelmia. He jatkoivat Curtisin ja muiden [21] tutkimusaiheen parissa omalla laajemmalla tutkimuksellaan. Tutkimuksen tarkoituksena oli luoda suosituksia, joiden avulla ammattilaiset voivat parantaa vaatimusmäärittelyprosessia, ja ohjata tulevaisuuden tutkimuksia menetelmistä ja työkaluista. Tutkimuksessa käytettiin lähdemateriaalina haastatteluja ja olemassa olevaa kirjallisuutta. Tutkimuksessa löydettiin seitsemän vaatimusmäärittelytyön ongelmakohtaa. Näitä ongelmakohtia käytiin läpi, ja ehdotettiin ongelmakohtiin korjaavia toimenpiteitä, sekä tutkittiin mitkä tekijät estävät korjaavia toimenpiteitä. Näitä olivat [28]:

- Valmisohjelmistovaihtoehdon tutkiminen
- Prosessimallin yksityiskohtaisuuden taso
- Nykyiseen järjestelmään tutustuminen
- Käyttäjän osallistuminen
- Epävarmuuden hallinta
- Tietokoneavusteisen ohjelmistotuotannon työkalut
- Projektinhallinnan valmiudet

Vuotta myöhemmin El Emam ja muut [29] tutkivat käyttäjän osallistumisen vaikutuksia vaatimusmäärittelyprosessin onnistumiseen. Tutkimuksen aihe saattoi löytyä osittain samojen kirjoittajien aiemmasta tutkimuksesta. Kyselytutkimusten vastauksista tutkittiin, onko asiakkaan osallistumisella ja projektin onnistumisella yhteyttä sekä selkeässä että paljon epävarmuustekijöitä sisältävässä projektissa. Tulokset

ten pohjalta tehtiin suosituksia loppukäyttäjien osallistumisen edistämiseen projektissa, ja tehtiin pohjaa tuleville tutkimuksille. Tutkimuksen lopputuloksena oli, että paljon epävarmuustekijöitä sisältävässä projektissa kannattaa loppukäyttäjän osallistumista kannattaa lisätä.

Keil ja muut [48] tunnistavat ohjelmistoprojekteihin liittyviä riskejä. Haastattelututkimukseen osallistui kokeneita projektipäälliköitä Suomesta, Hong-Kongista ja Yhdysvalloista. Tutkimuksessa löydettiin yleismaailmallisia riskejä, joita jaoteltiin nelikenttään niin, että toisessa ulottuvuudessa oli riskin suhteellinen suuruus ja toisessa projektipäällikön vaikutusmahdollisuudet riskin toteutumisen estämiseen. Tämän nelikentän ryhmitellyille riskeille tunnistettiin ja nimettiin kategoriat. Tutkimuksessa löydetty ohjelmistoprojektin riskit olivat seuraavat [48]:

- Johdon tuen puute
- Loppukäyttäjän sitoutumisen puute
- Vaatimusten väärinymmärtäminen
- Riittämätön loppukäyttäjän osallistuminen
- Loppukäyttäjän toiveiden hallinnan epäonnistuminen
- Muuttuva raja- ja odotukset
- Puutteet projektiryhmän osaamisessa
- Vaatimusten muuttaminen kesken projektin
- Uusi ja tuntematon teknologia
- Tehoton tai epäsovelias henkilökunta
- Loppukäyttäjäosastojen väliset ristiriidat

Wiegiers [85] käsittelee artikkelissaan vaatimusmäärittelyn ongelmia. Hänen mukaansa polku laadukkaaseen sovellukseen alkaa erinomaisista vaatimuksista. Vaatimusmäärittelytyön laiminlyönti on yleinen syy turhautumiseen ja projektin epäonnistumiseen. Hän tunnistaa kymmenen yleistä ongelmaa, joihin on helppo joutua, jos projektiryhmän jäsen tai asiakas ei ota niitä vakavasti. Hän myös esittää ensimmäisiä oireita mistä voi tunnistaa ongelman läsnäolon, sekä ehdottaa mahdollisia korjauksia ongelmiin. Wiegiersin ongelmat ovat seuraavat [85]:

- Sanan vaatimus epäselvä merkitys
- Loppukäyttäjän riittämätön osallistuminen
- Epämääräiset ja epäselvät vaatimukset
- Ei-priorisoidut vaatimukset
- Sellaisten toiminnallisuuksien toteuttaminen, joita kukaan ei käytä
- Vaatimusmäärittelytyön jatkaminen loputtomasti
- Jatkuva rajauksen muuttuminen
- Riittämätön muutosprosessi
- Riittämätön muutosvaikutusten analysointi
- Riittämätön versionhallinta

Hofmannin ja Lehnerin kenttätutkimuksessa [39] vuodelta 2001 käsiteltiin vaatimusmäärittelytyötä menestystekijänä ohjelmistoprojektissa. Tutkimus suoritettiin haastattelemalla 15 vaatimusmäärittelytyötiimiä. Haastattelujen pohjalta tutkittiin, miten tieto sovellusympäristöstä, käytettävät resurssit ja käytettävä ohjelmistoprosessi vaikuttavat ohjelmistoprojektin onnistumiseen. Puutteellinen vaatimusmäärittelytyö on suurin yksittäinen syy ohjelmistoprojektin epäonnistumiseen. Myös Hofmann ja muut pohjaavat tutkimustaan Curtisin ja muiden tutkimukseen [21].

Coughlanin ja Macredien [20] vertailivat vuonna 2002 vaatimusten keräämisen menetelmiä kirjallisuuden pohjalta. Käyttäjävaatimusten kerääminen ja niistä kommunikointi sisältyy järjestelmäkehityksen aikaiseen ja virhealttiin vaiheeseen. Tutkimuksen mukaan projektin alusta asti onnistunut kommunikaatio on projektin onnistumisen edellytys. Tästä johtuen Coughlan ja Macredie hakevat parhaita menetelmiä kehittäjien ja loppukäyttäjien välisen kommunikaation parantamiseen.

Beecham ja muut [10] ovat tehneet perusteellisen laadullisen tutkimuksen vuonna 2003 ohjelmistoprojektien ongelmista haastattelemalla kahdentoista ohjelmistoyrityksen kokenutta henkilökuntaa. Tutkimuksessa kerättiin määrällistä aineistoa 45:stä ryhmästä, johon kuului yli 200 jäsentä. Tarkastelun kohteena oli miten erilaiset ammattiryhmät vastaavat ohjelmistokehitysprosessin ongelmiin. Tarkoituksena oli saada parempi kuva eri ammattiryhmien kokemuksista siitä, miten ohjelmistoprosessia voidaan tehostaa. Tutkimusongelmana oli, ovatko eri ohjelmistokehitysongelmat liitoksissa eri tasoihin prosessien kypsyyksissä ja onko kehittäjillä,

projektipäälliköillä ja ylimmän tason johdolla erilaiset ongelmat ohjelmistoprosessin suhteen. Tutkimuksessa löydetty viisi yleisintä ongelmaa olivat seuraavat [10]:

- Henkilökunnan vastuullisuus, taidot, tiedot ja koulutus
- Uusien työkalujen ja teknologioiden käyttöönotto
- Dokumentaation puutteellisuus ja väärin asioiden dokumentointi
- Kommunikaation puute
- Puutteellisen ja huonosti ymmärretyt vaatimukset

Karlsson ja muut [47] käsittelevät artikkelissaan markkinavetoista vaatimusmäärittelytyötä. Tutkimus toteutettiin haastattelututkimuksena, johon kuului 14 haastattelua kahdeksassa ohjelmistoyrityksessä ja neljä kohderyhmätapaamista ammattilaisten kanssa. Tutkimuksen tarkoituksena on lisätä markkinavetoisen vaatimusmäärittelytyön ymmärtämystä ja ehdotuksia tulevaisuuden tutkimusaiheiksi. Tutkimuksessa löydettiin lukuisia haastavia ongelmia sisältäen vuorovaikutuskatkoja markkinoinnin ja kehityksen välillä, sopivan määrän prosessin tukemista, julkaisusuunnitelman muodostamisen epävarmoista arvioista, sekä jatkuvan vaatimusvirran hallinnan. Tutkimusongelmaksi valikoitui minkälaisia vaatimusmäärittelytyön haasteita markkinavetoisessa ohjelmistokehityksessä toimivat yritykset kohtaavat.

El Emamin ja Korun [27] artikkelissa selvitetään ohjelmistoprojektien epäonnistumista. Materiaalia kerättiin IT-osastoille suunnatulla WWW-kyselyllä vuosina 2005 ja 2007. Vuonna 1995 Standish Group [1] julkaisi CHAOS-raportin vuodelta 1994, jonka perusteella ohjelmistoprojektien keskeytysprosentit ovat suuria. El Emam ja Koru haastavat tämän käsityksen omalla tutkimuksellaan, ja kritisoivat Standish Groupia tutkimuksen menetelmien salaamisesta, vertaisarvioinnin puutteesta, epäjohdonmukaisesta raportoinnista ja väärinkäsityksistä projektin epäonnistumisen määrittelyssä. El Emam ja Koru tulevat siihen lopputulokseen, että huomattavasti suurempi osa projekteista, kuin CHAOS-raportissa väitetään, onnistuu tuottamaan jotain valmista.

Tarawneh ja muut [82] tutkivat ohjelmistoprojektin onnistumistekijöitä Jordania-laisissa yrityksissä. Ohjelmistoprojektin hallinta on yksi haastavimmista tehtävistä ohjelmistokehityksessä. Projektin sujuva suunnittelu ja ohjaus ei ole mahdollista ilman luotettavaa hallintaa. Projektipäällikön tulee olla tietoinen riskeistä, jotka vai-

kuttavat ohjelmistoprojektiin. Lähdetietoa kerättiin kyselytutkimuksella. Vastauksista tutkittiin kirjallisuudesta poimittujen yhdeksän menestystekijän esiintymistiheyttä, ja järjestettiin tekijät esiintymistiheyden mukaan. Tutkimuksen tarkoituksena oli tunnistaa tärkeimmät menestys- ja epäonnistumistekijät.

Zhu ja muut [90] toteavat, että muuttuvat vaatimukset ovat riskitekijä ohjelmistoprojektissa, ja tutkivat mistä syistä vaatimukset muuttuvat. He myös luokittelevat sidosryhmät, sekä pohtivat minkälaisia muutoksia nämä sidosryhmät yleensä aiheuttavat, ja kartoittavat näiden muutoksien riskejä. Tutkimus on laadullinen tutkimus kirjallisuudesta pohjalta. Zhun ja muiden tutkimuksen yleisimmät vaatimusten muuttumiseen johtavat syyt ovat seuraavat [90]:

- Vaatimus on huonosti ilmaistu
- Vaatimus voidaan ymmärtää kahdella tavalla
- Loppukäyttäjän tarve muuttuu
- Kehittäjän vaatimusten laiminlyönti ja muutokset
- Loppukäyttäjä peruu vaatimuksen
- Loppukäyttäjän toimintamalli muuttuu
- Järjestelmästä itsestään johtuvat syyt
- Ohjelmistosta ja laitteistosta johtuvat syyt

Viskovic ja muut [84] käsittelevät artikkelissaan huonoja toimintatapoja monimutkaisissa ohjelmistoprojekteissa. Artikkelissa kuvataan huonoja tapoja, ja keinoja, joilla ne voi välttää. Artikkelissa ongelmat jaetaan projektinhallinnan ongelmiin, huonon kommunikaation aiheuttamiin ongelmiin ja projektin toimintamallin puuttumisen aiheuttamiin ongelmiin. Viscovicin ja muiden tutkimuksen pahat tavat olivat seuraavat [84]:

- Projektin toteuttaminen ilman selkeää maalia
- Kyvyttömyys nähdä kaikkien projektien muodostama kokonaiskuva
- Johdon tuen puute
- Loppukäyttäjän tärkeyden huomiotta jättäminen

- Alihankkijoiden käyttäminen omien työntekijöiden sijasta
- Projektissa hankittujen tietojen hyödyntämättä jättäminen
- Virheistä oppimattomuus
- Olemassaolevaan prosessidokumentaatioon uskominen
- Projektin dokumentaation päivittämättä jättäminen
- Toimiminen hyvien toimintatapojen vastaisesti

Nasir ja Sahibuddin [61] tekivät vuonna 2011 erittäin perusteellisen analyytisen kirjallisuuskatsauksen ohjelmistoprojektin menestystekijöistä. Tutkimuksessa läpikäytyjen artikkeleiden projektien koko, toimiala ja maa vaihtelivat. Tutkimukseen valikoitui 43 artikkelia vuosilta 1990–2010. Valittujen tutkimusten pohjalta käytiin läpi kriittisiä tekijöitä, jotka vaikuttavat ohjelmistoprojektin onnistumiseen. Tuloksista muotoiltiin kaksi uutta tutkimusongelmaa: Ovatko ei-tekniset tekijät huomattava syy ohjelmistoprojektien epäonnistumiseen? Ovatko tässä tutkimuksessa ja muissa tutkimuksissa havaitut ongelmat yhteneväisiä? Tämä kirjallisuuskatsaus on hyvä yleiskuvaus niihin tekijöihin, jotka johtavat ohjelmistoprojektin onnistumiseen.

Alladi ja Vadari [3] käsittelevät vuonna 2011 kirjallisuuskatsauksessaan projektin johtamista, ja erityisesti sidosryhmien sitouttamista projektiin. Projektin onnistumisen kannalta on olennaista, että kaikki sidosryhmät sitoutuvat projektiin. Tutkimuksessa käydään läpi, miten löydetään kaikki sidosryhmät ja saadaan heidän tuki projektille. Tutkimuksessa ehdotetaan lopuksi kuuden askeleen mallia sidosryhmien sitouttamiseen, ja esitellään sitouttamisen etuja.

Khaiyum ja muut [49] käyvät läpi syitä, jotka johtavat ohjelmistoprojektien epäonnistumiseen. Heidän näkökulmansa on reaaliaikakäyttöjärjestelmät ja sulautetut järjestelmät. Tutkimus on toteutettu kirjallisuuden pohjalta. Khaiyumim ja muiden tutkimuksessa yleisimpiä projektin ongelmiin johtavia syitä olivat seuraavat [49]:

- Vaatimusten puutteellisuus
- Resurssien puute
- Epärealistinen aikataulu
- Huono suunnittelu
- Riskienhallinnan puutteet

3.2 Yleisimmät loppukäyttäjään liittyvät epäonnistumisen syyt

Seuraavissa alaluvuissa käydään läpi ohjelmistoprojektin epäonnistumiseen liittyviä syitä. Tarkemman tarkastelun kohteena ovat syyt, jotka liittyvät selkeästi tulevan ohjelmiston loppukäyttäjään, sillä nämä ovat tämän tutkimuksen kannalta mielenkiintoisimmat syyt.

3.2.1 Toimintaympäristön tuntemus on puutteellista

El Emam ja Madhavji lähestyvät tutkimuksessaan [28] toimintaympäristön kartoitusta olemassa olevan ratkaisun tutkimisella. Nykyisen järjestelmän tarkastelemisen tarkoitus on ymmärtää nykyistä järjestelmää ja arvioida se. Tarkastelemalla saadaan selville mitä järjestelmä tekee, mitkä ovat siihen liittyvät osapuolet ja miksi se toimii niin kuin toimii. Tarkastelun pohjalta arvioimalla selvitetään nykyisen järjestelmän vahvuudet, heikkoudet ja syyt miksi vanha järjestelmä halutaan korvata uudella.

El Emam ja Madhavji [28] toteavat, että nykyisen järjestelmän ymmärtäminen ja arviointi tarjoavat pohjan uuden järjestelmän rakentamiselle. Lisäksi olemassa olevan järjestelmän tutkiminen auttaa kehittämään toimintaympäristön ymmärtämistä. Edellisen järjestelmän toimintamalleja voidaan myös hyödyntää uuden järjestelmän kehityksessä. Joskus nykyisen järjestelmän tutkiminen jätetään tekemättä sillä perusteella, että nykyinen järjestelmä on jo maksettu ja siihen ei haluta käyttää enempää rahaa.

Bubenkon [15] mukaan vaatimusmäärittely on kuvaus perusolettamuksista, tarpeista ja ongelmaympäristön tuntemuksesta, jota tarvitaan ohjelmistoprojektin läpiviintiin. Vaatimusmäärittely kuvaa kysymyksiä kuten: miksi ohjelmistoa tarvitaan, mitkä ovat ohjelmiston tai ympäristön tavoitteet, mikä on tavoitteiden tärkeysjärjestys, mitkä on organisaation tavoitteet, minkälaisia prosesseja ja niihin liittyviä toimijoita ohjelmistoon liittyy ja minkälaisia muita vaatimuksia ohjelmistoon liittyy?

Curtisin ja muiden [21] mukaan toimintaympäristön syvälinen ymmärtäminen on tarpeen ohjelmistoprojektin läpiviennissä. Ohjelmisto koostuu useista osista, joiden tulee toimia yhdessä. Kokonaisuutta ei hahmoteta, vaikka yksittäiset henkilöt ymmärtävätkin eri komponentit yksittäin. Kaikkien ohjelmistoprojektiin osallistuvien tulisi ymmärtää sovelluksen toimintaympäristö ja sen asettamat vaatimukset. Yksi tutkimukseen osallistuneista järjestelmäinsinööreistä totesi, että koodin kirjoittaminen ei ole ongelma, vaan ongelman ymmärtäminen on ongelma.

Curtisin ja muiden [21] mukaan toimintaympäristön ymmärtäminen sisältää hy-

vin monentyyppistä tietoa. Lähtökohdat esimerkiksi korkean toimintavarmuuden sotilas- tai ilmailusovelluksiin ovat hyvin erilaiset kuin lähtökohdat sulautettuihin järjestelmiin. Projektiin tulevilla uusilla osallistujilla voi olla hyvät tietotekniset taidot, mutta monesti toimintaympäristön tuntemus on aluksi puutteellista.

Toimintaympäristön tuntemus auttaa Curtisin ja muiden [21] mukaan yksilön näkökulmasta ymmärtämään vaatimuksia paremmin. Jos ohjelmistokehittäjä ei tunne toimintaympäristöä, hänellä voi olla hankala toteuttaa vaatimusta niin, että se vastaisi toimintaympäristön ongelmaan. Ohjelmistoprojektista löytyy monesti yksi tai kaksi poikkeuksellista yksilöä, jotka ymmärtävät toimintaympäristöä paremmin kuin muut. Monesti nämä yksilöt määrittävät projektin suunnan ja onnistumisen. Heidän näkemyksensä toimintaympäristöstä auttaa yhdistämään reaali maailman rakenteet ohjelmiston toiminnoiksi ja tietorakenteiksi. Nämä poikkeukselliset yksilöt löytävät ratkaisut loppukäyttäjien ongelmiin, vaikka loppukäyttäjillä ei olisi-kaan kykyä ilmaista tarpeitaan tarpeeksi selkeästi. Yleensä projektin suunnan määrää yhden tai useamman poikkeuksellisen yksilön muodostamasta joukko, jolla on vahvin käsitys, mihin lopputulokseen halutaan päästä.

Projektin näkökulmasta on Curtisin ja muiden [21] mukaan tärkeää ymmärtää minkälaisia riippuvuuksia sovelluksen eri osilla on keskenään. Toimintaympäristön tuntemus auttaa näkemään riippuvuuksia eri osien välillä. Tämä ympäristötieto jalostuu ja syventyy esimerkiksi suunnitelluissa koulutuksissa, loppukäyttäjien kanssa vuorovaikutuksessa, prototyypeissä ja simulaatioissa, tekemällä virheitä, projektin aikataulun tai budjetin ylittyessä ja projektin peruuntuessa. Yleensä asiakkaat eivät halua maksaa tämän tiedon hankkimisesta, sillä he uskovat, että toimittajalla pitäisi jo olla kaikki tarvittava tieto. Projektiin tarvittava aika aliarvioidaan monesti tästä syystä, sillä aikataulussa otetaan huomioon vain projektin toteutus. Projektin henkilöstön toimintaympäristöön tutustumista ja sovelluksen tarkoituksen ymmärtämiseen käytettyä aikaa ei yleensä oteta huomioon. Monesti epäonnistuneista projekteista opittu toimintaympäristön tuntemus auttaa myöhempiä projekteja pääsemään oikeaan maaliin, sillä epäonnistumisissa opitaan toimintaympäristöstä asioita, joita ei onnistuneessa projektissa löydettäisi.

Curtisin ja muiden [21] mukaan yrityksen näkökulmasta toimintaympäristön ymmärtämisen syventäminen on merkittävä kustannus. Toimintaympäristön oppineet tekniset ihmiset voivat edetä urallaan kehittäjistä johtajiksi. Siinä työssä toimintaympäristön ymmärtäminen auttaa tekemään ratkaisuja, jotka palvelevat organisaation tarvetta. Valitettavasti tämä siirtymä vie toimintaympäristön ymmärtämys-

tä pois teknisten ihmisten keskuudesta. Lisäksi johtaja on kauempana toteutuksesta ja hänen toimintaympäristön ymmärtämyksensä voi sitä kautta vähentyä.

Myös Hofmann ja Lehner [39] toteavat, että toimintaympäristön tuntemuksen puute on yksi merkittävimmistä ongelmista ohjelmistoprojekteissa. Tutkimuksessa loppukäyttäjä otettiin mukaan mahdollisimman aikaisessa vaiheessa ja yritettiin saada häneltä mahdollisimman paljon palautetta vaatimusmäärittelytyön aikana. Loppukäyttäjällä on paljon toimintaympäristön tuntemusta ja se on tärkeää saada mukaan projektiin.

Viskovic ja muut [84] tuovat toimintaympäristön ymmärtämisen yhteydessä esille toimintaympäristön dokumentaation ajantasaisuuden. Heidän mukaansa ei tulisi olettaa, että olemassa oleva dokumentaatio toimintaympäristöstä on ajantasaista. Yleinen käytäntö on osoittanut, että liiketoimintaprosessit ovat yleensä dokumentoitu riittämättömästi. Dokumentaatio saattaa olla päivittämättä, tai sitä ei ole olemassa. Joskus liiketoimintaprosessien dokumentaatio on jopa ristiriitaista. Liiketoimintaprosessien määrittely toistuvasti kesken projektin aiheuttaa mahdollisesti viivästymistä projektissa. Projektin aikataulua suunniteltaessa tulee ottaa huomioon, että liiketoimintaprosesseja ei ole yleensä analysoitu ja dokumentoitu. Vaikka liiketoimintaprosessit olisikin dokumentoitu, niin prosessit pitää analysoida ja varmistaa, että dokumentaatio vastaa todellisuutta. Liiketoimintaprosesseja analysoiva voi aloittaa keskustelun projektin jäsenten ja prosessin käyttäjien välillä. Prosessin käyttäjien haastatteluun tulee valmistautua huolella, sillä projektin jäsenet tai konsultti kohtaavat prosessin käyttäjän ensimmäistä kertaa. Prosessin omistaja voi olla halukas hyväksymään konsultin kokemuksen aikaisemmista projekteista. Projektin jäsen voi aavistaa, että prosessin käyttäjä ei halua luopua vanhasta prosessista, vaikka se ei olisikaan ihanteellinen. Tällaisessa tapauksessa projektin jäsenen tulisi käyttää vaikutusvaltaansa ja hyvien ratkaisujen tuomaa uskottavuutta, vakuuttamaan prosessin uudelleensuunnittelun tarpeesta.

El Emamin ja Madhavjin [28] mukaan nykyisen toimintaympäristön kuvaamisessa parhaita asiantuntijoita ovat loppukäyttäjät, jotka toimivat kyseessä olevassa ympäristössä. Toimintaympäristön puutteelliseen kartoitukseen siis auttaa loppukäyttäjän mukaan ottaminen aikaisessa vaiheessa.

Bubenkon [15] kuvaaman perusteella toimintaympäristön tuntemus on tukeva pohja, jolle voidaan lähteä rakentamaan vaatimusmäärittelytyötä. El Emamin ja Madhavjin [28] kuvaamasti nykyisen järjestelmän kartoitus antaa arvokasta tietoa toimintaympäristöstä, auttaa ymmärtämään itse ongelman ja on hyvä lähtökoh-

ta toimintaympäristön tuntemuksen kasvattamiseen. Nykyisen järjestelmän heikkouksien arvioiminen auttaa ymmärtämään nämä ongelmakohdat ja niiden vaikutukset. Näin ollen kunnollinen toimintaympäristöön tutustuminen on onnistuneen sovellusprojektin olennainen osa.

Curtis ja muut [21] painottavat vahvasti toimintaympäristön ymmärtämisen tärkeyttä ja käyvät läpi erilaisia näkökulmia toimintaympäristön ymmärtämiseen. He painottavat sovelluksen kokonaisuuden ymmärtämistä toimintaympäristön ymmärtämisen kautta, sekä erilaisten arkkitehtuurien ja reaali maailman toimintamallien välisten yhteyksien ymmärtämistä.

Viskovic ja muut [84] tuovat esiin huolen projektin tietojärjestelmään liittyvien prosessien dokumentaation ajantasaisuudesta. Heidän mukaansa dokumentaatioon ei voi luottaa, vaan toimintaympäristö pitää joka tapauksessa analysoida. Heidän mielestään tietojärjestelmään liittyvän prosessin parissa toimiva käyttäjä, eli tulevan järjestelmän loppukäyttäjä, on se henkilö, jonka kautta taustakartoitusta kannattaa lähteä viemään eteenpäin. He yhdistävät siten myös taustan puutteellisen määrittelyn vahvasti loppukäyttäjään.

3.2.2 Yhteinen maali ja muut maalin ongelmat

Yhteisen maalin puuttuminen nousi selkeästi esiin lähdekirjallisuudesta. Myös muuttuvat vaatimukset korostavat selkeän maalin tärkeyttä, sillä kuva maalista auttaa ymmärtämään vaatimuksia ja niiden tarkoitusta. Ohjelmistoprojektin maalia ja siihen liittyviä ongelmia alettiin käsitellä artikkeleissa 1980-luvulla. Yksi ensimmäisistä oli Morris [60], joka määritteli artikkelissaan projektin ensimmäiseksi vaiheeksi toteutettavuuden toteamisen, johon sisältyi tarkastus onko projektilla liiketoimintamaali ja onko se toteutettavissa.

Pinto ja Slevin [65] nostivat tärkeimmäksi menestystekijäksi selkeästi määritellyn maalin, joka sisältää projektin yleisen päämäärän ja sitouttaa projektiryhmän jäsenet siihen. Pinto ja muut esittelevät käsitteen ”projektin missio”, joka tarkoittaa, että projektin maali on selvä ja ymmärretty sekä projektin jäsenille että sidosryhmille. ”Projektin missio” pitää sisällään myös sen, että maali on linjassa organisaation tavoitteiden kanssa.

Curtisin ja muiden mukaan [21] vaatimukset elävät lähes kaikissa ohjelmistoprojekteissa. Ympäristö aiheuttaa vaatimusten elämistä esimerkiksi useiden asiakkaiden ristiriitaisten vaatimusten tai säädösten ja lakien kautta. Nämä muutokset voivat aiheuttaa pienemmässä mittakaavassa vaatimusten elämistä myös organi-

saation sisällä yksikköjen välillä. Näihin vaatimusten muutoksiin ei voi maalin välillä juurikaan vaikuttaa. Projektin sisäisistä syistä tapahtuva vaatimusten eläminen kuitenkin johtuu yleensä selkeän maalin puutteesta. Ilman selkeää maalia projektin tarkoitusta ei saada käännettyä selkeiksi vaatimuksiksi. Ilman selkeää tarvetta poliittisista syistä käynnistetyn projektin vaatimukset heijastelevat yleensä projektin maksajan tarpeita, jolloin loppukäyttäjien kannalta olennaiset vaatimukset jäävät toteuttamatta.

Tarawnehin ja muiden mukaan [82] projektilla tulee olla selkeä liiketoiminnallinen tavoite. Toisin sanoen projektin maalin tulee vastata liiketoiminnalliseen tarpeeseen. Heidän tutkimuksessaan tämä sijoittui tärkeysjärjestyksessä neljänneksi. Selvä liiketoiminnallinen maali auttaa tiimityöskentelyssä ja vaatimusten määrittelyssä. Selvä liiketoiminnallinen maali auttaa myös projektipäällikköä projektin onnistumisen arvioinnissa. Myös Nasir ja muut tunnistivat tarpeen selkeään liiketoiminnalliseen maaliin tutkimuksessaan [61] ja pitivät sitä toiseksi tärkeimpänä onnistumistekijänä ohjelmistoprojektissa.

Tarawnehin ja muiden tutkimus [82] paljastaa, että mitä tiiviimpi projekti sen luultavammin se onnistuu. Tämä onkin tutkimuksen viidenneksi tärkein onnistumiseen vaikuttava menestystekijä. Projektin koko vaikuttaa projektin keston ja siten myös projektin budjettiin. Keil ja muut [48] esittävät tutkimuksessaan nelikenttäänalyysin, jossa toisella akselilla on uhkien vakavuus ja toisella akselilla uhkiin vaikutusmahdollisuudet. He sijoittavat projektin laajuuden ja vaatimukset vakavaksi uhaksi, joka on kuitenkin hyvin hallittavissa. Projektin onnistumisen mahdollisuudet parantuvat, jos projekti rajataan mahdollisimman tiiviiksi. Projektista tulisi siivota pois kaikki mikä ei ole olennaista tai välttämätöntä. Lyhyesti ilmaistuna: *"Tiedänkö mitä olen rakentamassa ja miten se muuttuu ajan kuluessa?"* [48]. Projektin edetessä projektin rajausta elää ja tarkentuu luonnostaan.

Viskovic ja muut [84] nostavat artikkelissaan ensimmäisenä yleisenä pahana tapana IT-projektin läpiviennin ilman selvää maalia. Esimerkkinä hyvästä projekti-maalista he esittävät J. F. Kennedyn Apollo-avaruusohjelmalle määrittelemän maalin [84]: *"Uskon, että kansakuntamme tavoitteena tulisi olla ihmisen vieminen kuuhun tämän vuosikymmenen aikana, ja hänen tuomisensa sieltä turvallisesti takaisin maahan."* Maali tulee määrittää huolimatta siitä, että se voi olla byrokraattista ja tylsää. Taulukossa 3.1 on esimerkki maalin määrittelystä.

Viskovicin ja muiden [84] mukaan projektin maalin pitää olla SMART-periaatteen mukainen. SMART-termi tulee sanoista, jotka on määritelty taulukossa 3.2 Vis-

Taulukko 3.1: Hankesuunitelman pääkohdat [84]

Projektin maali	Määrittelee selvästi projektin tärkeimmät liiketoimintatavoitteet perustuen organisaation strategiaan niin, että maali on selvä sekä johdolle että kehittäjille.
Projektin toimintaympäristö	Liiketoiminta- ja teknologiaympäristö, johon projekti vaikuttaa tai johon projekti tulee.
Projektin hyödyt	Projektilla saavutettavat selkeästi määritellyt hyödyt. Hyödyt eivät saa sisältää pelkästään teknisiä saavutuksia.
Projektin lopputulos	Projektin tuottaman järjestelmän kuvaus.
Riskit	Riskit, jotka voivat vaikuttaa projektiin.
Arvio projektin kestosta ja kustannuksista	Arvio projektin kestosta ja kustannuksista. Käytökelpoinen vertailtaessa projektia muihin projekteihin.

kovicin ja muiden [84] mukaan epäselvät projektin maalit voidaan välttää, jos ohjausryhmä vaatii määrämuotoista projektiehdotusta, joka sisältää selvästi määritellyt ja selitetyt maalit projektille.

Viscovic ja muut [84] käsittelevät myös epäpätevyyttä ymmärtää kaikkien projektien muodostamaa kokonaisuutta. Monissa organisaatioissa on meneillään useampia projekteja samaan aikaan. Monien projektien hallitseminen tehokkaasti tuo monia etuja organisaatiolle ja helpottaa projektinhallinnan ongelmien välttämistä. Isot projektit, jotka vaikuttavat koko organisaatioon, koostuvat monista prosesseista ja niiden aliprosesseista. Siksi kokonaiskuva kaikista projekteista on äärimmäisen tärkeää. Kokonaiskuvan puuttuminen johtaa resurssien hukkaamiseen lyhytaikaisiin projekteihin, ristiriitoihin samaan aikaan etenevien projektien välillä ja samanaikaisten projektien synergiaetujen menettämiseen. Projektipäälliköiden ja organisaation johdon tulee oppia ymmärtämään projekteja ja niiden keskinäisiä yhteyksiä, ja oppia hallitsemaan kokonaisuutta väliaikaisten projektien välttämiseksi. On tärkeää tunnistaa, jos uusi projekti on törmäyskurssilla jo käynnissä olevan projektin kanssa. Yleensä suurimmat ongelmat johtuvat resurssien puutteesta tai budjetista. Projekti kokonaisuuden tehokasta hallintaa voidaan tukea projektinhallintaohjelmistojen avulla.

Taulukko 3.2: S.M.A.R.T. -periaate [84] mukaan Haughey [36]

Specific	Maali on hyvin määritelty ja selkeä kenelle tahansa, jolla on perustietämys projektista.
Measurable	Maalin saavutettavuus, edistyminen ja valmistuminen voidaan mitata.
Agreed upon	Maali on kaikkien sidosryhmien kesken sovittu.
Realistic	Maali on saavutettavissa käytettävissä olevalla ajalla, resursseilla ja tiedolla.
Time based	Maalin saavuttamiseen on riittävästi aikaa.

Nasirin ja muiden analyttisessä kirjallisuuskatsauksessa [61] selkeä päämäärä ja maalit oli toiseksi yleisin kahdestakymmenestäkuudesta menestystekijästä. Se oli mainittu kahdessakymmenessäneljässä tapauksessa 43:sta tutkimukseen valitusta lähteestä.

Khaiyum ja muut [49] käsittelevät tutkimuksessaan reaaliaikajärjestelmiä, mutta puhuvat vaatimuksista myös yleisellä tasolla. Heidän mukaansa projektin liiketoimintaodotukset ja odotetut toiminnallisuudet tulee ymmärtää selkeästi. Projektin koko, monimutkaisuus ja kesto tulee myös rajata tiiviiksi.

Tarawnehin ja muiden [82] mukaan projektin maalin tulee tuottaa liiketoimintahyötyä organisaatiolle. Maali auttaa myös epäselvien vaatimusten tulkinnessa, ja projektin pitämisessä oikeassa suunnassa.

Viskovicin ja muiden [84] mukaan projektista tulisi olla selkeästi hyötyä, ja sen tulisi palvella organisaatiota. Tästä voidaan päätellä, että projektin pitää tuottaa jokin hyötyä jollekin käyttäjäryhmälle, jota yleensä kutsutaan loppukäyttäjiksi.

Projektin maali liittyy myös projektin toimintaympäristön ymmärtämiseen, sillä maali on se tilanne minne nykyisestä ympäristöstä halutaan päästä. Maali edustaa ratkaisua loppukäyttäjän ongelmaan.

3.2.3 Loppukäyttäjä ei ole mukana projektissa

Tämän luvun alaluvuissa 3.2.1 ja 3.2.2 kuvattiin ongelmia, joita aiheuttaa puutteellinen toimintaympäristön tuntemus ja puutteellinen maalin määrittäminen. Loppukäyttäjän rooli näiden ongelmien ratkaisussa on merkittävä. Loppukäyttäjän riittämätön osallistuminen ohjelmistoprojektiin on myös kirjallisuudessa esiintyvä riski.

El Emamin ja Madhavjin [28] artikkelin mukaan loppukäyttäjän osallistumisen koettiin vähentävän tarvetta vaatimusten tarkentamiseen projektin edetessä ja parantavan ratkaisun sopivuutta tarpeeseen. Loppukäyttäjät myös suhtautuvat paremmin uuteen ratkaisuun, kun he ovat olleet mukaan sen kehittämisessä alusta lähtien.

El Emam ja muut [29] tutkivat aikaisempien tutkimusten olettamaa, jonka mukaan loppukäyttäjien osallistuminen vaatimusmäärittelytyöhön on erityisen suotavaa, jos vaatimukset eivät ole selkeitä, ja projektiin liittyy epävarmuustekijöitä. Tutkimuksessa havaittiin, että loppukäyttäjien osallistumisen ja epävarmuuden suhteella on merkittävä vaikutus vaatimusmäärittelyn laatuun. Suurin hyöty loppukäyttäjien osallistumisesta saadaan, kun projektiin liittyy paljon epävarmuustekijöitä. Loppukäyttäjien osallistuminen auttaa saavuttamaan yhteisymmärryksen loppukäyttäjien välillä, ja auttaa loppukäyttäjää ymmärtämään liiketoimintaprosessiin ja sitä minkälainen tulevan järjestelmän tulisi olla.

Toisaalta El Emamin ja muiden [29] mukaan, jos projektiin ei liity epävarmuustekijöitä, niin loppukäyttäjien osallistumisella ei ole merkitystä vaatimusmäärittelyn laatuun. Lisäksi, jos projektiin ei liity epävarmuutta, niin se voi lisätä loppukäyttäjien turhautumista, sillä he eivät tunne saavansa samanlaista hyötyä osallistumisestaan. Tutkimuksen mukaan loppukäyttäjien osallistumisen lisääminen voi jopa laskea vaatimusmäärittelyn laatua, jos ympäristössä on vähän epävarmuustekijöitä. El Emam ja muut päätyivät seuraavaan kolmeen toteamaan [29]:

- Epävarmuuden kasvaessa loppukäyttäjien osallistumisen lisääminen vähentää epävarmuuden negatiivisia vaikutuksia ja parantaa vaatimusmäärittelytyön laatua.
- Epävarmuuden vähentyessä loppukäyttäjien osallistumisen lisäämisen hyvät puolet vähentyvät, ja osallistuminen syö resursseja, joten tunnetussa ympäristössä loppukäyttäjien osallistumista voidaan vähentää.
- Tutkimuksessa ei oletuksen vastaisesti kuitenkaan havaittu, että loppukäyttäjien osallistumisella tai epävarmuuden määrällä olisi vaikutusta projektin kustannuksiin tai sen tuottaman sovelluksen arkkitehtuurin laatuun.

Keil ja muut [48] ryhmittelevät riskit, joiden merkitys ohjelmistoprojektin kokonaisriskiin on suuri ja projektipäällikön vaikutusmahdollisuudet ovat pienet. Tutkimukseen osallistuneet projektipäälliköt kokivat useiden riskien liittyvät asiakkaan

toimeksiantoon. Usein ohjelmistoprojektit, jotka onnistuvat, ovat sellaisia ohjelmistoprojekteja, joihin tarpeellisuuteen sekä johto että loppukäyttäjät uskovat ja ovat sitoutuneet. Ilman selvää tarkoitusta ja päämäärää olevat ohjelmistoprojektit eivät ole järkeviä. Tähän kategoriaan kuuluvat esimerkiksi seuraavat ongelmat [48]:

- Johdon sitoutumisen puuttuminen
- Loppukäyttäjien sitouttamisen epäonnistuminen
- Loppukäyttäjän osallistumisen puute

Keilin ja muiden [48] mukaan tämän kategorian riskeihin projektipäällikkö voi vaikuttaa luomalla ja ylläpitämällä hyviä suhteita loppukäyttäjään ja edistämällä loppukäyttäjän osallistumista projektiin. Näiden suhteiden luominen vie aikaa, ja jo luotuja suhteita pitää ylläpitää myös ohjelmistoprojektin jatkuessa. Yksi tapa ylläpitää laajapohjaista sitoutumista on toteuttaa mahdollisimman monien sidosryhmien toiveita. Projektit, joilla on laaja kannatus eri sidosryhmien joukossa, onnistuvat todennäköisemmin kuin projektit, joilla on vain yhden sidosryhmän edustajan kannatus. Toinen tapa edistää sitoutumista projektiin on tehdä siitä mahdollisimman palkitseva osallistujille. Myös loppukäyttäjän toiveiden hallitseminen liittyy tämän ryhmän riskien hallintaan, sillä epärealistiset odotukset aiheuttavat ongelmia loppukäyttäjän sitoutumisen suhteen. Projektipäällikkö ei voi hallita tämän ryhmän riskejä, mutta voi vaikuttaa niihin. Mahdolliset vaikutuskeinot liittyvät suhteiden luontiin, luottamuksen rakentamiseen ja vuorovaikutustaitoihin.

Hofmann ja Lehner [39] määrittelevät loppukäyttäjän sidosryhmän jäsenenä, joka haluaa aikaansaada muutoksen mahdollisimman pienellä häiriöllä omaan työhönsä. Loppukäyttäjältä saa arvokasta tietoa liiketoimintaprosesseista ja toimintatavoista. Tutkimuksessa keskityttiin tarkastelemaan vaatimusmäärittelytyötä toimintaympäristön puutteelliseen ymmärtämisen, käytössä olevien resurssien ja vaatimusmäärittelyprosessin kannalta. Loppukäyttäjän sitouttamisen epäonnistuminen vaikuttaa merkittävästi toimintaympäristöstä saatavan tiedon saamiseen ja sitä kautta myös vaatimusmäärittelytyön onnistumiseen.

Tarawneh ja muiden [82] haastattelututkimuksessa loppukäyttäjän osallistuminen on toiseksi suurin tunnistettu tekijä projektin onnistumisessa. Loppukäyttäjä voi selkiyttää työn tavoitteita avaamalla omaa työnkuvaansa, ja näin auttaa muuta projektiryhmää ratkaisemaan työhön liittyviä epäselvyyksiä. Loppukäyttäjän osallistuminen ohjelmistoprojektiin vähentää myös muutosvastarintaa uuteen järjestelmään siirryttäessä, sillä hän voi vakuuttaa muut tulevat loppukäyttäjät projektin

tärkeystensä. Loppukäyttäjä voi myös selventää muiden loppukäyttäjien näkökulmia muulle projektiryhmälle.

Viskovic ja muut [84] käsittelevät loppukäyttäjän sitouttamisen tarpeellisuutta käsitellessään vuorovaikutuksen tärkeyttä loppukäyttäjän kanssa. Monet projektit epäonnistuvat, sillä loppukäyttäjät eivät hyväksy lopputulosta. Huono vuorovaikutus kehittäjien ja loppukäyttäjien kanssa on usein syy siihen, miksi loppukäyttäjän eivät hyväksy projektin tulosta. Organisaation rakenteen vuoksi projektiryhmä on fyysisesti eri tilassa kuin loppukäyttäjä. Joskus myös kehittäjien näkökulma on liian teknologiakeskeinen, ja loppukäyttäjän tarpeet unohtuvat. Ulkopuoliset konsultit työskentelevät useissa projekteissa ja heillä on liikaa työkuormaa. Avainkäyttäjät ovat kiireisiä omien töidensä parissa, ja eivät ehdi keskittyä projektiin. Nämä syyt johtavat Viskovicin ja muiden [84] mukaan huonoon vuorovaikutukseen kehittäjien ja loppukäyttäjien välillä. Monissa tapauksissa kehittäjät toteuttavat sitä mitä ajattelevat asiakkaan haluavan, eikä sitä, mitä asiakas oikeasti haluaa. Vuorovaikutusongelmien tiedostaminen on ensimmäinen askel kohti näiden ongelmien ratkaisua. Vuorovaikutusongelmien ratkaisemiseksi kannattaa siirtää kehittäjät ja loppukäyttäjät fyysisesti samaan tilaan, sillä muuton kustannukset korvaa parempi lopputulos, joka säästää työaika ja tehostaa toimintaa. Lisäksi kannattaa järjestää tapahtumia ja yhdessäoloa joukkuehengen korottamiseksi ja ideoiden vaihtamiseksi kehittäjien ja asiakkaiden välillä, sekä järjestää koulutusta vuorovaikutustaidoissa, ja tukea kehittäjien ja loppukäyttäjien yhteistoimintaa kaikin mahdollisin tavoin. Näillä keinoilla loppukäyttäjien ja kehittäjien välisiä vuorovaikutusongelmia voidaan vähentää. WWW-tekniikapolijaiset työkalut, kuten intranet ja erilaiset portaalit, ovat hyviä työkaluja käynnissä olevien projektien tiedon jakamiseen.

Wiegersin [85] artikkelissa toisena vaatimusmäärittelyn ongelmakohtana mainitaan puutteellinen loppukäyttäjän sitoutuminen. Tämän tilanteen voi tunnistaa siitä, että lähtökohdat vaatimuksille tulevat loppukäyttäjän korvaavilta tahoilta. Näitä ovat esimerkiksi loppukäyttäjien esimiehet, markkinointi-ihmiset tai ohjelmistokehittäjät. Toinen oire on, kun ohjelmistokehittäjät joutuvat tekemään vaatimuksia koskevia päätöksiä puutteellisilla pohjatiedoilla ja ilman oikeaa näkökulmaa ongelmasta. Jos vaatimuksia ei kerätä oikeilta käyttäjäryhmiltä, on todennäköistä, että joku ei ole tyytyväinen valmiiseen tuotteeseen. Wiegers kertoi tilanteesta, jossa loppukäyttäjä ei hyväksynyt kelvotonta tuotetta ensimmäisessä käyttöönotossa, jolloin näki ensimmäisen kerran tuotteen. Tämä on äärimmäinen ja kallis esimerkki siitä, mihin puutteellinen loppukäyttäjän sitoutuminen voi johtaa.

Wiegers [85] esittelee ratkaisuksi puutteelliseen loppukäyttäjän sitouttamiseen kaikkien erilaisten käyttäjäryhmien löytämisen. Eri käyttäjäryhmien käyttötavat eroavat toisistaan esimerkiksi käyttömäärän, käytettyjen toimintojen ja käyttöoikeuksien suhteen. Käyttäjäryhmistä kannattaa tunnistaa kyseessä olevan käyttäjäryhmän erityisen taitava edustaja, jolta saadaan käyttäjävaatimuksia. Hän voi antaa todenmukaista tietoa laatuominaisuuksista ja vaatimusten tärkeysjärjestyksestä. Tämä lähestymistapa toimii erityisen hyvin organisaation sisäisissä projekteissa. Markkinavetoisessa ohjelmistokehityksessä käyttäjäryhmät ja niiden edustajat ovat helpommin tunnistettavissa. Vaatimusmäärittelytyössä kannattaa pyrkiä luomaan avoimuuteen ja yhteistyöhön kannustava ilmapiiri.

Nasirin ja Sahibuddin [61] analyysisessä kirjallisuuskatsauksessa loppukäyttäjän osallistuminen oli kuudenneksi yleisin kahdestakymmenestäkuudesta menestekijästä. Se oli mainittu kahdessakymmenessä tapauksessa neljästäkymmenestäkolmesta tutkimukseen valitusta lähteestä.

Alladi ja Vadari puhuvat kirjallisuuskatsauksessaan [3] erilaisten sidosryhmien sitouttamisesta projektiin ja erilaisten sidosryhmien merkityksen tunnistamisen tärkeydestä. Tarvetta sidosryhmien sitouttamiseen voi syntyä seuraavista syistä [3]:

- Asiakkaiden vaatimukset ovat epäselviä
- Kaikkia projektiin liittyviä sidosryhmiä ei osata tunnistaa etukäteen
- Kaikki sidosryhmät eivät ymmärrä ongelmaa
- Ei ymmärretä minkälaisen vaikutuksen sidosryhmän jäsen voi aiheuttaa projektille
- Ei ymmärretä riskejä etukäteen
- Ei ole selvää kuvaa toteutettavasta tuotteesta tai palvelusta
- Väärinymmärretyt asiakkaiden odotukset
- Ristiriitaiset odotukset sidosryhmien keskuudessa
- Sosiokulttuuriset tekijät
- Maantieteelliset ympäristötekijät

Ratkaisuksi ehdotetaan kuusiaskelista sitouttamismetodia, jonka vaiheet ovat [3]:

1. Vuorovaikutusstrategia
2. Käyttäjärhmiön kartoitus
3. Analyysi ja suunnitelma
4. Sitouttamiskykyjen vahvistaminen
5. Prosessin suunnittelu ja toteutus ja toiminta
6. Arviointi ja mittaus

Alladi ja Vadari [3] kuusiaskelisen sitouttamismethodin ensimmäinen vaiheen, joka on vuorovaikutusstrategia, tarkoitus on kiinnittää huomiota, että eri sidosryhmille suunnatussa viestinnässä otetaan huomioon sidosryhmän tarpeet. Toinen vaihe, joka on käyttäjärhmiön kartoitus, auttaa ymmärtämään käyttäjärhmiön väliset erot, ja ratkaisemaan mahdollisia ristiriitaisia odotuksia käyttäjärhmiön välillä. Kolmannessa vaiheessa, joka on analysointi ja suunnittelu, keskitytään seuraamaan tilannetta ja asettamaan tavoitteita tilanteen mukaan. Neljännessä vaiheessa huolehditaan siitä, että projektiryhmä ottaa sidosryhmät mukaan projektityöhön. Viides vaihe, prosessin suunnittelu ja toteutus, pitää sisällään sidosryhmien tarpeiden ja odotusten mukaisen toteutusmallin valitsemisen. Kuudennessä vaiheessa on olennaista mitata tehtyjen päätösten vaikutusta, ja pitää huoli, että tehdyistä päätöksistä opitaan.

Alladi ja Vadari [3] tunnistavat ehdotetun methodin hyötyjä: Sidosryhmien mukaan ottaminen alkuvaiheessa helpottaa sidosryhmien toiveiden päättymistä vaatimukseen ja siten selkiyttää lopputuotetta tai palvelua. Sidosryhmien sitouttaminen auttaa kestävään ja pitkäaikaiseen yhteisymmärrykseen sidosryhmien kanssa. Sitouttaminen on ennakoiva toimenpide epäselvyyksien välttämiseen ja toisten ymmärtämiseen. Sidosryhmäkarta auttaa ymmärtämään eri sidosryhmien päämäärien ristikkäisyyksiä. Jatkuva viestintä projektin sisällä auttaa sidosryhmiä tuomaan toiveensa esiin. Sidosryhmäkarta auttaa tunnistamaan ja luokittelemaan sidosryhmiä ja niiden päämääriä.

El Emamin ja muiden [29] tutkimus käsittelee myös loppukäyttäjän sitouttamista ohjelmistoprojektiin. Tutkimuksessa havaittu muutosvastarinnan vähentyminen uutta ohjelmistoa kohtaan, kun loppukäyttäjä on ollut mukana vaatimusmäärittelyn tekemisessä, on tärkeä ja vähemmän huomioitu tulos. El Emam ja Madhavji [28] myös totesivat, että loppukäyttäjän sitoutuminen on erityisen tärkeää silloin kun vaatimukset ovat epäselviä.

Tarawnehin ja muiden [82] tutkimus paljasti loppukäyttäjän osallistumisen erittäin olennaiseksi osaksi ohjelmistoprojektia. Heidän mukaansa loppukäyttäjä voi auttaa oman työnkuvansa avaamisella selkeyttämään projektin maalia. Tarawnehin ja muiden tutkimus liittyy loppukäyttäjän sitouttamisen maaliin liittyviin ongelmiin.

Wiegersin [85] lähestymistapa on aika perinteinen ja edustaa vesiputoustyylistä prosessimallia. Hänen tutkimuksestaan paistaa läpi ajatus, että vaatimusmäärittely tehdään ohjelmistoprojektin alussa ja siinä pyritään löytämään kaikki mahdolliset ongelmat. Wiegers painottaa ongelmien ennakoimista, kun taas ketterät menetelmät keskittyvät ongelmaan reagoimiseen. Siitä huolimatta ongelmat ovat samat kuin tuorempien ketterien menetelmien kanssa. Ajatus siitä, että ongelmallisen tilanteen tunnistaa, kun vaatimukset eivät tule suoraan loppukäyttäjältä, vastaa käyttäjälähtöisen suunnittelun ideologiaa. Wiegersin pahana pitämät puutteelliset tiedot ja niiden selvitys kesken ohjelmistoprojektin kuuluvat olennaisena osana käyttäjälähtöiseen suunnitteluun, jossa riittävien tietojen saanti jätetään loppukäyttäjän sijasta ohjelmistokehittäjän vastuulle. Myös Wiegersin ajatukset loppukäyttäjän sitouttamisen suhteen ovat hyvin nykyaikaisia, ja niissä painotetaan, että kaikki erilaiset käyttäjäryhmät tulee tunnistaa. Kaikkien käyttäjäryhmien tunnistus on myös suoraan käyttäjälähtöisen suunnittelun lähtökohta. Wiegersin ajattelumalli vaikuttaakin olevan vesiputousmalli, johon hän ottaa mukaan myös käyttäjälähtöisessä suunnittelussa hyväksi havaittuja malleja.

Myös Alladin ja Vadarin [3] tutkimuksesta käy selvästi ilmi, että loppukäyttäjän sitouttaminen on tärkeää. Loppukäyttäjän osallistuminen auttaa määrittämään projektille maalin sekä selkiyttämään epäselviä vaatimuksia. Heidän mukaansa siis loppukäyttäjän läsnäolo auttaa sekä maalin määrittämiseen että vaatimusten elämiin liittyviin ongelmiin.

Loppukäyttäjien osallistuminen ohjelmistoprojekteihin todettiin edellä kuvatuissa tutkimuksissa äärimmäisen tärkeäksi. Loppukäyttäjän sitouttamiseen liittyvien ongelmien yleisyys viittaa selvästi siihen, että loppukäyttäjät eivät osallistu tarpeeksi ohjelmistoprojektien läpivientiin.

3.2.4 Vaatimukset eivät ole pysyviä

Keil ja muut [48] ryhmittelevät riskit, joiden merkitys ohjelmistoprojektin kokonaisriskiin on suuri, mutta myös projektipäällikön vaikutusmahdollisuudet ovat suuret. Nämä riskit syntyvät epävarmuuksista ja epäselvyyksistä, jotka liittyvät ohjel-

mistoprojektin sisältöön ja vaatimukseen. Tämän ryhmän riskejä kuvaavat kysymykset: Mitä kuuluu projektin sisältöön ja mitä ei? Mikä toiminnallisuus on olennaista onnistumisen kannalta ja mikä on vain mukava saada? Olennaista on tietää, mitä ollaan rakentamassa, ja miten tarve voi muuttua ajan kuluessa. Esimerkkejä tähän ryhmään kuuluvista riskeistä ovat vaatimusten väärinymmärrykset ja muutoshallinnan epäonnistuminen. Kummatkin mainituista ongelmista ovat vakavia, mutta projektipäällikön hallittavissa. Näitä riskejä voidaan hallita välttämällä epäselvyyksiä ja parantamalla muutosten hallintaa. Alussa on usein hankala löytää täsmälleen kaikki vaatimukset, jotka kuuluvat projektin sisältöön, mutta projektin edetessä sen rajausta ja vaatimukset tarkentuvat. Samalla myös loppukäyttäjien odotukset muuttuvat realistisemmiksi. Keilin ja muiden [48] mukaan yksi tapa hallita projektin rajausta on määrittää, mikä ei sisälly projektiin. Projektipäällikkö voi myös ehkäistä projektin rajauksen elämistä painottamalla muuttuvan rajauksen vaikutuksia kustannuksiin ja aikatauluun. Projektipäällikön tulee olla myös valmis rajaamaan, mitkä vaatimukset on pakko toteuttaa niistä vaatimuksista, jotka olisi mukava saada toteutettua. Toinen tapa hallita tämän ryhmän riskiä on antaa vastuuta projektin etenemisestä loppukäyttäjille. Loppukäyttäjille pitäisi muistuttaa jatkuvasti, että heidän roolinsa on korvaamaton määriteltäessä järjestelmän toimintoja. Myös tämän ryhmän riskien hallintaan liittyy vahvasti asiakkaan sitouttamiseen.

El Emamin ja Korun [27] mukaan liian usein muuttuvat vaatimukset ja projektin rajausta ovat suurin syy ohjelmistoprojektien keskeytyksiin. Kolmasosassa tutkimukseen osallistuneista keskeytyneistä ohjelmistoprojekteista muuttuvat vaatimukset ja projektin rajausta oli osallisena keskeytykseen.

Myös Wiegiers [85] artikkelissaan vaatimusmäärittelyn ongelmista tunnistaa epäselvät vaatimukset. Wiegiers esittää kolmitasoisien vaatimusmallin, jossa hän määrittelee liiketoimintavaatimukset, käyttäjävaatimukset ja toiminnalliset vaatimukset. Eri käyttäjäryhmien vaatimukset tulevat tämän portaikon eri tasoilta, joten eri käyttäjäryhmien kesken vaatimukset tarkoittavat hyvin erilaisia asioita. Eri lukijat voivat siis ymmärtää vaatimukset eri tavalla, ja jokainen lukija kuvittelee oman tulkintansa oikeaksi. Väärinymmärretyt vaatimukset paljastuvat liian myöhään, mistä seuraa ylimääräisiä kustannuksia. Ongelman läsnäolon voi tunnistaa siitä, kun vaatimusmäärittelystä ei saa tarpeellisia tietoja vaatimusten toteuttamiseen. Jos vaatimuksista ei pysty johtamaan vaatimusten oikeellisuuden varmistavia testitapauksia, niin vaatimukset ovat puutteellisia. Ohjelmistokehittäjä voi luulla, että vaatimukset ovat täydellinen ja lopullinen kuvaus tarpeesta, vaikka niin ei välttämättä

ole. Äärimmäinen oire puutteellisista vaatimuksista on, että ohjelmistokehittäjä joutuu kysymään vaatimuksista loppukäyttäjiltä, tai arvaamaan vaatimusten merkitystä. Tämä voidaan huomata vasta, kun tuotteen kehitys on jo pitkällä, ja kehitettävän tuotteen olemus on eronnut alkuperäisestä tarpeesta. Tässä tapauksessa tuotteen korjaaminen tarpeen mukaiseksi vaatii paljon työtä ja on kallista.

Hyvin konkreettiseksi ratkaisuksi epäselviin vaatimuksiin Wiegers [85] esittää, että moniselitteisiä sanoja, kuten minimoi, maksimoi, optimoi, nopea, käyttäjäystävällinen, helppo, yksinkertainen, usein, normaali, tavanomainen, iso, intuitiivinen, toimintavarma, viimeisintä teknologiaa edustava, parannettu, tehokas tai joustava, tulisi välttää. Myös ilmauksia ja, tai ja jne. ei tulisi käyttää. Sana tukea sopii myös huonosti vaatimusmäärittelytyöhön, sillä sen toteutumista ei voi varmentaa. Vaatimusmäärittelytyössä tulisi olla selkeitä vaatimuksia, joiden avulla kuvataan jokin tehtävä. Vaatimusmäärittelytyössä voi jättää asioita päätettäväksi myöhemmin, mutta ne tulee ratkaista ennen toteutusta. Vaatimusten tarkastamiseen kannattaa koota ryhmä, joka edustaa mahdollisimman monenlaisia näkökulmia. Tällaisia näkökulmia voisi edustaa esimerkiksi loppukäyttäjä, testaaja ja ohjelmistokehittäjä. Yksi hyvä tapa on kirjoittaa testitapaukset ennen vaatimuksia. Testitapauksien kirjoittaminen käyttäjätapauksien avulla auttaa ymmärtämään, kuinka ohjelmiston tulisi toimia tietyissä tilanteissa. Testitapauksien kirjoittaminen auttaa paljastamaan moniselitteisyydet ja puuttuvat tiedot vaatimuksista ja auttaa luomaan vaatimuksia, jotka mahdollistavat hyvien testitapauksien tekemisen. Myös prototyyppien luominen auttaa tekemään vaatimuksista konkreettisempia kuin pelkästään kirjoitetut vaatimukset. Epäselvistä vaatimuksista kannattaa luoda osittainen, alustava tai mahdollinen toteutus, jonka avulla puuttuvat tiedot paljastuvat. Myös analyttiset mallit, kuten tietovuokaaviot, oliomallit, luokkakaaviot, rajapintamallit, tilasiirtymäkaviot ja vuorovaikutuskartat tarjoavat vaihtoehdoisen kuvan järjestelmään ja paljastavat sitä kautta puutteellisia tietoja.

Zhu ja muut [90] tutkivat vaatimuksien muuttumista jaoteltuna eri käyttäjäryhmien kesken. He eivät erottele tutkimuksessa loppukäyttäjää, vaan keräävät sekä asiakkaan että loppukäyttäjän asiakas termin alle.

Zhun ja muiden [90] tutkimuksessa taulukon 3.3 vaatimusmuutoksista suurin osa oli suoraan käyttäjätyyppien eroavista tarpeista lähtöisin. Tutkimuksessa tunnistettiin viisi erilaista asiakasryhmää. Projektin päättäjä pitää huolen projektin resursseista, koordinoinnista, sisällöstä ja päättää projektin hyväksymisestä. Projektin teknologiajohtaja tuntee käyttöympäristön sekä käytettävän teknologian ja huoleh-

Taulukko 3.3: Syyt vaatimuksien muuttumiselle [90]

Syy vaatimuksen muuttumiselle	Muutoksien lukumäärä
Vaatus on huonosti ilmaistu	102
Vaatus voidaan ymmärtää kahdella tavalla	66
Loppukäyttäjän tarve muuttuu	52
Kehittäjän vaatimusten laiminlyönti ja muutokset	40
Loppukäyttäjä peruu vaatimuksen	18
Loppukäyttäjän toimintamalli muuttuu	5
Järjestelmästä itsestään johtuvat syyt	13
Ohjelmistosta ja laitteistosta johtuvat syyt	4

tii projektin vetämisestä sekä vaatimusten muuttumisesta. Projektipäällikkö vastaa tarvittavan tiedon hankinnasta ja projektin etenemisen varmistamisesta. Loppukäyttäjä on lopullinen tuotteen käyttäjä, jolta saadaan tarvittava tieto tarvittavista tuotteen ominaisuuksista ja käyttöympäristöstä. Ylläpitäjä on henkilö, jonka vastuulle jää valmiin tuotteen ylläpito. Tässä tutkimuksessa riskialttiimpien muutoksen alkuunpanijaksi tunnistettiin projektin päättäjä, sillä hänen päätöksiensä vaikutus voi olla laaja esimerkiksi rajausten muuttuessa ja testaaja, jonka vaatimusmuutokset tulevat myöhäisessä vaiheessa.

Zhun ja muiden [90] tutkimuksen yhteenvedona vaatimusten muuttuminen on väistämätöntä. Olennaista on opastaa aikaisessa vaiheessa vaatimusten määrittelyssä, eriyttää eri käyttäjäryhmien vaatimusmuutokset huolella ja tehdä riskiarviointia, jonka avulla vaatimusmuutosten vaikutukset voidaan pitää mahdollisimman pieninä.

Keilin ja muiden [48] tutkimus yhdistää toisiinsa loppukäyttäjän sitouttamisen tärkeyden ja vaatimusten muuttuvuuden. Tutkimuksessa selvitetään onko ohjelmistoprojektiin sitoutunut loppukäyttäjä ratkaisu muuttuviin vaatimuksiin. Tutkimuksen mukaan erityisesti paljon muuttuvia tai tuntemattomia tekijöitä sisältävässä ympäristössä loppukäyttäjien sitouttaminen projektiin on tärkeää. Tästä voidaan päätellä, että jos loppukäyttäjän sitouttaminen epäonnistuu, niin on myös suuri riski, että muuttuvista vaatimuksista seuraa ongelmia.

Wiegers [85] ryhmittelee tutkimuksessaan vaatimuksia ja loppukäyttäjiä, ja myös liittää loppukäyttäjien läsnäolon ja vaatimukset hyvin läheisesti yhteen. Hänen mu-

kaansa on olennaista, että vaatimukset pysyvät vakaampina silloin, kun ne on kerätty mahdollisimman laajapohjaisesti loppukäyttäjiltä tunnistuen erilaisten käyttäjäryhmien erilaiset tarpeet. Tämä sama ajattelumalli on käytössä myös käyttäjälähtöisessä suunnittelussa.

Myös Zhu ja Yeshengin [90] kantavat huolta erilaisten käyttäjäryhmien tunnistamisesta. Heidän tutkimuksensa terä on erilaisten käyttäjäryhmien tunnistaminen ja niiden erilaiset tarpeet.

3.3 Yhteenveto

Tässä luvussa käsitellyissä tutkimuksissa on osoitettu, että suuri osa ohjelmistoprojektien ongelmista johtuu loppukäyttäjän vähäisestä osallistumisesta ohjelmistoprojektiin. Loppukäyttäjä tuntee käyttötarpeen parhaiten ja on sitä kautta paras asiantuntija kertomaan tulevan järjestelmän vaatimuksista. Loppukäyttäjä kuuluu olennaisena osana toimintaympäristön kartoitukseen ja maalin selvitykseen, sekä auttaa tuomaan vakautta projektiin, jossa vaatimukset muuttuvat. Loppukäyttäjän sitoutumisen puuttuminen ohjelmistoprojektissa on myös itsessään tunnistettu ongelmaksiksi.

Tämän luvun tutkimukset osoittavat, että loppukäyttäjä ei osallistu tarpeeksi ohjelmistoprojektin läpivientiin. Näissä tutkimuksissa on myös esitetty erilaisia ratkaisuja loppukäyttäjän osallistumisen parantamiseksi, mutta loppukäyttäjältä itseltään ei ole kysytty, miksi ohjelmistoprojektiin osallistuminen koetaan hankalana.

4 Loppukäyttäjän suhtautuminen vaatimusmäärittelytyöhön

Tässä luvussa käydään läpi syitä, jotka vaikuttavat loppukäyttäjän osallistumiseen vaatimusmäärittelytyöhön. Luvussa käydään läpi aiempaa kirjallisuutta aiheesta ja nostetaan esiin muutamia ongelmakohtia loppukäyttäjän osallistumisessa. Huomionarvoista on, että vaikka tässä luvussa pyritään keskittymään loppukäyttäjän näkökulmaan, niin aikaisempi tutkimus käsittelee loppukäyttäjän osallistumisen ongelmia useasti kehittäjien näkökulmasta. Myös näitä ongelmia on otettu mukaan tähän lukuun laajemman ymmärryksen saavuttamiseksi.

Alaluvussa 4.1 esitellään lyhyesti aikaisemmat loppukäyttäjän osallistumista vaatimusmäärittelytyöhön käsittelevät tutkimukset. Alaluvussa 4.2 käsitellään palautteen saamista loppukäyttäjältä ja siihen liittyviä ongelmia. Alaluvussa 4.3 käsitellään loppukäyttäjän saavutettavuuteen liittyviä asioita. Alaluvussa 4.4 käsitellään loppukäyttäjän motivaatiota. Alaluvussa 4.5 käsitellään loppukäyttäjän motivaation ylläpitämiseen liittyviä ongelmia. Alaluvussa 4.6 käsitellään loppukäyttäjän osallistumisesta kehittäjille aiheutuvia haasteita. Alaluku 4.7 käy läpi loppukäyttäjän osallistumisen ongelmia koskevan tutkimuksen. Alaluku 4.8 on yhteenveto.

4.1 Tutkimuksia loppukäyttäjän osallistumisesta

Baroudi ja muut [7] tutkivat aikaisemmissä tutkimuksissa todistettua väitettä, jonka mukaan loppukäyttäjän osallistuminen ohjelmiston suunnitteluun sekä lisää järjestelmän käyttöä että parantaa käyttäjätyytyväisyyttä. Tämän olettaman he nimeävät perinteiseksi malliksi. Lisäksi tutkimuksessa testataan kahta vaihtoehtoista mallia [7]:

- Loppukäyttäjän osallistuminen lisää käyttäjätyytyväisyyttä sekä suoraan että osallistumisesta seuranneen järjestelmän käytön lisääntymisen kautta
- Loppukäyttäjän osallistuminen lisää järjestelmän käyttöä sekä suoraan että osallistumisesta lisääntyneen käyttäjätyytyväisyyden kautta

Doll ja Torkzadeh [23] käsittelevät loppukäyttäjän osallistumista esittelemällä

yhteyttä loppukäyttäjän osallistumisinnokkuuden ja havaitun osallistumisen tason välillä. Tutkimus on erityisesti keskittynyt tiedonhallintajärjestelmiin, mutta se on hyvin yleistettävissä myös laajemmin ohjelmistotuotantoon. Tutkimuksen ajatuksena on, että osallistumisinnokkuuden tulisi vastata osallistumisen tasoa, sillä se on olennaisempaa onnistumisen kannalta kuin pelkkä aktiivinen osallistuminen.

Grudinin tutkimus [33] käsittelee yleisiä esteitä, jotka liittyvät loppukäyttäjän osallistumiseen ohjelmistoprojektiin. Hänen mukaansa monet näistä esteistä voidaan johtaa organisaatorakenteisiin ja kehityskäytäntöihin, jotka ovat kehittyneet ennen vuorovaikutteisia järjestelmiä. Tutkimus pätee yleisesti ihmisen ja ohjelmiston välisen vuorovaikutuksen kehittämiseksi, mutta se on erityisen merkityksellinen työryhmäsovelluksien suhteen.

El Emamin ja Madhavjin [28] jo luvussa 3 mainittu tutkimus sivuaa myös oikean, eniten hyötyä vaatimusmäärittelytyöhön tuovan loppukäyttäjän löytämisen vaikeutta. Heidän mukaansa ohjelmistoprojektiin sopivalla loppukäyttäjällä on hyvät tietotekniset taidot, paljon tietoa toimintaympäristöstä, liiketoimintaprosesseista ja organisaation tarpeista, hyvät tiedot järjestelmäkehityksen prosesseista, hyvät tiedot informaatiojärjestelmien toteutuksen suunnittelusta ja hallinnasta ja hyvät vuorovaikutustaidot. Lisäksi osallistuvalla loppukäyttäjällä tulee olla uskottavuutta organisaatiossa, sillä hän markkinoi kehitettävää järjestelmää myös oman osastonsa sisällä muille loppukäyttäjille. Osallistuvilla loppukäyttäjillä tulee olla toimivalta ottaa mukaan myös muita loppukäyttäjiä muista osastoista.

Sioukas [77] käsittelee omassa tutkimuksessaan loppukäyttäjän osallistumisen hyötyjä ohjelmiston tai palvelun muokkaamisessa. Hän käy läpi loppukäyttäjän osallistumisen, projektin tehokkuuden, toimittajan tehokkuuden palvelun tuottamisessa ja projektin monimutkaisuuden välisiä yhteyksiä, kuten myös loppukäyttäjän osallistumisen mahdollistavia tekijöitä. Tutkimus on tapaustutkimus, jonka yhteydessä on toteutettu kysely ja johon liittyy puheverkkoratkaisun muokkaaminen. Analyysi paljastaa kaksi projektin monimutkaisuuteen liittyvää tekijää: projektin koon ja loppukäyttäjän kokemattomuuden. Tutkimuksen tulos tukee sitä, että loppukäyttäjän osallistuminen on positiivisesti yhteydessä projektin tehokkuuteen. Tutkimus antaa viitteitä siitä, että palveluhenkinen toimittaja voi edistää loppukäyttäjän osallistumista, ja siten vaikuttaa projektin menestykseen. Tutkimus myös paljastaa, että kokemattomuus lisää loppukäyttäjän osallistumista, ja osoittaa siten, että oppimishalu on tärkeä motivaattori loppukäyttäjän osallistumiselle.

Heinbokelin ja muiden [38] tutkimus on yksi harvoista loppukäyttäjän osallistu-

miseen kriittisesti suhtautuvista tutkimuksista. Poikittaistutkimus käsittää 29 kaupallista ohjelmistoprojektia, ja siinä analysoidaan loppukäyttäjän osallistumisen hyötyjä ja haittoja. Tutkimuksessa otetaan tarkastelun alle sekä loppukäyttäjän osallistuminen että käyttäjäkeskeisyys. Loppukäyttäjän osallistuminen pitää sisällään loppukäyttäjän osallistumisen tiimin toimintaan. Käyttäjäkeskeisyys tarkoittaa loppukäyttäjän ottamista toiminnan keskipisteeksi. Tutkimuksen tarkoituksena on tarkastella loppukäyttäjän osallistumisen ja käyttäjäkeskeisyyden ongelmia.

Käyttäjien osallistuminen järjestelmän suunnitteluun jo aikaisessa vaiheessa on todettu hyödylliseksi, mutta siihen liittyy haasteita. Wilson ja muut [86] käsittelevät loppukäyttäjän osallistumista järjestelmän suunnitteluun helpottavia tekijöitä. Tämä tutkimus on tapaustutkimus räätälöidyn ohjelmiston suunnittelusta. Tutkimuksessa käydään seikkaperäisesti läpi suunnitteluvaiheessa löydetty loppukäyttäjän osallistumiseen liittyvät helpottavat ja haittaavat tekijät. Näitä ongelmia ovat esimerkiksi loppukäyttäjien valitseminen ja yhteydenotto, loppukäyttäjien motivointi, tapaamisten järjestäminen ja hallinta ja loppukäyttäjän työpanoksen ohjaaminen. Tutkijat haastattelivat erilaisia käyttäjäryhmiä, ja osallistuivat suunnitteluprosessiin osallistumattomina tarkkailijoina, sekä vertailivat haastattelujen tuloksia omiin havaintoihinsa.

Browne ja Ramesh [14] käsittelevät tutkimuksessaan vaatimusmäärittelytyön parantamista kognitiiviselta kannalta. He käsittelevät lyhyesti myös loppukäyttäjän motivaatioon liittyviä asioita. Heidän lopputuloksensa on, että vaatimusmäärittelytyön parantamiseen ei ole yksinkertaista ratkaisua. Vaatimusmäärittelytyöhön osallistuvien pitää ymmärtää toimintaympäristöä ja ihmisten käyttäytymistä ja kognitiivisia kykyjä.

Nuseibehin ja Easterbrookin [62] mukaan organisaatioiden tulisi lisätä monialaista perehdytystä vaatimusmäärittelytyötä suorittaville henkilöille. Perehdyttämisen tärkeys korostuu, mikäli organisaatio ei palkkaa ulkopuolista asiantuntijaa avustamaan vaatimusmäärittelytyössä. Vaatimusmäärittelytyötä suorittavan henkilön tulee kyetä toimimaan yhteistyössä eri sidosryhmien kanssa ja erilaisen teknisen taustan omaavien henkilöiden kanssa. Vaatimusmäärittelytyöhön soveltuvan henkilön tulee pystyä ymmärtämään loppukäyttäjän vaatimuksia, olivatpa ne kuvattu teknisesti riittämättömällä tasolla tai toisaalta hyvinkin teknisellä kielellä.

Coughlan ja Macredien [20] mukaan loppukäyttäjien vaatimusten ja toimintaympäristön ymmärtäminen moninkertaistaa ohjelmistoprojektin onnistumisen todennäköisyyden. Jotta hankittavasta ohjelmistosta saataisiin mahdollisimman hyödyll-

linen ja käytettävä, täytyy hankinnan syistä olla selkeä käsitys. Ongelma, jota ohjelmistolla pyritään ratkaisemaan, on siten vaatimusten paras lähde. Ongelman ymmärtäminen varmistaa ohjelmiston käytettävyyden ja ymmärtämistä voidaan lisätä hyödyntämällä käyttäjäkeskeisiä lähestymistapoja vaatimusten keräämiseen.

Kujalan [51] perusteellisessa kirjallisuuskatsauksessa käydään läpi loppukäyttäjän osallistumisen hyödyt ja haasteet. Hänen mukaansa loppukäyttäjän osallistuminen on laajalti hyväksytty toimintatapa helppokäyttöisten järjestelmien tekemiseen. Tutkimuksen tarkoituksena on selvittää, mitä loppukäyttäjän osallistuminen kehitykseen pitää sisällään ja esitellä kolme tutkimussuuntaa, jotka käsittelevät erilaisen loppukäyttäjän osallistumiseen liittyvien menetelmien hyötyjä ja haittoja. Tutkimus keskittyy erityisesti ohjelmistoprosessin aikaiseen vaiheeseen. Aikaisempien tutkimuksien mukaan loppukäyttäjän osallistumisella on yleisesti positiivinen vaikutus erityisesti käyttäjätyytyväisyyteen. On myös todisteita siitä, että loppukäyttäjän pitäminen ensisijaisena tietolähteenä on hyvä tapa kerätä vaatimuksia. Loppukäyttäjän rooli prosessissa pitää kuitenkin valita huolella. Pitää löytää kustannustehokkaampia tapoja kartoittaa loppukäyttäjien tarpeita ja vaatimuksia loppukäyttäjän oikeassa toimintaympäristössä.

Kujala ja muut [52] tutkivat loppukäyttäjän osallistumisen vaikutusta vaatimusten laatuun ja projektin onnistumiseen. Loppukäyttäjän osallistuminen kuuluu olennaisena osana hyödyllisten ja käytettävien järjestelmien kehitykseen, sekä vaikuttaa positiivisesti järjestelmän menestykseen ja käyttäjätyytyväisyyteen. Tutkimus on kysely- ja haastattelututkimus loppukäyttäjän osallistumisen hyödyistä vaatimusmäärittelytyöhön. Kyselyyn osallistui 18 ohjelmistoammattilaista, jotka työskentelevät Suomessa 13:sta yrityksessä ohjelmistokehitykseen liittyvissä projekteissa. Lisäksi haastateltiin kolmessa yrityksessä kahdeksaa ohjelmistoammattilaista. Sitä, miten loppukäyttäjä osallistuu ohjelmistoprojektiin ja miten se vaikuttaa projektiin, tutkittiin laadullisella ja määrällisellä analyysillä. Loppukäyttäjän osallistumisen ohjelmistoprojektiin alkuvaiheesta asti todettiin olevan harvinaista, mutta johtavan parempilaatuisiin vaatimuksiin. Loppukäyttäjän tiedon tuomisen projektiin todettiin olevan yhteydessä projektin onnistumiseen.

Harris ja Weistroffer [35] ovat tehneet erittäin perusteellisen kirjallisuuskatsauksen loppukäyttäjän osallistumisen järjestelmän kehitykseen yhteydestä järjestelmän menestykseen. Tutkimuksen hakuehdoilla löytyi n. 200 artikkelia, joista tutkimuksen valintakriteerien perusteella valikoitui 28. Kirjallisuuskatsauksen lopputuloksena oli, että loppukäyttäjän osallistuminen ohjelmistoprojektiin on yhteydessä ohjel-

mistoprojektin menestykseen.

Bano ja Zowghi [6] käsittelevät erittäin kattavassa analyttisessä kirjallisuuskatsauksessaan loppukäyttäjän osallistumisen yhteyttä ohjelmistoprojektin onnistumiseen. Kirjallisuuskatsauksessa käytiin läpi aluksi 2776 artikkelia, joista lopulta mukaan valikoitui 37. Näistä 37 löytyi lisää lähteitä, ja lisäksi aikaisemmat kirjallisuuskatsaukset sisälsivät katsauksesta puuttuvia lähteitä, joten lopulliseksi artikkelimääräksi tuli 87. Kirjallisuuskatsauksessa löydetty selkeimmät osallistumisen hyödyt olivat [6]:

- Käyttäjäytyväisyys paranee
- Hyväksymistestaus helpottuu
- Muutosvastarinta vähenee
- Loppukäyttäjän asenne uutta järjestelmää kohti paranee
- Loppukäyttäjien ja kehittäjien välinen kommunikaatio paranee
- Vaatimusten laatu paranee
- Projektin lopputulos paranee
- Käyttöympäristön ymmärtämys lisääntyy

4.2 Palautteen saaminen loppukäyttäjältä

Palautteen keräämiseen loppukäyttäjiltä liittyy Grudinin [33] mukaan haasteita. Palautetta voi kerätä loppukäyttäjiltä vapaamuotoisesti tai vikailmoitusten ja muutospyyntöjen kautta. Vikailmoituksen ja muutospyyntöt keskittyvät pääasiassa siihen, mikä on tärkeää järjestelmän toiminnan kannalta, kuten laitteiston luotettavuuteen ja korkean tason toimintoihin, eikä juurikaan käyttöliittymään liittyviin asioihin. Usein ei ole selvää, kuka voisi päättää ryhmän toimintaan vaikuttavista vaatimuksista. Vähäinen kerätty tieto ei kulje kehittäjille asti. Järjestelmätuki ylläpitää tuotetta ja hoitaa yksittäisten loppukäyttäjien ongelmia ja samalla estää yhteydenpidon loppukäyttäjien ja kehittäjien välillä. Alkuperäisen tuotteen kehittäjät siirtyvät uusiin tuotejulkaisuihin, projekteihin tai organisaatioihin.

Grudin [33] toteaa, että palautteen laajuus voi vaihdella markkinointitavan tai tuotteen käytön mukaan. Applen tapaiset yritykset, joiden tuotteista suuren osan

ostavat suoraan loppukäyttäjät eikä johdon tai informaatioteknologian asiantuntijat, hyötyvät erityisen äänekkästä käyttäjäjoukosta. Yleisesti palautteen puute saattaa kuitenkin olla suurin este hyvälle käyttöliittymäsuunnittelulle ja mahdollisimman vähän virheitä sisältävälle järjestelmälle. Kehittäjät eivät voi käyttää kaikkea aikaansa loppukäyttäjien tarpeiden selvittämiseen. Puutteellinen asiakaspalaute estää nykyisen tuotteen parantamisen sekä käyttäjälähtöisen suunnittelun tärkeyden ymmärtämisen. Kehittäjät eivät yleensä ymmärrä loppukäyttäjän tuskaa.

Grudinin [33] mukaan on hyvä muistaa, että kehittäjät ovat tottuneet prosessiin, jossa toivottujen vaihtoehtojen välillä tehdään valintoja. Käyttöliittymäparannuksille annetaan enemmän painoarvoa, mikäli kehittäjät ovat tietoisia niistä kauaskantoisista seurauksista, joita huonon suunnittelun hyväksyminen aiheuttaa. Esimerkkeinä voidaan mainita seuraavat tyypilliset vaihtoehdot [33]: *”Tämä toteutus säästää 10 kilotavua, mutta on vähemmän modulaarinen.”*, *”Tämä malli on on hieman nopeampi, mutta sen toteutus kestää kuukauden pitempään.”* ja *”Tässä sirussa on kaksi korttipaikkaa lisää, mutta se lisää tuotteen myyntihintaa 500\$.”*. Jokainen edellä mainitusta vaihtoehdoista vaatii päätöksen. Kun päätös on tehty, niin sen seuraukset jätetään taka-alalle. Tässä ympäristössä myös käyttöliittymään liittyvät päätökset ovat päätöksiä muiden joukossa. Kehittäjät siirtyvät päätöksiänsä jälkeen nopeasti eteenpäin. Käyttöliittymään liittyvät päätökset ovat kehittäjille samanarvoisia kuin mitkä muut tahansa päätökset, ja ilman loppukäyttäjien palautetta ne on helppo jättää taakseen päätöksen jälkeen. Nämä päätökset voivat vaikuttaa tuhansiin loppukäyttäjiin tuotteen eliniän aikana, mutta kehittäjät eivät tiedä seurauksista. Käyttöliittymä on erittäin tärkeä, mutta kehittäjät eivät saa siitä palautetta, sillä kun tuote on valmis ja toimitettu, niin tuotteen kehittäjät siirtyvät uusiin projekteihin.

Palautteen puute koskee Grudinin [33] mukaan erityisesti työryhmäsovelluksia, sillä palautteen kerääminen ja läpikäynti yhden käyttäjän sovelluksissa on helpompaa ja vie vähemmän aikaa. Ryhmän prosessit ovat monimutkaisempia ja hitaampia ja ympäristön muuttujat vaikuttavat enemmän. Onnistumiset ja epäonnistumiset ovat hankalampia ymmärtää. Tämän johdosta kalliit virheet toistetaan yhä uudelleen.

4.3 Loppukäyttäjän saavutettavuus

El Emamin ja Madhavjin [28] mukaan loppukäyttäjät, jotka olisivat sopivimpia osallistumaan vaatimusmäärittelytyöhön, ovat kaikista kiireisimpiä ja hankalimpia saa-

vuttaa. Myös loppukäyttäjän osaston ja tietojärjestelmäosaston väliset ristiriidat, tai jopa avoin vihamielisyys, voivat vaikeuttaa loppukäyttäjien osallistumista.

Myös Wilson ja muut [86] käsittelevät tutkimuksessaan loppukäyttäjän saamista projektiin. Ensimmäinen este loppukäyttäjän saamisessa projektiin voi olla johtoporras. Loppukäyttäjän osallistumista voi helpottaa, jos se liittyy organisaation sisäisiin kehitysprosesseihin, tai sitä kannattaa joku sidosryhmien jäsen. Tässä tutkimuksessa käyttöliittymäsuunnittelija ajoi loppukäyttäjän osallistumista ja hän oli hyvin motivoitunut käyttämään suunnitteluprosessia, johon loppukäyttäjä osallistuu aktiivisesti. Käyttöliittymäsuunnittelija sai johtoportaan ymmärtämään loppukäyttäjän osallistumisen tärkeyden. Suunnitteluprosessi oli helppo hyväksyä, koska edellinen yritys tämän järjestelmän kehittämiseen oli epäonnistunut, koska siinä ei otettu tarpeeksi huomioon erilaisten käyttäjäryhmien tarpeita. Käyttöliittymäsuunnittelija ei ollut projektissa mukana toteutusvaiheeseen asti. Tässä yhteydessä kävi ilmi loppukäyttäjän osallistumista ajavan projektiryhmän jäsenen tarpeellisuus. Loppukäyttäjät eivät juurikaan osallistuneet projektin toteutusvaiheeseen lukuun ottamatta muutamia muodollisia kommentteja. Loppukäyttäjille ei ollut enää selvää, kuinka he voisivat tarjota mielipiteitään. Heistä tuntui, että heidän toiveitaan ei enää otettu huomioon. Yksi loppukäyttäjä kommentoi, että hän toivoisi näkevänsä järjestelmässä muutoksia, mutta ei tiedä kenelle niistä pitäisi puhua. Projektissa oli tarkoitus ottaa loppukäyttäjä mukaan koko projektiin, mutta lopulta sovittiin, että loppukäyttäjät eivät osallistu yhden osakokonaisuuden suunnitteluun. Tämän osakokonaisuuden piti olla niin yksinkertainen, että loppukäyttäjältä saatavaa tietoa ei tarvita. On mielenkiintoista, että tämän osakokonaisuuden merkitys kutistui kehityksen aikana, ja sitä ei ole otettu lainkaan käyttöön.

Wilsonin ja muiden [86] tutkimuksessa oli vähän aikaa loppukäyttäjän osallistumisen järjestämiseen. Ensimmäisessä vaiheessa suunnittelua piti tehdä paljon päätöksiä, ja ne muodostuivat esteeksi loppukäyttäjän osallistamiseen. Käyttöliittymäsuunnittelija määritteli prosessin niin, että siihen osallistuu vain tietty määrä loppukäyttäjiä, ja valitsi halutut henkilöt. Haastatelluksi tuli 14 loppukäyttäjää, jotka edustivat kaikkia työhön liittyviä ryhmiä. Monet haastatelluista olivat oman osastonsa vanhempia asiantuntijoita ja heillä oli hyvä näkemys oman alueensa työtehtävistä. Alkuperäisistä haastateltavista kolme kutsuttiin mukaan myöhempiin mallinnus- ja suunnittelutehtäviin. Nämä kolme valittua edustivat kolmea suurinta työhön liittyvää ryhmää ja sen lisäksi heillä oli työn mukanaan tuomaa ammattitaitoa ja kokemusta sekä intoa osallistua projektiin.

Wilsonin ja muiden [86] tutkimuksen käyttöliittymäsuunnittelija koki valinneesaa loppukäyttäjät hyvin. Oikeiden loppukäyttäjien valitseminen on olennaista käyttöliittymäsuunnittelijalle, sillä on hänen oman edun mukaista, että valitut loppukäyttäjät tietävät paljon työstään. Loppukäyttäjien haastattelu kuitenkin paljasti, että muiden loppukäyttäjien tarpeet oli laiminlyöty. Erityisesti kokemattomat, osaikaiset ja väliaikaiset työntekijät oli jätetty huomiotta, mutta yllättäen myös vanhemmat esimiehet kokivat samoin. Haastatteluissa selvisi, että nämä työntekijät eivät näe hyötyä sovelluksen käyttämisestä. Laiminlyötyjen käyttäjäryhmien mielestä järjestelmä ei auttanut heitä työnsä tekemisessä tai lisäsi heidän työkuormaansa. Käyttöliittymäsuunnittelija teki parhaansa ja valitsi loppukäyttäjät, joilla oli tietoa useista työtehtävistä. Tämä johti siihen, että työntekijät, jotka tekevät työtä päivittäin, eivät osallistuneet suunnitteluun.

Kujalan [51] mukaan oikeiden loppukäyttäjien löytäminen voi olla haastavaa. Banon ja Zowghin [6] kirjallisuuskatsauksessa yksi yleisimmistä ongelmista loppukäyttäjän osallistumisen suhteen oli ajan puute. Tämä löytyi neljästä artikkelista valitun 87:n joukosta. Myös johdon tuen puute oli havaittu ongelma.

4.4 Loppukäyttäjän motivaatio

Dollin ja Torkzadehin [23] mukaan onnistuneen osallistumisen edellytys on motivaatio osallistumiseen. Loppukäyttäjän osallistamisen ja ohjelmistoprojektin onnistumisen välillä on tunnistettu olevan yhteys, mutta loppukäyttäjän kiinnostusta osallistumiseen on tutkittu varsin vähän. Loppukäyttäjien osallistuminen ohjelmistokehitykseen voi parantaa osaamista käyttöönottossa, lisätä ymmärrystä ohjelmiston toiminnasta ja sitouttaa loppukäyttäjää ohjelmiston käyttöön. Yleisesti osallistuminen ohjelmistoprojektiin on hyödyllisintä silloin, kun loppukäyttäjien sallitaan osallistua sen verran kuin he haluavat. Sopivaa osallistumisen määrää on hankala määrittää. Se on tasapainossa silloin, kun loppukäyttäjä ei koe joutuvansa osallistumaan liikaa tai liian vähän, sillä molemmat ääripäät voivat vähentää tyytyväisyyttä. Doll ja Torkzadeh [23] toteavat, että ohjelmistotuotantoon liittyvä kirjallisuus on jättänyt huomioimatta miksi loppukäyttäjät osallistuvat ohjelmistoprojekteihin ja miten heitä voitaisiin motivoida osallistumaan.

Grudinin [33] mukaan organisaation omana projektina toteutettavan ohjelmistoprojektin kehittäjillä on ainakin jollakin tasolla sama esimies kuin tulevilla loppukäyttäjillä. Tämä ei päde silloin, kun loppukäyttäjät eivät kuulu samaan organisaatioon.

tioon, jolloin loppukäyttäjillä voi olla hankalampi käyttää aikaa projektiin. Lisäksi mahdolliset loppukäyttäjät eivät ole yhtä motivoituneita, sillä he tietävät, että heistä ei välttämättä tule järjestelmän käyttäjiä. Loppukäyttäjän osallistumisen korostaminen projektissa on tunnistettu merkittäväksi sekä organisaation ulkoisissa että sisäisissä projekteissa. Työnantaja ja loppukäyttäjät suhtautuvat yleensä positiivisesti osallistumiseen, jollei se vie juurikaan aikaa.

Grudin [33] toteaa, että loppukäyttäjä ei välttämättä ole motivoitunut osallistumaan, jos hän ei näe hyötyvänsä osallistumisesta. Tämä koskee erityisesti suurempia järjestelmiä, monia työryhmäsovelluksia tai sovelluksia, jotka hyödyttävät johtajia. Tähän ryhmään kuuluu esimerkiksi projektin- tai työnhallintasovellukset, erilaiset päätöksiä tukevat sovellukset, palaverien hallintaan käytetyt sovellukset ja jopa puhesovellukset. On huomattavaa, että joskus työryhmäsovelluksien kategoriaan luokitellut menestyneet sovellukset, kuten sähköposti, versionhallintajärjestelmät ja tietokannat, eivät hyödytä pelkästään johtajia. Suurimman motivaation puutteen aiheuttaa, jos loppukäyttäjä kokee työnsä uhatuksi uuden järjestelmän lisäämän tehokkuuden takia.

Projektiin liittyvä ohjelmiston tulevien loppukäyttäjien iso määrä ja tunnistamisen hankaluus aiheuttavat Grudinin [33] mukaan sen, että muutaman parhaiten sopivan loppukäyttäjän löytäminen voi olla hankalaa. Vaarana on myös, että pientä ryhmää osallistuvia loppukäyttäjiiä yleistetään liikaa, tai projekti jumiutuu loppukäyttäjän kieltäytymiseen. Lisäksi työ on vasta alussa, kun loppukäyttäjä on saatu onnistuneesti osallistumaan suunnitteluun. Riippumatta siitä, mistä ideat ovat lähtöisin, ne pitää saada vietyä kehitysprosessin läpi ominaisuuksiksi. Loppukäyttäjän osallistuminen tähän prosessiin saattaa auttaa auttaa projektin onnistuneeseen läpivientiin, vaikkei se ole helppoa edellä kuvatuista syistä.

Sioukas [77] määrittelee loppukäyttäjän osallistumisen suorana osallistumisena tuotteen kehityksen suunnittelu- ja toteutusvaiheisiin. Tämän prosessin aikana loppukäyttäjän organisaatio ja toimittaja suorittavat sarjan ongelmanratkaisua lopullista tuotetta varten. Myös Sioukas päätyy tutkimuksessaan siihen lopputulokseen, että loppukäyttäjän osallistumisen lisäämisellä on positiivinen vaikutus projektin lopputulokseen.

Sioukas [77] toteaa loppukäyttäjien asiantuntijuuden ja aikaisemman kokemuksen ohjelmistoprojekteista vaikuttavan heidän motivaatioonsa osallistua projekteihin uudelleen. Myönteinen kokemus aikaisemmista osallistumiskerroista lisää motivaatiota osallistua uudelleen ja vähentää siten riskien määrää projekteissa. Hyvä

toimittajayhteistyö ja erityisesti toimittajan asiakaslähtöisyys voi vaikuttaa myönteisesti loppukäyttäjän asenteeseen projektia kohtaan. Asiakasorganisaatio sitoutuu sekä toimittajaan että ohjelmistoon paremmin, kun luottamus toimittajan kykyyn palvella on hyvä.

Sioukas [77] esittää väitteen, jonka mukaan loppukäyttäjät eivät välttämättä ole motivoituneita osallistumaan pieniin projekteihin, jotka eivät ole kustannuksiltaan suuria ja jotka eivät tuota heille lisäarvoa. Sioukas arvelee loppukäyttäjien osallistumisen määrän olevan korkeampi, kun kehitettävä ohjelmisto on laaja ja moniulotteinen.

Wilson ja muut [86] käsitelivät tutkimuksessaan loppukäyttäjän asennetta ja motivaatiota. Heidän mukaan on selvää, että loppukäyttäjien asenne ja motivaatio oli ratkaiseva tekijä heidän osallistumisessaan ohjelmistoprojektiin. Motivoituneet loppukäyttäjät tulivat helposti valituksi osallistumaan. Suunnittelijat olivat innokkaita valitsemaan loppukäyttäjät, jotka osoittivat kiinnostusta projektia kohtaan. Loppukäyttäjän motivaation, tai sen puutteen, syitä tulee ymmärtää, sillä on tärkeää, että projektiin saataisiin mukaan myös loppukäyttäjiiä, jotka ovat vähemmän innokkaita osallistumaan. Loppukäyttäjän suhtautumisen taustalta löytyi monia erilaisia syitä [86]:

- Osa loppukäyttäjistä oli ollut aiemmin tilanteessa, jossa he kokivat, että heidän mielipiteitään ei oltu kysytty, ja he eivät olleet myöskään tyytyväisiä lopputulokseen.
- Joidenkin loppukäyttäjien taustalla vaikutti aikaisemmat osastojen väliset ristiriidat, ja nämä loppukäyttäjät kokivat parantavansa omaa asemaansa osallistumalla.
- Yksi loppukäyttäjä halusi varmistaa, että järjestelmään tulee toiminnallisuus, joka auttaa häntä omassa työssään.
- Eräs esimies halusi osallistua, sillä hän koki olevansa vastuussa projektista.

Loppukäyttäjän innokkuuden puutteeseen osallistua vaikutti puolestaan seuraavat syyt [86]:

- Yksi loppukäyttäjä ei halunnut osallistua, sillä hän ei kokenut osallistumista olennaiseksi työnsä kannalta.
- Jotkut loppukäyttäjät eivät olleet tietoisia siitä, että he olisivat voineet osallistua, tai kokivat, että heillä ei ollut mitään annettavaa.

- Eräät loppukäyttäjät olivat epävarmoja omasta osaamisestaan, ja kertoivat, että heidän ei-tekni­sen taustansa vuoksi heillä ei ollut mitään annettavaa projektille.

Projektiin paljon antaneet loppukäyttäjät kokivat hyötynensä osallistumisesta ja olivat siksi kiinnostuneista osallistumaan myös jatkossa. He olivat lisäksi vapaaehtoisia lisätietojen hankintaan sekä innostivat muita osallistumaan. Erityisesti eräs loppukäyttäjä levitti tietoa muille osastollaan ja rohkaisi heitä jakamaan omia mielipiteitään suunnitellusta järjestelmästä.

Loppukäyttäjän motivaation tärkeys tuli esiin myös kirjallisuuskatsauksissa. Kujala [51] löysi kirjallisuuskatsauksessaan loppukäyttäjien motivoinnin olennaiseksi loppukäyttäjän osallistumisen kannalta. Tämä tuli ilmi kahdessa kirjallisuuskatsauksen artikkeleista. Myös Bano ja Zowghi [6] löysivät systemaattisessa kirjallisuuskatsauksessaan loppukäyttäjän motivaation puutteen yleisimmäksi loppukäyttäjän osallistumiseen liittyvistä ongelmista. Loppukäyttäjän motivaation puute löytyi kymmenestä valituista 87:stä artikkelista. Muita yleisiä ongelmia olivat loppukäyttäjän asenne ja käytös.

4.5 Loppukäyttäjän osallistumisen ylläpito

Wilson ja muut [86] käsittelevät tutkimuksessaan loppukäyttäjän osallistumisen jatkuvuuden ylläpitoa. Suunnittelijat kokivat, että loppukäyttäjän olisi helpompi myötävaikuttaa lopputulokseen, jos he ovat osallistuneet läpi koko projektin, sillä silloin heillä olisi parempi käsitys prosessista ja tuotteen nykytilasta. Tämä oli selvästi havaittavissa tehtävien määrittelyssä, johon käytettiin samaa lähestymistä sekä projektin alussa että lopussa. Loppukäyttäjät, joka olivat olleet mukana alusta asti huomasivat, että osallistuminen alusta asti helpotti osallistumista myöhemmissä vaiheissa. Yksi loppukäyttäjä ei osallistunut keskusteluun tulevista tehtävistä, ja tyytyikin kommentoimaan vain ilmiselviä virheitä mallissa.

Toinen tärkeä tekijä loppukäyttäjän osallistumisen ylläpitämisessä on tehokas tapaamisten järjestäminen. Wilsonin ja muiden [86] tutkimuksessa suunnittelijat yrittivät useita taktiikoita helpottamaan loppukäyttäjän tunnetta tapaamisiin osallistumisen suhteen ja osoittamaan, että loppukäyttäjän osallistumista arvostetaan. Suunnittelijat valmistautuivat ensimmäiseen tapaamiseen loppukäyttäjien kanssa ottamalla selvää loppukäyttäjän nimestä ja roolista. Yhden loppukäyttäjän mukaan oli tärkeää, että suunnittelijat eivät olleet tuomitsevia, ja että tapaamiset eivät olleet

pelottavia tai liian muodollisia. Lisäksi hänelle oli osaston politiikan takia erittäin tärkeää vastausten luottamuksellisuus. Tapaamisissa nousi esiin myös, että loppukäyttäjät antoivat liian nopeasti periksi suunnittelijan mielipiteille, sillä he luottivat liikaa suunnittelijan asemaan ja osaamiseen.

Wilsonin ja muiden [86] tutkimuksessa löydettiin viitteitä, että käyttöliittymäsuunnittelija johdatteli loppukäyttäjiä. Yhtä loppukäyttäjää häiritsi se, että hän ei tiennyt käytettävän teknologian suunnittelun rajoitteita, ja hänen mukaansa olisi ollut hyödyllistä, jos käytettävä teknologia olisi ollut saatavilla suunnittelun mahdollisuuksien kartoittamiseen. Kehittäjien ja loppukäyttäjän välisenä välittäjänä toimineen käyttöliittymäsuunnittelijan rooli koettiin merkittäväksi. Yhdessä tapaamisessa tämä käyttöliittymäsuunnittelija ei toiminut tehokkaasti muiden projektien aiheuttamien paineiden takia. Tämä tapaaminen epäonnistui, eikä sen tavoitteita saavutettu. Tapaamiseen osallistuneet olivat ärtyneitä tapaamisen suhteen.

Wilsonin ja muiden [86] tutkimuksessa suunnittelijat haastattelivat aluksi loppukäyttäjiä yksitellen, ja selvittivät miten loppukäyttäjien tehtävät voisivat muuttua uuden järjestelmän myötä. Tämän jälkeen suunnittelijat toivat kaikki loppukäyttäjät yhteen ja kokosivat näkemykset yhdeksi suureksi tehtävämalliksi. Tällä varmistettiin kaikkien loppukäyttäjien mielipiteiden esiintuonti. Yhden loppukäyttäjän mukaan tämä toimi hyvin. Sen lisäksi, että se auttoi häntä ilmaisemaan mielipiteensä, se auttoi häntä selvittämään omat ajatuksensa tarpeistaan ennen ryhmätapaamista. Saman loppukäyttäjän mukaan oli myös selvää, että loppukäyttäjät, jotka eivät osallistuneet ennen ryhmätapaamista, eivät saaneet omia ideoitaan esille. Ryhmätapaamisessa paljastui myös ongelmia. Eräs loppukäyttäjä ei halunnut ilmaista mielipidettään, sillä hän pelkäsi muiden osallistujien reaktioita. Toinen loppukäyttäjä koki hankaluuksia tuoda oman osastonsa ajatuksia esiin, sillä toinen osasto oli osallistunut suuremmalla loppukäyttäjämäärällä. Suunnittelijoiden mukaan ristiriidat käyttäjäryhmien välillä tuotiin esille, mikä haittasi aluksi loppukäyttäjien osallistumista suunnitteluprosessiin. Tämä johti lopulta siihen, että jotkut ongelman soviteltiin ja päädyttiin uusiin hyödyllisiin suunnittelumalleihin.

Wilson ja muut [86] nostavat esiin myös huomion kohdistamisen merkityksen. Aluksi suunnittelijat kysyivät loppukäyttäjiltä osallistumisen helpottamiseksi heidän työhönsä liittyviä kysymyksiä. Loppukäyttäjät tehtävien suunnitteluun valittiin huolella niin, että heillä oli laaja ymmärrys erilaisista rooleista ja työtehtävistä omalla osastollaan. Näitä loppukäyttäjiä pyydettiin myös ottamaan kantaa muilta loppukäyttäjiltä jo kerättyyn tietoon. Suunnittelijat käyttivät lukuisia menetel-

miä, kuten lehtiötaulua, valkotaulua ja post-it-lappuja huomion kohdistamiseen tapamisissa. Huomiota ohjattiin myös antamalla loppukäyttäjille pääsy jo aiemmin suunniteltuihin ratkaisuihin. Esimerkiksi paperiprototyypitapaamiseen osallistuneelle loppukäyttäjälle annettiin pääsy tehtävämalliin muistutukseksi tehdyistä ratkaisuksista. Joskus loppukäyttäjät kehittivät omia tapoja ohjata huomiota ja parantaa tuottavuutta. Tästä esimerkkinä loppukäyttäjä, joka tuli paperiprotoilutapaamiseen oman tarkastuslistan kanssa, tai loppukäyttäjä, joka halusi paperikopion suunnittelumallista, jotta voisi tarkastella sitä kotona.

Myös suunnittelumallin esitys, käytettävä mediat ja työkalut ovat Wilsonin ja muiden [86] mukaan olennaisia loppukäyttäjän osallistumisen ylläpitämisessä. Tutkimuksessa käytettiin erilaisia esitystapoja suunnitelman esittämiseen ja helpottamaan loppukäyttäjän osallistumista. Käyttöliittymäsuunnittelija uskoi, että valkotaulu on hyödyllinen, sillä se tarjoaa helpon tavan kohdistaa huomiota, mahdollistaa loppukäyttäjien yhteistyön ja mallin muokkaamisen. Jotkut loppukäyttäjät olivat erittäin aktiivisia valkotaululla ja paperiprotosessioissa, kun taas toiset loppukäyttäjät olivat passiivisempia osallistumaan. Eräs loppukäyttäjä kertoi, että ottaisi mieluummin kopion mukaansa kotiin, jossa hän voisi tutustua siihen omassa rauhassa. Tämä loppukäyttäjä ehdotti, että hänen kopionsa voisi jakaa myös muille kommentteja varten. Suunnittelijat kokivat, että paperiprototyyppi on nopea ja helposti muutettava. Yhden loppukäyttäjän mukaan paperiprototyypit olivat hyödyllisiä käyttöliittymän visualisointiin, kun taas toinen loppukäyttäjä ei pitänyt paperiprotoista, ja siksi hän ei osallistunut paperiprotoiluun. Tämä loppukäyttäjä piti enemmän ohjelmistoprototyypeistä, sillä hänen mukaansa käyttöliittymän vuorovaikutuksen arviointi ei onnistu paperiprotolla. Wilsonin ja muiden [86] tutkimuksessa tehtävämallin kuvaamisen menetelmä kehitettiin ensimmäisen loppukäyttäjän kanssa. Siitä johtuen tulevat loppukäyttäjät joutuivat hyväksymään kehitetyn mallin. Joidenkin mielestä malli oli hyvä, mutta kaikki eivät pitäneet siitä. Sovelluksen kehittäjän mukaan kaikki loppukäyttäjät eivät ymmärtäneet mallia, sillä se oli erittäin laaja, ja siihen tutustuminen vaatisi aikaa. Loppukäyttäjille ei aina annettu tarvittavaa aikaa tutustua malliin ennen suunnittelupalavereita. Yhden loppukäyttäjän mukaan malli ei onnistunut kuvaamaan tehtävän kaikkia näkökohtia, ja tutkimuksessa löytyi viitteistä siitä, että jotkut loppukäyttäjät eivät ymmärtäneet kaikkia näkökohtia. Tämä johti vääriin olettamuksiin, joita jouduttiin korjaamaan myöhemmin.

Brownen ja Rameshin [14] mukaan loppukäyttäjän motivaatio on tärkeää vaa-

timusmäärittelytyössä. Loppukäyttäjä voi ulkoisien syiden takia olla haluton osallistumaan vaatimusmäärittelytyöhön. Näitä syitä voi olla esimerkiksi yrityksen politiikka tai organisaation kannustuksen puute. Haluttomuus osallistua voi johtua myös henkilökohtaisista periaatteista tai taloudellisesta hyödystä, itsensä suojelemisesta, itsensä säästämisestä tai siitä, että vastaukset arvioi ja näkee joku muu. Lisäksi organisaation ja ryhmän asenne voi vaikuttaa loppukäyttäjän intoon osallistua vaatimusmäärittelytyöhön. Loppukäyttäjä voi esimerkiksi pelätä loukkaavansa jotakuta tai huonontaa omaa asemaansa. Äärimmillään tämä voi johtaa itsesensuuriin tai siihen, että loppukäyttäjä päättää voivansa tulla toimeen huonolla ratkaisulla. Itsesensuuri voi johtaa epätäydellisiin vaatimuksiin, joidenkin vaihtoehtojen jättämiseen huomiotta ja valheelliseen kuvaan yksimielisyydestä.

4.6 Kehittäjän motivointi

Grudin [33] nostaa esiin myös kehittäjien motivoinnin aiheuttamat ongelmat. Kaikkien kehittäjien tulee olla sitoutuneita loppukäyttäjän osallistumiseen kehitykseen. Teoriassa yksi kehittäjä voi keskustella loppukäyttäjien kanssa ja tuoda keskustelun lopputuloksen kehitysprosessiin, mutta iteratiivinen ohjelmistokehitys vaatii laajemman sitoutumisen, sillä esimerkiksi protoilu ja testaus voi vaatia kehittäjien tukea ja asiakkaiden arviointia samaan aikaan. Lisäksi johdon pitää olla valmis antamaan resursseja, ja myös muiden organisaation jäsenten apua voidaan tarvita helpottamaan yhteydenottoa loppukäyttäjään.

Grudinin [33] mukaan suurin osa kehittäjistä kannattaa periaatteessa loppukäyttäjän osallistumista, mutta toimiva loppukäyttäjän osallistuminen vaatii sitoutumista. Ohjelmistoprojektin jäsenten halua sitoutua voi laskea monet syyt, heiltä voi esimerkiksi puuttua myötätuntoa ja auttamishalua kokemattomien ja ei-tekniisten loppukäyttäjien suhteen. Projektin jäsenten ja loppukäyttäjien kanssakäymistä voi haitata myös erilaiset arvot, erilaiset työskentelytyylit tai jopa eri kieli. Kehittäjät ovat yleensä nuoria, rationaalisia, idealistisia ja omaavat samanlaisen akateemisen koulutustaustan. Usein kehittäjillä ei ole paljon kokemusta tai ymmärrystä monien järjestelmän loppukäyttäjien erilaisiin työtilanteisiin ja asenteisiin. Nämä syyt voivat ylittää loppukäyttäjän osallistumisen hyvät vaikutukset, varsinkin kun loppukäyttäjän osallistumisen saattaa olla epätarkkaa ja edetä hitaasti.

4.7 Käyttäjän osallistumisen negatiivinen vaikutus

Heinbokel ja muut [38] tutkivat loppukäyttäjän osallistumisen vaikutusta lopputulokseen. Loppukäyttäjän osallistuminen jaettiin käyttäjäkeskeisyyteen ja loppukäyttäjän osallistumiseen. Lopputuloksen mittarina käytettiin ohjelmistoprosessin laatua, lopputuotteen laatua ja projektin onnistumista.

Heinbokelin ja muiden [38] tutkimuksen mukaan projektit, joissa loppukäyttäjät osallistuivat paljon, tuottivat vähemmän innovaatioita, olivat joustamattomampia, olivat tiimityön osalta huonompia ja päättyivät huonompaan lopputulokseen. Käyttäjäkeskeisyys vaikutti positiivisesti stressin määrään, mutta käyttäjäkeskeisyydellä oli negatiivinen vaikutus tiimin tehokkuuteen ja tiimin sisäiseen vuorovaikutukseen. Tutkimuksen mukaan käyttäjäkeskeiset projektit ja projektit, joihin loppukäyttäjä osallistuu, eivät onnistuneet täydellisesti. Käyttäjäkeskeisyyden ja loppukäyttäjän osallistumisen voisi luulla aiheuttavan hankaluuksia isommissa ja helpommin epäonnistuvissa projekteissa, mutta tutkimukseen mukaan tällaista riippuvuutta ei huomattu.

4.8 Yhteenveto

Erityisesti Wilson ja muut [86] käsitelivät perusteellisesti loppukäyttäjän osallistumiseen liittyviä ongelmia myös loppukäyttäjän näkökulmasta. He tarkastelivat loppukäyttäjän motivaatiota osallistua ohjelmistoprojektiin. Erityisen mielenkiintoinen huomio oli, että osa loppukäyttäjistä kokee olevansa teknisesti liian osaamattomia osallistua vaatimusmäärittelytyöhön. Tämä on iso ongelma, sillä loppukäyttäjien ei tarvitse olla teknisesti osaavia, vaan edustaa oman työnsä osaajaa, jonka kautta saadaan arvokasta tietoa järjestelmän tarpeesta ja vaatimuksista. Toinen arvokas huomio Wilsonin ja muiden [86] tutkimuksessa on, että osa loppukäyttäjistä kokee ohjelmistoprojektiin osallistumisen pelottavana ja vaikeana asiana ja on jopa epävarma omasta mielipiteestään pelätessään muiden mielipiteitä.

Tässä luvussa esitellyt tutkimukset osoittavat, että loppukäyttäjän vaatimusmäärittelytyöhön osallistumiseen vaikuttavat lukuisat tekijät. Loppukäyttäjän motivaation kannalta osallistumista on tutkittu vähän, ja luultavasti sen taustalta löytyy ilmiöitä, joita on syytä tutkia tarkemmin.

5 Tutkimus- ja analysointimenetelmät

Tässä luvussa esitellään opinnäytteen tutkimusstrategia, aineistonhankintamenetelmät ja analyysimenetelmät. Tutkimusstrategiana on tapaustutkimus, jonka aineistonhankintamenetelmiksi valitsimme sekä määrällisen lomakekyselyn että laadullisen haastattelututkimuksen. Aineistojen analysoinnissa hyödynnetään ristiintaulukointia ja teemoittelua. Alaluvussa 5.1 esitellään tutkimusstrategia, alaluvussa 5.2 käydään läpi aineistonhankintamenetelmät ja alaluvussa 5.3 käydään läpi aineiston analyysimenetelmät. Alaluvussa 5.4 kuvataan tutkimuksen toteutus.

5.1 Tutkimusstrategia

Corbin ja Strauss [18, s. 21] esittävät tutkimuskysymyksen muotoilun olevan haastavaa etenkin kokemattomille tutkijoille. Tutkimuskysymyksen tulisi liittyä tutkijan omiin kiinnostuksen kohteisiin, jotta motivaatio tutkimusta kohtaan säilyisi korkeana. Tutkimusongelma voi löytyä heidän mukaansa neljästä kategoriasta:

- Toisen tutkijan tai ohjaajan tarjoamat tutkimusongelmat
- Teknisestä tai ei-teknisestä kirjallisuudesta johdetut tutkimusongelmat
- Henkilökohtaisesta tai työhön liittyvistä kokemuksista johdetut tutkimusongelmat
- Tutkimuksessa esille nousevat tutkimusongelmat

Tutkimuksen suunnittelussa täytyy pitää mielessä miten kerättävä aineisto auttaa tutkimuskysymyksen mukaisen ongelman selvittämisessä [76, s. 71]. Eisenhardtin [26] mukaan tutkimuskysymyksen pitäminen mielessä tapaustutkimuksen aineiston keruun ja teoriaksi saattamisen aikana on tärkeää, jotta tutkimuksen ydin ei unohdu. Tutkimusaineiston määrä ja sen läpikäynti voi kasvaa liian suureksi ja työlääksi, jos tutkimuskysymys ei ole riittävän tarkka. Hyödyllisintä on saada täsmennettyä mahdollisimman aikaisessa vaiheessa tutkimuskysymys ja tutkimuksen rakenne, jotta voidaan myös varmistua, että ne ovat suunniteltuun tutkimukseen soveltuvat.

Tapaustutkimus

Metsämuurosen [58, s. 222] ja Yinin [87, s. 1] mukaan tapaustutkimuksen voi kuvata yksinkertaisesti jonkin parhaillaan olemassa olevan tapahtuman tutkimukseksi. Tapaus voi olla ihminen, ihmisryhmä tai suurempi kokonaisuus kuten oppilaitos. Eisenhardtin [26] mukaan tapaukset tulisi aina valita huomioiden, että ne on mahdollista toistaa myöhemmin ja että niiden avulla voidaan tuottaa uutta teoreettista tietoa. Metsämuuronen [58, s. 223] toteaa tapaustutkimuksen etuina olevan sen toiminnallisuuden ja todenmukaisuuden. Tapaustutkimuksessa tutkittava ilmiö on usein toiminnallinen ja tutkimuksen tuloksia sovelletaan myös käytännössä. Tapaustutkimuksessa syntyvä aineisto on todenmukaista, mutta vaikeasti analysoitavaa, koska tapaustutkimus perustuu tutkittavan henkilön omakohtaisiin kokemuksiin. Tapaustutkimuksessa syntyvä aineisto on kuvailevaa, josta voidaan tehdä erilaisia tulkintoja.

Yinin [87, s. 28] mukaan tapaustutkimuksen suunnittelu tulee aloittaa aina taustateorian kartoittamisella ja opiskelulla. Tapaustutkimus edellyttää tutkimusongelmaan liittyvän aikaisemman tutkimusaineiston tuntemusta, jonka jälkeen oma aineistonkeruu on mahdollista aloittaa. Robsonin [69, s. 140] mukaan valmisteluvaiheeseen kuuluu myös tutkijoiden aktiivinen läsnäolo tutkimuksen suunnittelussa, mikäli tutkimukseen osallistuu useampi kuin yksi tutkija. Jos tutkijoita on kaksi tai useampi, heidän on suositeltavaa ottaa samankaltaiset roolit tutkimuksessa, jotta he ovat aktiivisesti tietoisia tutkimuksen tilanteesta kaikissa vaiheissa.

Silvermanin [76, s. 98] ja Tuomen ja Sarajärven [83, s. 71] mukaan tapaustutkimuksen aineistoa voidaan kerätä hyödyntäen useita erilaisia tapoja kuten haastatteluja, kyselyjä, havainnointia, tai dokumentteihin perustuvaa tietoa. Tutkimuksesta riippuen on mahdollista hyödyntää yhtä tai useampaa menetelmää. Useamman aineistonkeruumenetelmän hyödyntäminen antaa kattavamman pohjan tutkimuksen rakenteelle ja hypoteeseille. [26], [31, s. 13]

5.2 Aineiston hankintamenetelmät

Määrällinen tutkimus

Määrällinen eli kvantitatiivinen analyysi on liitetty positivismiin, mutta positivismin ongelmien takia siihen liitetään nykyään enemmänkin jälkipositivismi. Jälkipositivismi on hajanaisempi käsite kuin positivismi, joka on joukko yhtenäisiä oppeja,

ja se pyrkii vastaamaan positivismin ongelmiin. Esimerkiksi positivismi lähtee oletuksesta, että havainnoinnin kohde näyttäytyy samanlaisena havainnoitsijasta riippumatta, kun taas jälkipositivismin mukaan kaikki havainnoijat eivät näe kohdetta samalla tavalla. Jälkipositivismissa mikään tutkimus ei ole täydellinen, ja se pyrkii seuraamaan kaikista todennäköisintä teoriaa. Näin jälkipositivismi yrittää ottaa huomioon positivismin heikkouden havainnoitsijan täydellisyydestä. Sekä positivismi että jälkipositivismi pyrkivät kuitenkin ymmärtämään maailmaa teorioiden kautta. Molemmissa pyritään kehittämään teorioita, ja testataan niitä havainnoinnin kautta. [69, ss. 20–23]

Useimmat tutkimukset tuottavat jollain tavalla numeroilla mitattavaa tietoa, jonka takia määrällisen tutkimuksen menetelmät ovat olennaisen tärkeitä. Määrällisen tutkimuksen menetelmiin kuuluu useita keinoja yksinkertaisesta tietojen järjestelystä ja monimutkaisiin tilastollisiin analyyseihin. [69, ss. 413–415]

Ensimmäinen vaihe määrällisessä analyysissä on tutkittavan tiedon järjestely käsiteltävään muotoon. Tutkimusta suunniteltaessa tulee ottaa huomioon tiedon käsittelyn vaatimukset, jotta kerätty tieto on hyödynnettävissä. Ylimääräisiä vaiheita tiedon käsittelyssä tulee välttää. [69, ss. 415–418]

Tiedon käsittely voidaan jakaa kahteen malliin: selvittävä ja varmistava. Selvittävässä mallissa tutkitaan mitä kerätty tieto kertoo, kun taas varmistavassa mallissa lähdetään olettamuksesta, ja katsotaan vastaavatko tulokset oletusta. Selvittävässä mallissa on tärkeää saada yleiskuva kerätystä tiedosta, ja ajatus siitä, mitä tiedosta voi päätellä. [69, ss. 419–420]

Helppo tapa käsitellä kerättyä tietoa on järjestää se esiintymistiheyden mukaan. Esiintymistiheys voidaan esittää joko lukumäärinä tai prosenttiosuuksina. Tällaista tietoa voidaan jäsenellä taulukoihin, pylväsdiagrammeihin, graafeihin tai piirasdiagrammeihin. [69, ss. 421–423]

Laadullinen tutkimus

Gillham [31, ss. 10–11], Metsämuuronen [58, s. 223] ja Patton [64, s. 33] kuvaavat laadullisten menetelmien keskittyvän jonkin ilmiön ymmärtämiseen, mitä ihmiset voivat kertoa tai mitä he tekevät. Laadulliset menetelmät auttavat ymmärtämään, mitä tapahtuu. Ne auttavat ongelmien ymmärtämisessä ja niiden selittämisessä. Jos ihmisen käyttäytymistä, ajatuksia tai tunteita halutaan ymmärtää, täytyy tutkimus tehdä samassa ympäristössä, missä luontaisesti toimitaan. Tuomi ja Sarajarvi [83, s. 28] mieltävät omien sanojensa mukaan laadullisen tutkimuksen erityisesti ymmär-

tävänä tutkimuksena. Ymmärtäminen on heidän mukaansa eläytymistä ja aikomuksellisuuden tutkimista.

Eskolan ja Suorannan [30, ss. 18–20] mukaan laadullisessa tutkimuksessa keskitytään pieneen määrään tapauksia ja pyritään analysoimaan niitä mahdollisimman perusteellisesti. Aineistoa ei mitata sen määrässä vaan laadussa. Laadullisen tutkimuksen perusluonteeseen kuuluu hypoteesittomuus ja teorian rakentaminen jo aineiston keruun aikana. Ennako-olettamukset pitäisi pystyä rajaamaan ajatuksista pois, vaikka on otettava huomioon että tekemämme havainnot ovat aina sidoksissa aikaisempiin kokemuksiin. Laadullisessa analyysissä tutkijan pitäisi yllättyä, oppia uutta ja löytää uusia näkökulmia tutkimuksen kuluessa. Corbin ja Strauss [18, s. 16] esittävät, että tutkijan tulisi olla utelias ja luova tutkimusta kohtaan sekä luottaa ihmisten tulkinnassa omiin vaistoihinsa.

Corbinin ja Straussin [18, s. 27] mukaan laadullisen tutkimuksen luonteeseen kuuluu monet erilaiset aineistonkeruutavat. Tutkija voi käyttää lähteinään esimerkiksi haastatteluja, kyselyitä, havainnointia, videoita, dokumentteja ja uutisia. Tutkimuksessa on mahdollista hyödyntää useampaa kuin yhtä aineistonkeruumenetelmää. Vaikka aineistoa voidaan kerätä usealla eri tavalla ja jopa kohtuullisen helposti, ei tutkijan tulisi kuitenkaan aloittaa aineiston keruuta valmistautumatta tutkimukseen. Haastateltavista henkilöistä saatava tieto voi olla vajaata, mikäli tutkija ei ole valmistautunut ja perehtynyt tutkittavaan aihepiiriin ja haastateltaviin riittävästi.

Lomakekysely ja haastattelu

Robsonin [69, s. 235] mukaan lomakekyselyt ovat hyvin paljon käytetty aineistonkeruumenetelmä kerättäessä tietoa ihmisen käyttäytymisestä ja asenteista. Lomakekyselyä voidaan käyttää yhdistetyissä aineistonkeruumenetelmissä joko ensisijaisena tai toissijaisena tiedonkeruumenetelmänä. Lomakekysely voidaan toteuttaa paperisena tai sähköisenä internetin välityksellä.

Eskolan ja Suorannan [30, ss. 85–86] mukaan strukturoidussa lomakekyselyssä kysymysten muotoilu ja järjestys kaikille vastaajille on sama. Myös vastausvaihtoehdot on annettu valmiiksi, vastaaja valitsee vaihtoehdoista itselleen sopivimman. Puolistrukturoidussa lomakekyselyssä kysymykset ovat myös kaikille vastaajille samat, mutta osa kysymyksistä voi olla avoimia kysymyksiä, jolloin vastaaja saa vastata kysymykseen omin sanoin. Robsonin [69, s. 245] mukaan lomakehaastattelun suunnittelussa tulisi huomioida lomakkeen täyttämiseen vaadittava aika, sekä kysymysten vaativuus. Monimutkaiset kysymykset on turvallisempaa esittää kasvok-

kain tapahtuvassa haastattelussa, jolloin tutkija voi varmistua että haastateltava on ymmärtänyt tutkittavan asian ja kysymyksen oikein.

Tuomen ja Sarajärven [83, ss. 75–77] mukaan puolistrukturoidusta lomakekyselystä voidaan käyttää nimitystä teemahaastattelu. Teemahaastattelu on luonnollinen nimitys, koska puolistrukturoidun kyselyn luonteeseen kuuluu kysymysten sijoittelu aihealueittain. Teemahaastattelussa pyritään saamaan selville ihmisten tulkintoja asioista ja kokemuksista, merkityksellisiä vastauksia tutkimustehtävän mukaisesti. Teemahaastattelussa voidaan hyödyntää puoliavoimia kysymyksiä ja avoimia kysymyksiä.

Metsämuuronen [58, ss. 245–247] suosittelee teemahaastattelun hyödyntämistä aineiston keruussa aina kun se on tutkimuksen kannalta mielekästä. Teemahaastattelu sopii hyödynnettäväksi tutkimuksissa, joissa halutaan selvittää asioita, joista voidaan arvella kirjallisesti saatavan luotettavampia vastauksia. Tällaisia tilanteita voivat olla intiimit tai arat aiheet, tai aiheet, joita tutkittavat eivät ole tiedostaneet hyvin.

Haastattelu

Eskola ja Suoranta [30, ss. 85–86] kuvailevat haastattelua yleisimmäksi tavaksi kerätä laadullista aineistoa. Robsonin [69, ss. 279–280] mukaan haastattelu voi olla täysin strukturoitu, puolistrukturoitu tai täysin vapaamuotoinen eli strukturoimaton haastattelu. Täysin strukturoitu haastattelu noudattaa tarkkaan määriteltujen kysymysten esittämistä määritetyssä järjestyksessä. Puolistrukturoitu haastattelu on vapaamuotoisempi, mutta siinä hyödynnetään tarkistuslistaa tutkijan tukena. Tarkistuslistan avulla tutkija voi varmistua kysyneensä kaikki tutkimuksen kannalta oleelliset asiat haastateltavalta. Vapaamuotoinen haastattelu ei sisällä etukäteen tehtyjä kysymyksiä tai tarkistuslistoja, vaan keskustelun annetaan kehittyä sille luontaiseen suuntaan.

Haastattelumuodoista puolistrukturoitu haastattelu on tärkein tapaustutkimuksessa, hyvin muotoiltuna se voi tuottaa laadukkaimman aineiston [31, s. 65]. Haastattelussa pyritään selvittämään mitä tutkittavalla on mielessään. Haastattelussa olennaisinta on kuitenkin että tutkija ohjaa ja johdattelee keskustelua. [30, ss. 85–86] Corbinin ja Straussin [18, s. 28] mukaan haastattelutilanteissa on aina viisasta pitää mukana muutamia keskustelua ohjaavia kysymyksiä, vaikka kyseessä olisi-kin täysin vapaamuotoinen haastattelu. Kokemattomat haastattelijat voivat tuntea haastattelutilanteissa epätietoisuutta etenemissuunnasta, mikäli haastateltavalla ei

ole paljon sanottavanaan. Hiljaisuus haastattelussa voi johtua myös jännittävästä tilanteesta, joka voidaan purkaa helppojen kysymysten tai yleisen keskustelun avulla.

Yinin [87, s. 92] mukaan haastattelu on erinomainen keino kerätä aineistoa, koska tapaustutkimukselle on luontaista sen ihmisläheisyys. Ihmisten toimintatapoihin ja asenteisiin liittyvä tutkimus voi kartuttaa arvokasta aineistoa, mikäli haastateltavat henkilöt on valittu huolella ja heille on kerrottu tutkimuksen tarkoituksesta. Haastateltavat henkilöt voivat tarjota sellaistaakin tietoa, jota tutkija ei ole osannut kaivata, esimerkiksi tapaukseen liittyvää historiatietoa. Haastatteluissa tulisi aina käyttää vain sanallista ilmaisua, joka tulee nauhoittaa. Haastattelu nauhoitetaan, jotta voidaan varmistua aineiston säilyvyydestä ja toisaalta välttyä muistiinpanojen varassa olevasta tiedosta.

Robsonin [69, ss. 293–295] mukaan haastattelu voidaan toteuttaa kahdenkeskisen keskustelun sijaan myös ryhmähaastatteluna. Ryhmähaastattelu voi olla muodoltaan täysin samankaltainen kuin aikaisemmin mainitut strukturoitu, puolistrukturoitu ja avoin haastattelu. Ryhmähaastattelun etuna on erityisesti ryhmädynamiikka, ryhmän jäsenet täydentävät toisiaan ja ryhmässä osallistuminen vähentää tilanteen jännittyneisyyttä. Haastateltavat pitävät ryhmätilannetta miellyttävämpänä ja ryhmähaastattelun järjestäminen on myös tutkijalle aikaa ja resursseja säästävää. Ryhmän koko tulisi pitää alle 10 henkilön suuruisena, jotta haastattelu aika ei veny liian pitkäksi ja jokainen osallistuja tulee huomioiduksi. Haastateltavien valinta tulee tehdä huolella, toisilleen tutut henkilöt voivat joissakin tapauksissa tuottaa erilaisen tuloksen kuin toisilleen tuntemattomat henkilöt. Ryhmän jäsenten valinnassa tulee huomioida myös mahdollinen esimies ja alainen -tilanne.

Robsonin [69, s. 231] mukaan haastattelu suoritetaan yleensä kasvokkain tutkijan ja haastateltavan kesken, mutta haastattelu voidaan suorittaa myös puhelimitse. Haastattelu voidaan suorittaa myös etäyhteyksien avulla esimerkiksi videoyhteydellä. Tuomen ja Sarajärven [83, s. 86] mukaan tutkimukseen osallistuvien määrä ei tarvitse kaikissa tapauksissa olla suuri, harkinnanvaraisessa aineistonkeruussa tutkimuksen tiedonantajiksi valitaan vain sellaisia henkilöitä, joilta oletetaan saatavan parhaiten tietoa tutkittavasta ilmiöstä. Henkilöiden valinnan kriteerinä voivat olla mitkä tahansa ominaisuudet, kokemukset tai taidot joiden katsotaan edesauttavan tiedon keräämistä tutkittavasta ilmiöstä.

5.3 Aineiston analyysimenetelmät

Eisenhardtin [26] ja Eskolan ja Suorannan [30, ss. 62–63] mukaan aineistoa on silloin riittävästi, kun aineiston keruu ei tuota enää uutta tietoa. Sanotaan, että aineisto saavuttaa kylläntymisen. Tutkimuskysymyksen pitäminen mielessä aineistoa läpikäydessä on oleellista. Kylläntymistä ei voi saavuttaa, ellei tiedä mitä aineistosta halutaan saada irti. Aineiston määrään tapaustutkimuksessa voidaan antaa myös tulokinnanvarainen ohje; yksikin tapaus voi riittää, vastauksia tarvitaan juuri sen verran kuin on tarpeellista.

Aineiston keräämisen jälkeen aineisto tulee analysoida ja esittää. Aineisto voi olla monessa eri muodossa, mutta yleensä se on esitettävissä joko numeroina tai sanoina. Laadullinen aineisto on yleensä enemmän kuvailevaa kuin lukuihin perustuvaa, mutta myös se analysoidaan samaan tapaan kuin määrällinen aineisto. Analysointi tarkoittaa aineiston jakamista pienempiin osiin, osien välisen yhteyden tutkimista ja rakentamista suuremmaksi kuvaksi. [69, ss. 408–412]

Ristiintaulukointi

Määrällisessä tutkimuksessa tutkitaan usein kahden muuttujan välistä riippuvuutta. Tämä tarkoittaa sitä, että yhden muuttujan arvo on jotenkin yhteydessä toisen muuttujan arvoon. Esimerkiksi opiskelijan ja pistemäärän välillä voi olla havaittava yhteys. Tällaisen yhteyden havaitsemiseen voidaan käyttää ristiintaulukointia. [69, ss. 430–431]

Ristiintaulukointi on määrällisen tutkimuksen menetelmä, jonka avulla voidaan havainnoida satunnaismuuttujien jakautumista, ja niiden välistä riippuvaisuuksia. Ristiintaulukoinnissa muuttujat jaetaan selittäviin muuttujiin ja selitettäviin muuttujiin, joiden välistä yhteyttä halutaan tutkia. Selittävät muuttujat esitetään sarakkeissa ja selitettävät muuttujat riveittäin. Jokaiselle riville ja sarakkeelle lasketaan kokonaismäärät rivien ja sarakkeiden loppuun. Sarakkeiden lopussa olevat kokonaismäärät osoittavan selittävien muuttujien välisen jakauman. [69, s. 431]

Selittävien muuttujien ja selitettävien muuttujien välisen riippuvuuden merkittävyyttä voidaan testata khiin neliö -testillä, josta käytetään jatkossa nimeä χ^2 -testi. χ^2 -testi perustuu oletetun jakauman ja havaitun jakauman välisiin eroihin. Oletettu jakauma on jakauma, jossa muuttujilla ei ole minkäänlaista riippuvuutta. χ^2 -testissä oletettua jakaumaa kutsutaan nollahypoteesiksi, johon mitattua jakaumaa verrataan. Nollahypoteesin mukaan muuttujilla ei ole mitään yhteyttä. χ^2 -testin tu-

los on todennäköisyys p , jolla havaitun jakauman ja nollihypoteesin ero selittyy sattumalla. Mikäli χ^2 -testin tulos on tarpeeksi pieni voidaan olettaa, että muuttujien välillä on yhteys. Vakiintuneen tavan mukaan yksi kahdeskymmenesosa, (ts. 0,05 tai 5%) on riittävän pieni tulos todistamaan, että mitatuilla arvoilla on yhteys, joka ei selity pelkästään satunnaisuudella ja on siten tilastollisesti merkitsevä ero. Pienempi χ^2 -testin tulos tarkoittaa siis isompaa riippuvuutta. [69, ss. 432–433]

Khiin neliö -testin tulkintaan liittyy vapausaste df , joka kuvaa tutkittavan taulukon kokoa, ja lasketaan vähentämällä sekä sarakkeiden että rivien lukumäärästä yksi, ja kertomalla erotukset keskenään. χ^2 kertymäfunktio on erilainen eri vapausasteilla. Vertaamalla χ^2 -arvoa vapausasteen mukaiseen kertymäfunktioon saadaan χ^2 -testin tulos, joka on siis todennäköisyys p . Mikäli pieneen taulukkoon, jonka vapausaste on yksi, tulee johonkin soluun oletetuksi frekvenssiksi alle 5 niin khiin neliö -testiä ei voi käyttää. [69, ss. 432–433]

Selittävien muuttujien ja selitettävien muuttujien välisen riippuvuuden merkitsevyyttä voidaan testata myös Fischerin tarkalla testillä. Metsämuurosen mukaan [58, ss. 361–363] Fischerin tarkka testi perustuu ideaan, että lasketaan todennäköisyys saada havaintoa vastaava nelikenttä. Havaintoa vastaava nelikenttä on nelikenttä, jossa sarakkeiden ja rivien yhteenlasketut frekvenssit ovat samat. Fisherin tarkkaa testiä voidaan käyttää myös, kun taulukon solun frekvenssit jäävät alle viiden.

Taulukko 5.1: Yleisesti hyväksytyt merkitsevyystasot [37, s. 185]

p -arvo	Merkitsevyys
$0,05 < p \leq 0,1$	Tilastollisesti suuntaa antava eli oireellinen
$0,01 < p \leq 0,05$	Tilastollisesti melkein merkitsevä
$0,001 < p \leq 0,01$	Tilastollisesti merkitsevä
$p \leq 0,001$	Tilastollisesti erittäin merkitsevä

Taulukossa 5.1 kuvataan yleisesti hyväksytyt merkitsevyystasot. χ^2 -testillä tai Fisherin tarkalla testillä saadaan todennäköisyys p , joka kertoo, millä todennäköisyydellä havaitun jakauman ja nollihypoteesin välinen ero on sattumaa. Tätä todennäköisyyttä verrataan yleisesti hyväksytyihin merkitsevyystasoihin, jolloin saadaan selville, miten merkitsevä tulos on tilastollisesti. Mikäli todennäköisyys on pienempi kuin valittu merkitsevyystaso, niin nollihypoteesi hylätään, ja tutkitun

jakauman muuttujilla todetaan olevan tilastollisesti merkitsevä yhteys. [37]

Tässä tutkimuksessa p -arvoja laskettiin χ^2 -testin osalta käyttäen Microsoft Excel 2016 ohjelmaa. Ristiintaulukoiduille jakaumille laskettiin vertailujakauma, joka on jakauma, jossa selitettävän tekijän frekvenssit ovat jakautuneet tasaisesti selittävän tekijän suhteen. Tätä vertailujakaumaa kutsutaan nollahypoteesiksi. Saadun jakauman ja vertailujakauman vastaavien alkioiden erotuksen neliöiden summa jaetaan vertailujakauman alkiolla. Tätä kutsutaan χ^2 -arvoksi, jota vastaava p -arvo saadaan vapausasteen mukaan. Microsoft Excelistä löytyy myös funktio, jolla p -arvon voi laskea suoraan käyttäen saatua jakaumaa ja vertailujakaumaa. Käytimme p -arvojen laskemiseen Fischerin tarkalla testillä R-kieltä. R-kieli on tilastotiedettä varten kehitetty ohjelmointikieli, jolla voidaan helposti käsitellä aineistoja ja laskea tunnuslukuja. R-kielestä löytyy funktio Fischerin tarkan testin laskemiseksi ristiintaulukoidulle jakaumalle.

Teemoittelu

Aineiston analysointi on laadullisen tutkimuksen ongelmallisin vaihe, sillä siihen ei ole olemassa suoraviivaisia ohjeita [30, s. 145], [26], [64, s. 432]. Tuomi ja Sarajärvi [83, ss. 69–70] tarjoavat käytännöllisen ohjeen, jonka mukaan analyysimenetelmä tulisi valita jo ennen tutkimusaineiston keruun aloittamista. Analyysitavan valinta auttaa tiedonkeruutavan, esimerkiksi haastattelun suunnittelussa. Eskola ja Suoranta [30, s. 145] antavat kaksi vaihtoehtoa laadulliseen analyysiin. Analyysin ja tulkintojen tekemisessä on mahdollista kiinnittää huomio ainoastaan aineistoon, tai vaihtoehtoisesti käyttää aineistoa tutkijan teoreettisen ajattelun lähtökohtana tulkinnoille.

Robsonin [69, s. 478] mukaan ensimmäisenä tehtävänä nauhoitettujen aineistojen osalta on puheen muuntaminen tekstiksi eli litterointi. Jokaisen sanan kirjoittaminen tekstiksi on hyvin paljon aikaa kuluttava prosessi, jonka vuoksi kaikkea nauhoitettua aineistoa ei välttämättä ole mielekästä kirjoittaa puhtaaksi. Teemoittelu analyysimenetelmänä ei vaadi yhtä tarkkaa litterointia kuin esimerkiksi diskursianalyysi.

Teorian luomisessa kriittinen osa on tarkastella hypoteesia suhteessa olemassa olevaan kirjallisuuteen. Olemassa olevan kirjallisuuden tutkiminen ja kehitettävän teorian vertaaminen siihen edellyttävät pohdintaa *"Onko tämä teoria samankaltainen olemassa olevan kirjallisuuden kanssa, onko ristiriitoja, jos on niin miksi"* [26]. Yhtäläisyyksien ja myös eroavaisuuksien löytyminen vastaavasta kirjallisuudesta on teo-

rian muodostuksessa oleellista. Uusi teoria voi vahvistaa olemassa olevaa kirjallisuutta, jos se tulkitaan oikein. [26]

Gillhamin [31, ss. 85–86] ja Tuomen ja Sarajärven [83, s. 93] mukaan laadullinen aineisto muotoutuu tapaustutkimuksessa usein jo valmiiksi teemoittain, tai on mahdollista luokitella analysointivaiheessa. Aineisto voidaan järjestellä teemoittain ja sijoitella sen jälkeen esimerkiksi taulukkoon, jolloin eri vastaajien tai vastaajaryhmien eroavaisuuksia voidaan vertailla. Jäsennellyssä haastattelussa teemat seuraavat usein haastattelun rakennetta, mutta aineiston läpikäynnin yhteydessä voi nousta esiin myös uusia teemoja.

Robson [69, ss. 474–476] esittää teemoittelun määrittävän, mitä analysoitava aineisto sisältää. Jakaessaan aineistoa teemoihin tutkijan tulisi miettiä, mikä aineiston osa liittyy johonkin toiseen osaan. Teemojen rajaaminen ei ole teemoittelussa määritetty tarkasti. Usein tutkija aloittaa suhteellisen pienellä määrällä teemoja, jotka voivat muotoutua pääteemoiksi ja alateemoiksi analysoinnin edetessä. Mahdollisia teemoja kannattaa miettiä jo aineistonkeruun aikana ja toistuvia rakenteita kannattaa hyödyntää analysointivaiheessa.

Robsonin [69, ss. 476–479] mukaan analysoinnin ensimmäisessä vaiheessa kerättyyn aineistoon tulee tutustua tarkasti. Aineistoon tutustuminen voi olla luettavan aineiston tarkastelua tai äänitetyn aineiston litterointia. Kaikkea nauhoitettua aineistoa ei ole aina tarpeen kirjoittaa puhtaaksi, mutta se on erinomainen tapa tutustua aineistoon, erityisesti siinä tapauksessa, jos aineiston on kerännyt toinen henkilö. Ennen kuin litteroitua aineistoa voidaan jakaa teemoihin, siitä tulee tunnistaa ydinsisältö ja siihen liittyvät osat. Alkuvaiheen jaottelua kutsutaan alustavaksi koodaamiseksi, joka voidaan toteuttaa hyvinkin perinteisiä menetelmiä käyttäen. Koodausta varten litteroitu materiaali voidaan esimerkiksi tulostaa ja sen jälkeen paloitella materiaalia korostuskynien, saksien ja muistilappujen avulla. Koodaaminen tulee tehdä aina tutkimuskysymys mielessä pitäen.

Kun alustava koodaaminen on suoritettu loppuun, voidaan teemojen tunnistaminen ja ryhmittely aloittaa. Koodattuja aineiston osia sijoitellaan havaittujen teemojen alle. Tämä vaihe on usein tutkijalle melko suoraviivainen, koska teemat saattavat syntyä jo aineistonkeruuvaiheessa ja viimeistään koodauksen yhteydessä. Kun koodatut kokonaisuudet on jaoteltu teemojen ja aliteemojen alle, pitäisi tutkijalla olla ymmärrys, kuinka teemat sopivat yhteen toistensa kanssa ja mitä aineisto kertoo. Teemoiteltua aineistoa voidaan alkaa analysoida eri keinoin. Teemat voidaan sijoitella taulukkoon, listoiksi tai miellekartaksi, jossa teemojen yhteydet näkyvät.

Teemoittelun lopputuloksena pitäisi olla aineiston kerronta sellaisessa muodossa ja yhteyksissä, jotka vakuuttavat lukijan aineiston luotettavuudesta. [69, ss. 479–486]

5.4 Tutkimuksen luotettavuus ja toistettavuus

Metsämuurosen [58, s. 125] mukaan tutkimuksen luotettavuuden arviointiin liittyvät termit reliabiliteetti ja validiteetti. Reliabiliteetti tarkoittaa tutkimuksen toistettavuutta. Validiteetilla varmistetaan, että tutkitaan sitä, mitä on tarkoitus tutkia. Tutkimuksen luotettavuudesta puhuttaessa tarkoitetaan sitä, kuinka samankaltaisia tai poikkeavia tuloksia saataisiin tutkimuksen toistamisella.

Heikkilä [37, s. 28] kuvailee reliabiliteetin tarkoittavan tutkimuksessa samaa asiaa, kuin luotettavuus missä tahansa arkelämän asiassa. Tutkimuksen tulokset eivät voi olla sattumanvaraisia, jotta tutkimus voi pysyä luotettavana. Heikkilän [37, s. 28] mukaan tuloksiin vaikuttavia virheitä voi sattua jo aineistonkeruun aikana. Tulosten vaihteluun vaikuttaa tutkimuksen kohderyhmän monimuotoisuus ja suppea otanta. Reliabiliteetin suhteen virhe voi syntyä myös otoksen valinnassa. Otos pitäisi olla mahdollisimman yleinen joukko tutkittavasta kohderyhmästä. Hyvin valitun otoksenkin voi kohdata kato, jolloin vastauksia ei saavuteta tavoiteltua määrää.

Heikkilän [37, s. 27] mukaan tutkimuksen validiteetti tarkoittaa tutkimuksen pätevyyttä. Pätevyydellä varmistetaan, että tutkitaan sitä, mitä pitääkin tutkia. Validiteetti konkretisoituu esimerkiksi lomaketutkimuksessa. Lomakkeen kysymysten tulisi mitata oikeita asioita. Oikeiden asioiden määrittelyä ohjaa tutkimuskysymys, jonka selvittämiseksi tutkimusta tehdään. Pätevän tutkimuksen toteuttamista edistävät hyvin määriteltä perusjoukko ja otanta. Korkea vastausprosentti tukee validiteettia.

5.5 Tutkimuksen toteutus

Tässä tutkimuksessa hyödynnetään sekä määrällistä että laadullista menetelmää aineistonhankinnassa. Heikkilän [37, s. 15] mukaan määrällisessä tutkimuksessa pyritään yleensä kartoittamaan olemassa oleva tilanne, mutta sillä ei pystytä yleensä selvittämään syitä. Laadullinen tutkimus auttaa ymmärtämään tutkimuskohdetta, ja syitä tilanteen taustalla. Taulukko 5.2 kuvaa määrällisen ja laadullisen tutkimuksen oleelliset erot. Tutkimuksessa hyödynnetään määrällistä menetelmää yleiskuvan saavuttamiseksi koko Lapin AMKin henkilökunnan suhtautumisesta vaati-

musmäärittelytyöhön. Laadullista menetelmää hyödynnetään ymmärryksen syventämiseen tutkittavasta ilmiöstä, eli laadullisella haastattelulla selvitetään syitä asenteille.

Taulukko 5.2: Määrällisen ja laadullisen tutkimuksen oleelliset erot [37, s. 15]

Määrällinen	Laadullinen
Vastaa kysymyksiin: Mikä?, Missä?, Paljonko?, Kuinka usein?	Vastaa kysymyksiin: Miksi?, Miten?, Millainen?
Numeerisesti suuri, edustava otos	Suppea, harkinnanvaraisesti koottu näyte
Ilmiön kuvaus numeerisen tiedon pohjalta	Ilmiön ymmärtäminen ns. pehmeän tiedon pohjalta

Tutkimus on Lapin ammattikorkeakouluun kohdistuva tapaustutkimus. Tutkimuksen ensimmäisessä vaiheessa suoritettiin sähköinen lomakekysely, jonka jälkeen järjestettiin ryhmähaastatteluja pienelle otannalle. Tutkimuksen perusjoukko on Lapin ammattikorkeakoulun päätoiminen henkilökunta, yhteensä 450 henkilöä. Sähköiseen lomakekyselyyn kutsuttiin osallistumaan koko päätoiminen henkilökunta sähköpostitse. Vastaajat pystyivät ilmaisemaan kiinnostuksensa osallistua tutkimuksen toiseen vaiheeseen eli ryhmähaastatteluun täyttämällä oman työ sähköpostiosoitteensa lomakkeella siihen varattuun kenttään. Tutkimuksen toisessa osassa, ryhmähaastatteluissa käsiteltiin syvällisemmin lomakehaastattelun avulla kerättyjä huomioita käyttäjien suhtautumisesta ohjelmistojen vaatimusmäärittelytyöhön. Haastattelut järjestettiin kahdelle ryhmälle: henkilöille, jotka ovat joskus osallistuneet vaatimusmäärittelytyöhön, sekä henkilöille, joilla ei ole aikaisempaa kokemusta vaatimusmäärittelytyöstä. Molempien ryhmien jäsenet valittiin lomakehaastattelussa kerättyjen ilmoittautumisten perusteella.

Aineistonkeruu suoritettiin kahdessa osassa, koska riittävän ymmärryksen saavuttaminen tutkimusongelmista olisi voinut olla riittämätöntä pelkän lomakekyselyn avulla. Toisaalta lomakekyselyssä ei ole mielekästä kysyä kovin syvällisiä aihealueita, ottaen huomioon vaatimusmäärittelytyön aiheena ja tutkimuksen monialaisen kohderyhmän. Määrällisen lomakekyselyn tarkoituksena oli siten tuottaa tutkittavasta ilmiöstä yleiskuva, jota hyödynnettiin pohja-aineistona haastatteluille. Laadullisten ryhmähaastattelujen sisältö perustui lomakekyselyssä kerättyyn ai-

neistoon. Haastatteluissa pyrittiin keskustelun avulla saavuttamaan tarkempaa tietoa suhtautumisesta vaatimusmäärittelytyöhön. Haastattelemalla ihmisiä kasvokkain saavutettiin paljon monimuotoisempaa ja yksityiskohtaisempaa aineistoa tutkimusongelmista.

Tutkimuksen molemmat osat toteutettiin maaliskuussa 2017. Ensin toteutettiin lomakekysely Lapin ammattikorkeakoulun henkilökunnalle Webropol-kyselytyökalun avulla. Lomakekyselyn avulla saavutettiin yleiskuva henkilökunnan suhtautumisesta ohjelmistojen vaatimusmäärittelytyötä kohtaan. Lomakekyselyn tulokset toimivat pohja-aineistona haastattelututkimuksen suunnittelussa. Haastattelu toteutettiin ryhmähaastatteluina kahdelle ryhmälle. Ensimmäiseen ryhmään kutsuttiin henkilöitä, jotka ovat joskus osallistuneet vaatimusmäärittelytyöhön ja toiseen ryhmään kutsuttiin henkilöitä, joilla ei ollut aikaisempaa kokemusta vaatimusmäärittelytyöstä.

Tutkimuksen toistettavuudessa tulee huomioida, että toinen tutkijoista työskentelee tutkittavassa organisaatiossa. Siten osa haastateltavista henkilöistä on tutkijalle tuttuja henkilöitä, joka voi vaikuttaa haastattelutilanteeseen. Tuttu organisaatio voi vaikuttaa myös otannan valintaan.

5.5.1 Aineistonhankintamenetelmät

Aineistonkeruu toteutettiin kahdessa osassa, ensin sähköisellä lomakekyselyllä ja sen jälkeen ryhmähaastatteluilla. Haastattelun puolistrukturoitu rakenne suunniteltiin lomakekyselyn avulla kerätystä aineistosta. Kyselyyn vastanneet pystyivät ilmaisemaan kiinnostuksensa tutkimuksen toiseen osaan eli haastatteluun osallistumiseen. Ryhmähaastatteluihin osallistuneet henkilöt valittiin vapaaehtoisten joukosta, huomioiden vastaajien kokemus vaatimusmäärittelytyöstä.

Lomakekysely

Lomakekyselyllä haluttiin saavuttaa yleiskuva Lapin ammattikorkeakoulun henkilökunnan suhtautumisesta vaatimusmäärittelytyöhön. Yleiskuvan tuottaminen nähtiin välttämättömänä, jotta tutkittavaan ongelmaan voidaan pureutua syvällisemmin. Lomakekyselystä kehitettiin yksinkertainen, lyhyt ja vastaajaa motivoiva. Kyselystä jätettiin tietoisesti yksilöivät tiedot pois, kuten titteli tai työyksikkö. Tutkimusongelmien kannalta tärkeiksi taustatiedoiksi nähtiin vastaajan ikä, korkein suoritettu tutkinto ja tietokoneen käyttötaidot. Lomake sisältää 15 kysymystä, joista yk-

si on avoin kysymys ja loput kysymykset sisältävät valmiiksi annetut vastausvaihtoehdot. Lomakekyselyn saatekirje ja varsinainen kysely löytyvät liitteestä A. Neljä ensimmäistä kysymystä liittyvät vastaajan taustatietoihin:

1. Ikä
2. Mikä on korkein suoritettu tutkintosi?
3. Koulutusalasasi?
4. Millaiseksi arvioit omat tietokoneen käyttötaitosi?

Ikäryhmät määritettiin alle 30-vuotiaat ja sen jälkeen ikäryhmät 10 vuoden välein. Korkein suoritettu tutkinto ja koulutusala ovat Tilastokeskuksen määrittämiä yleisiä kategorioita. Kysymykseen 4. vastaaja voi vastata oman arvionsa mukaan omat tietokoneen käyttötaidot vaihtoehdoista "erinomainen", "hyvä", "keskinkertainen" tai "välttävä". Taustatietoja seuraavat kysymykset selvittävät vastaajan kokemuksiä vaatimusmäärittelytyöstä. Kysymyksessä 5. selvitetään tietääkö vastaaja mitä vaatimusmäärittelytyö tarkoittaa. Kysymykseen voi vastata "kyllä", "ei" tai "en osaa sanoa". Jos vastaaja valitsee kohdan kyllä, pyydetään häntä kirjoittamaan lyhyesti oma käsitys/määritelmä vaatimusmäärittelytyöstä.

5. Tiedätkö mitä vaatimusmäärittely tarkoittaa?
8. Koetko että sinulla olisi annettavaa vaatimusmäärittelyissä?
11. Missä rooleissa olet osallistunut vaatimusmäärittelyihin?

Kysymyksessä 8. selvitetään kokeeko vastaaja, että hänellä olisi annettavaa vaatimusmäärittelytyössä. Kysymykseen voi vastata "erittäin paljon", "melko paljon", "jonkin verran", "vähän" tai "ei lainkaan". Kysymyksessä 11. selvitetään millaisessa roolissa vastaaja on osallistunut vaatimusmäärittelytyöhön. Kysymykseen on mahdollista vastata, ettei ole koskaan osallistunut vaatimusmäärittelytyöhön. Jos vastaaja on osallistunut joskus vaatimusmäärittelytyöhön, vastausvaihtoehdot ovat tulevan ohjelmiston loppukäyttäjä tai jokin muu rooli. Jos käyttäjä vastaa jokin muu rooli, häntä pyydetään kirjoittamaan missä roolissa vaatimusmäärittelytyöhön on osallistunut. Kysymyksiin 6., 7., 9., 10., 12. ja 13. on mahdollista vastata ainoastaan "kyllä", "ei" tai "en osaa sanoa". Kysymysten avulla pyritään selvittämään erityisesti asenteita vaatimusmäärittelytyötä kohtaan. Kysymysten avulla saadaan tietoa

asenteista niissäkin tapauksissa, kun vastaaja ei ole koskaan osallistunut vaatimusmäärittelytyöhön. Kysymyksiä ovat:

6. Oletko osallistunut aikaisemmin vaatimusmäärittelyyn?
7. Tunnetko olevasi pätevä osallistumaan vaatimusmäärittelyyn?
9. Koetko että sinulla on työnantajan tuki vaatimusmäärittelyyn osallistumiseen?
10. Koetko vaatimusmäärittelyyn osallistumisen olevan vaivalloista?
12. Koetko että vaatimusmäärittelyyn osallistuminen lisää mahdollisuuksiasi vaikuttaa kehitettävän ohjelmiston ominaisuuksiin?
13. Oletko kiinnostunut osallistumaan jatkossa vaatimusmäärittelyihin?

Kysymys 14. liittyy edellisen kysymyksen 13. vastaukseen. Jos vastaaja on valinnut vastausvaihtoehdon "ei" kysyttäessä kiinnostuksesta osallistua jatkossa vaatimusmäärittelytyöhön, pyydetään kysymyksessä 14. kertomaan mitä pitäisi muuttaa, jotta vaatimusmäärittelytyöhön osallistuminen olisi mielekkäämpää ja siihen että siihen osallistuminen tuntuisi tärkeältä. Viimeisessä kysymyksessä 15. kerrotaan tutkimuksen jakautuvan kahteen osaan, lomakekyselyyn ja haastattelututkimukseen. Tutkimuksen onnistumisen kannalta molemmat menetelmät ovat tärkeitä, jonka vuoksi kysymykseen liittyy pyyntö ilmoittautua vapaaehtoiseksi haastattelututkimukseen. Vastaaja voi halutessaan ilmoittaa oman työsähköpostiosoitteensa vastauskenttään.

14. Jos vastasit kysymykseen 13. kieltävästi, niin mitä pitäisi muuttaa, jotta vaatimusmäärittelyyn osallistuminen olisi mielekkäämpää ja kokisit osallistumisen tärkeäksi?
15. Tutkimuksemme jakautuu kahteen osaan: kyselytutkimukseen ja haastattelututkimukseen. Molemmat osat ja vastauksesi ovat tärkeitä pro gradu -tutkielmamme onnistumisen kannalta. Toivoisimme, että voisit osallistua myös haastatteluun. Jos haluat ilmoittautua vapaaehtoiseksi haastattelututkimukseen, voit ilmoittaa työsähköpostiosoitteesi tässä.

Robsonin [69, s. 264] mukaan lomakekysely on syytä koekäyttää ennen sen lähettämistä suuremmalle joukolla. Testivastajina voivat toimia esimerkiksi kollegat,

ystävät tai perheenjäsenet. Läheiset ihmiset haluavat yleensä vilpittömästi auttaa lomakkeen suunnittelussa, ja tutustuvat kyselyyn ajatuksella. Läheisten ihmisten antama palaute on usein myös rakentavaa, he voivat kertoa ovatko kysymykset selkeitä, yksinkertaisia ja yksiselitteisiä. Tämän tutkimuksen Webropol-järjestelmän avulla suunniteltu lomake testattiin monialaisesti eri koulutusasteen omaavilla henkilöillä, jotka eivät kuulu tutkimuksen perusjoukkoon. Lomake koettiin toimivaksi ja ymmärrettäväksi, jonka jälkeen lomakkeen URL-osoite lähetettiin saatekirjeen yhteydessä sähköpostin välityksellä Lapin ammattikorkeakoulun päätoimiselle henkilöstölle.

Lapin ammattikorkeakoulun päätoimiseen henkilöstöön kuului tutkimuksen aikana yhteensä 450 henkilöä. Lapin AMKin henkilöstömäärä on kokonaisuudessaan 461 henkilöä. Lapin AMK toimii kolmella paikkakunnalla, Kemissä, Rovaniemellä ja Torniossa. Henkilöstö on jakautunut siten, että Rovaniemellä työskentelee 256 henkilöä, Kemissä 145 henkilöä ja Torniossa 60 henkilöä. Lapin AMKin ydinprosesseja ovat opetus ja tutkimus-, kehitys- ja innovaatiotoiminta (TKI). Henkilöstöä sijoittuu opetuksen ja TKI:n lisäksi myös hallintoon ja opetuksen tukipalveluihin.

- Opetus 227
- TKI 77
- Hallinto ja opetuksen tukipalvelut 157
- Yht. 461

Lomakekysely suunniteltiin Webropol-järjestelmään, jonka välityksellä vastauslinkit lähetettiin Lapin ammattikorkeakoulun päätoimiselle henkilökunnalle organisaation sähköpostiosoitteisiin. Webropolin avulla voidaan toteuttaa kyselytutkimuksia ja analyysjä verkon välityksellä. Lomake oli mahdollista täyttää myös ilman aikaisempaa kokemusta vaatimusmäärittelytyöhön osallistumisesta. Vastaajia motivoitiin osallistumaan tutkimukseen sekä sähköpostin saatekirjeessä, että varsinaisella lomakkeella. Vastaajille korostettiin osallistumisen tärkeyttä, vaikka kokemuksiä vaatimusmäärittelystä ei olisikaan. Kysely oli avoinna neljän arkipäivän ajan, jonka jälkeen vastauslinkki suljettiin. Ensimmäisen vastauspäivän aikana saavutettiin valtaosa vastauksista, 97 vastausta. Kaikkiaan kyselyyn vastasi vastaajan puitteissa 111 henkilökunnan jäsentä, jolloin vastausprosentiksi saatiin 25%.

Ryhmähaastattelut

Tutkimuksen toinen osa koostui ryhmähaastatteluista, joiden avulla pyrittiin saavuttamaan yksityiskohtaisempaa tietoa vaatimusmäärittelytyöhön suhtautumisesta. Haastattelut järjestettiin kahdelle ryhmälle, henkilöille, joilla oli aikaisempaa kokemusta vaatimusmäärittelytyöstä, ja henkilöille, joilla ei ollut aikaisempaa kokemusta vaatimusmäärittelytyöstä. Haastattelututkimukseen vapaaehtoisia ilmoitettiin lomakekyselyn kautta yhteensä 34. Vapaaehtoisista 9 henkilöä eivät olleet aikaisemmin osallistuneet vaatimusmäärittelytyöhön. Muilla 25 ilmoittautuneilla oli kokemusta joko loppukäyttäjän roolissa osallistumisesta tai jossakin toisessa roolissa osallistumisesta.

Valitsimme molempiin haastatteluryhmiin 3 osallistujaa. Haastateltavien määrä valittiin sen perusteella, että yhden tai kahden henkilön näkemys olisi voinut olla liian kapea. Toisaalta, koska haastattelu tahdottiin pitää napakkana noin puolen tunnin mittaisena, olisi useampi kuin kolme haastateltavaa voinut pidentää haastattelun kestoa. Ryhmähaastattelut haluttiin järjestää kasvokkain samassa tilassa, jonka vuoksi ilmoittautuneiden työsuorituspaikka vaikutti valintaan. Haastatteluun valittiin taustoiltaan mahdollisimman erikaltaisia osallistujia aineiston monipuolisuuden tavoittelemiseksi. Koska toinen haastattelijoista työskentelee samassa organisaatiossa, pyrittiin haastateltavat valitsemaan huomioiden, että haastateltavat eivät ole liian tuttuja henkilöitä haastattelijalle.

Haastatteluun vapaaehtoisiksi ilmoittautuneita vaatimusmäärittelytyöstä kokemusta omaavia henkilöitä oli yhteensä 25. Ilmoittautuneiden jakauma työsuorituspaikan mukaan oli: Kemi 5, Rovaniemi 14 ja Tornio 6. Kemistä ilmoittautuneiden koulutusalat painottuivat tekniikan ja liikenteen alalle, sekä yhteiskuntatieteiden, liiketalouden ja hallinnon alalle. Ilmoittautuneiden iät Kemissä olivat 40 – 60 -ikävuosien välillä. Tornioista ilmoittautuneiden koulutusaloissa oli Kemiä enemmän hajontaa, mutta ilmoittautuneiden ikäjakauma oli pääasiassa 40 – 59, yhden ilmoittautuneen ollessa 30 – 39. Suurin hajonta kaikissa taustatiedoissa saavutettiin Rovaniemellä työskentelevien osalta. Haastateltavat päätettiin siten valita Rovaniemeltä monipuolisuuden vuoksi.

Haastatteluryhmä 1. sisälsi kolme vaatimusmäärittelytyöhön aikaisemmin osallistunutta Lapin AMKin henkilökunnan jäsentä. Ryhmän jäsenten korkein suoritettu koulutus oli joko toisen asteen tutkinto, alempi korkeakoulututkinto tai ylempi korkeakoulututkinto. Lisäksi jokaisella henkilöllä oli poikkeava koulutusala. Henkilöt sijoittuivat kolmeen ikäryhmään välillä 40 – 49, 50 – 59 ja 60–. Kahdella haasta-

teltavalla kolmesta oli kokemusta vaatimusmäärittelytyöhön osallistumisesta myös muussa roolissa kuin ohjelmiston loppukäyttäjänä. Jokaisen haastateltavan työnsuorituspaikka oli Rovaniemi. Ryhmähaastatteluun valittujen henkilöiden taustatiedot on esitetty liitteessä B Ryhmähaastattelut.

Haastatteluun vapaaehtoisiksi ilmoittautuneita henkilöitä ilman kokemusta vaatimusmäärittelytyöstä oli yhdeksän. Henkilöistä 3 työskentelee Kemissä, ja he kaikki olivat tekniikan ja liikenteen alalta. Torniossa ilmoittautui 2 henkilöä. Loput 4 henkilöä olivat Rovaniemeltä ja muodostivat monipuolisimman joukon taustatietojen perusteella. Haastateltavat päätettiin siten valita Rovaniemeltä, koska Kemissä työskentelevien koulutusalat olivat jokaisella samat, ja Torniossa ilmoittautuneita ei ollut riittävästi.

Haastatteluryhmä 2. sisälsi kolme Lapin AMKin henkilökunnan jäsentä, joilla ei ole aikaisempaa kokemusta vaatimusmäärittelytyöstä. Ryhmän jäsenistä kahdella korkein suoritettu koulutus oli ylempi korkeakoulututkinto ja yhdellä alempi korkeakoulututkinto. Lisäksi jokaisella henkilöllä oli poikkeava koulutusala. Henkilöt sijoittuivat kolmeen ikäryhmään välillä 30 – 39, 40 – 49 ja 60–. Kaksi haastateltavaa kolmesta ei ollut varma onko hänellä työnantajan tuki vaatimusmäärittelytyöhön osallistumiseen ja yksi henkilö koki, ettei hänellä ollut työnantajan tukea vaatimusmäärittelytyöhön osallistumiseen. Jokaisen haastateltavan työnsuorituspaikka oli Rovaniemi. Ryhmähaastatteluun valittujen henkilöiden taustatiedot on esitetty liitteessä B Ryhmähaastattelut.

Ryhmähaastattelut järjestettiin maaliskuussa 2017 Rovaniemellä. Molemmat tutkielman kirjoittajat olivat haastattelutilanteessa läsnä haastattelijan roolissa. Molemmilla haastattelijoilla oli samanlainen rooli keskustelun ohjaamisessa. Haastattelutilanteet äänitettiin kokonaisuudessaan, jonka ansioista muistiinpanoja ei ollut tarpeen tehdä. Haastattelut noudattivat suunniteltuja aihealueita ja niihin liittyviä kysymyksiä. Keskustelun annettiin molemmissa ryhmähaastatteluissa edetä luonnollisesti, joskin haastattelijat ohjasivat keskustelua kysymysten muodossa huomioiden käytettävissä oleva aika. Haastattelujen keston suunniteltiin olevan noin puoli tuntia. Käytännössä haastatteluryhmän 1. kanssa aikaa kului 56 minuuttia ja haastatteluryhmän 2. kanssa aikaa kului 35 minuuttia.

Aihealueet haastatteluryhmälle 1.

- Aikaisemmat kokemukset vaatimusmäärittelytyöstä

- Motivaatio ohjelmistohankintaan osallistumiseen
- Kuvailu hyvästä vaatimusmäärittelytyöstä

Haastatteluryhmään 1. kutsuttiin 3 osallistujaa, joista yhdelle tuli äkillinen este, eikä hän pystynyt osallistumaan haastatteluun. Haastattelu järjestettiin sovitun mukaisesti kahden muun haastateltavan henkilön kanssa. Keskustelussa käytettiin tukena edellä mainittuja aihealueita, joiden avulla selvitettiin henkilöiden suhtautumista ohjelmistojen vaatimusmäärittelytyöhön. Molemmat haastateltavat olivat osallistuneet aikaisemmin vaatimusmäärittelytyöihin, lähinnä kehittäjän näkökulmasta.

Aihealueet haastatteluryhmälle 2.

- Ajatuksia vaatimusmäärittelytyöstä ja sen sisällöstä
- Ennakkoasenteet vaatimusmäärittelytyötä kohtaan
- Kokemukset mahdollisuudesta osallistua vaatimusmäärittelytyöhön

Haastatteluryhmään 2. kutsuttiin 3 osallistujaa, joista jokainen osallistui haastatteluun. Haastateltavilla ei ollut aikaisempaa kokemusta vaatimusmäärittelytyöhön osallistumisesta, mutta heillä oli selkeä käsitys vaatimusmäärittelytyöstä osana ohjelmistoprojektia. Keskustelussa käytettiin tukena edellä mainittuja aihealueita, joiden avulla selvitettiin henkilöiden suhtautumista ohjelmistojen vaatimusmäärittelytyöhön.

5.5.2 Aineiston analysointi

Ristiintaulukointi

Lomakekyselyn analysointiin käytettiin ristiintaulukointia. Eri muuttujien välisiä yhteyksiä tutkittiin χ^2 -testillä. Jos χ^2 -testiin ei saatu riittävän suuria frekvenssejä, niin yhteyksien tutkimiseen käytettiin Fischerin tarkkaa testiä. Tässä tutkimuksessa merkitsevyysrajaksi on valittu 5%. Osassa taulukoista tuloksia on ristiintaulukointia varten yhdistetty, jotta solujen frekvenssit saatiin riittäviksi χ^2 -testiä varten. Lomakekyselyn ristiintaulukoinnit on kerätty liitteeseen C.

Ristiintaulukoinnissa selvitettiin vastaajien suhtautumista vaatimusmäärittelytyöhön mittaamalla selittävien tekijöiden ja selitettävien tekijöiden välisiä riippuvuuksia.

Selittävinä tekijöinä käytettiin henkilön

- ikää
- omaa arviota tietokoneen käyttötaidosta
- koulutusastetta
- koulutusala
- omaa arviota siitä tietääkö hän mikä on vaatimusmäärittelytyö
- aiempaa osallistumista vaatimusmäärittelytyöhön.

Selitettävänä tekijöinä käytettiin sitä

- kokeeko henkilö olevansa pätevä vaatimusmäärittelytyöhön
- kokeeko henkilö, että hänellä on annettavaa vaatimusmäärittelytyöhön
- kokeeko henkilö voivansa vaikuttaa vaatimusmäärittelytyön avulla
- kokeeko henkilö vaatimusmäärittelytyön vaivalloiseksi
- kokeeko henkilö, että hänellä on oman organisaation tuki vaatimusmäärittelytyöhön
- haluaako henkilö osallistua jatkossa vaatimusmäärittelytyöhön.

Teemoittelu

Molemmat ryhmähaastattelut nauhoitettiin, jonka jälkeen äänitteet litteroitiin niiden keskeisimpien asiasisältöjen osalta. Litterointi suoritettiin kirjoittamalla keskustelu puhtaaksi, siten kuin haastattelussa keskusteltiin. Litteroinnissa korostettiin haastattelussa esitetyt kysymykset ja niiden vastaukset vastaajakoodittain. Haastattelut merkittiin tunnuksin H1 ja H2, jonka lisäksi haastateltavat merkittiin tunnuksin H1, H2 ja H3. Molempien haastattelujen aineistot yhdistettiin litteroinnin jälkeen analysointia varten. Litteroitu teksti analysoitiin teemoittelua hyödyntäen,

jonka ensimmäisenä vaiheena suoritettiin koodaaminen. Tekstistä pyrittiin hahmotamaan asiakokonaisuuksia taulukoinnin avulla.

Molempien haastattelujen litteroitu teksti purettiin ensin lauseiksi ja lauseiden muodostamiksi kokonaisuuksiksi, jotka liitettiin taulukkolaskentaohjelmassa samaan sarakkeeseen omiksi riveiksi. Riveille annettiin koodit H1 tai H2 sen mukaisesti, kummasta ryhmähaastattelusta aineisto oli kerätty. Seuraavaksi jokainen lause käsiteltiin raakatana, josta kuvattiin vain sen ydinsisältö seuraavaan sarakkeeseen. Ydinsisällön kuvailun jälkeen muodostettiin kolmas sarake, joka sisälsi yleisellä tasolla ydinsisällön viestin. Kolmannen tason sarakkeet järjestettiin aakkosjärjestykseen, jonka myötä hahmotettiin samankaltaiset asiakokonaisuudet. Sopivat asiakokonaisuudet yhdistettiin edelleen neljänneksi sarakkeeksi, joka muodosti teeman. Esimerkki teemoittelusta on esitetty liitteen B ryhmähaastattelut taulukossa B3.

Teemoja muodostui järjestelyn myötä yhteensä kolme. Koodauksen jälkeen tarkastettiin, kuinka aineisto vastasi alkuperäisiä aihealueita.

Alkuperäiset aihealueet haastatteluryhmälle 1.

- Aikaisemmat kokemukset vaatimusmäärittelytyöstä
- Motivaatio ohjelmistohankintaan osallistumiseen
- Kuvailu hyvästä vaatimusmäärittelytyöstä

Alkuperäiset aihealueet haastatteluryhmälle 2.

- Ajatuksia vaatimusmäärittelytyöstä ja sen sisällöstä
- Ennakkoasenteet vaatimusmäärittelytyötä kohtaan
- Kokemukset mahdollisuudesta osallistua vaatimusmäärittelytyöhön

Teemoittelussa otettiin huomioon ne haastattelun kohdat, jotka liittyivät tutkimuskysymyksiin. Ryhmähaastattelun ja analyysin tavoitteena on saada tietoa loppukäyttäjien suhtautumisesta ohjelmistojen vaatimusmäärittelytyöhön. Teemoiksi muodostuivat:

- Asennoituminen vaatimusmäärittelytyötä kohtaan

- Vaatimusmäärittelytyöhön liittyvät haasteet ja mahdollisuudet
- Vastaajien kehitysehdotukset

6 Tutkimuksen tulokset

Valitsimme tutkimuksen aiheeksi selvittää, miten ohjelmistojen vaatimusmäärittelytyöhön suhtaudutaan, ja miten osallistamista voitaisiin lisätä. Tässä luvussa esitellään tutkimuksen tulokset. Tutkimuksessa kerättiin määrällistä tietoa lomakekyselyllä ja laadullista tietoa ryhmähaastatteluilla.

Alaluvussa 6.1 esitetään loppukäyttäjien asennoituminen vaatimusmäärittelytyötä kohtaan. Alaluvussa 6.2 esitetään vaatimusmäärittelytyöhön osallistumisen haasteet ja mahdollisuudet. Viimeisessä alaluvussa 6.3 esitetään mahdollisia kehitysehdotuksia havaittuihin ongelmiin.

6.1 Asennoituminen vaatimusmäärittelytyötä kohtaan

Tämän tutkimuksen ja tutkimuskysymysten suunnittelun pohjana oli ennakko-oletus loppukäyttäjien negatiivisesta suhtautumisesta ohjelmistojen vaatimusmäärittelytyöhön. Loppukäyttäjien asennetta vaatimusmäärittelytyötä kohtaan on tutkittu varsin vähän, ja arvelimme loppukäyttäjien olevan haluttomia osallistumaan vaatimusmäärittelytyöhön. Tutkimuksen tulokset osoittautuivat kuitenkin lopulta päinvastaisiksi olettamuksiamme kohtaan. Tutkimukseen osallistuneet vastaajat olivat kiinnostuneita osallistumaan vaatimusmäärittelytyöhön, riippumatta olivatko he osallistuneet siihen aikaisemmin. Tutkimuksen mielenkiintoinen piirre oli se, että vastaajat suhtautuivat myönteisesti ohjelmistojen vaatimusmäärittelytyöhön, vaikka asenne nykyhetken tilanteeseen oli negatiivinen.

Lomakekyselyn tulosten mukaan vastaajat, jotka olivat osallistuneet vaatimusmäärittelytyöhön ja jotka ymmärsivät mitä vaatimusmäärittelytyö pitää sisällään, suhtautuivat siihen huomattavasti myönteisemmin kuin vastaajat, joille vaatimusmäärittelytyö ei ollut ennestään tuttu. Tämä tulos on esitetty esitetty liitteen taulukoissa C.25 – C.30 ja C.32 – C.37. Vaatimusmäärittelytyöhön suhtauduttiin myönteisesti myös ryhmähaastatteluissa. Kaikki haastateltavat olivat tietoisia vaatimusmäärittelytyön luonteesta. Haastateltavien ryhmä, joka ei ollut osallistunut aikaisemmin vaatimusmäärittelytyöhön, koki myös vaatimusmäärittelytyön tärkeäksi ja halusi osallistua siihen jatkossa.

Lomakekyselyn ja haastattelujen tulosten mukaan vaatimusmäärittelytyöhön aikaisemmin osallistuneilla ihmisillä on useammin positiivinen kuva mahdollisuudesta vaikuttaa tulevaan ohjelmistoon vaatimusmäärittelytyön kautta. Vaatimusmäärittelytyön sisällöstä ja tarkoituksesta tietoiset vastaajat kokivat prosessin kevyeksi ja hyödylliseksi, sekä uskoivat myös olevansa päteviä vaatimusmäärittelytyöhön. He tunsivat pystyvänsä vaikuttamaan ohjelmiston kehitykseen osallistumalla vaatimusmäärittelytyöhön. Tulos vahvistui erityisesti keskusteltaessa haastateltavien kanssa, jotka eivät aikaisemmin olleet osallistuneet vaatimusmäärittelytyöhön.

Haastateltavien motivaatio osallistumista kohtaan oli korkea, koska he tiedostivat osallistumisella olevan mahdollista vaikuttaa heidän omassa työssä käytettäviin ohjelmistoihin ja niiden käytettävyyteen. Lomakekyselyn tulokset eivät olleet täysin yhdenmukaiset tältä osin haastatteluihin nähden. Kyselyyn vastanneista ne henkilöt, jotka olivat aikaisemmin osallistuneet vaatimusmäärittelytyöhön kokivat enemmän vaikutusmahdollisuuksia. Lomakekyselyn mukaan vain aikaisemmin osallistuneet kokivat useammin, että heillä on johdon tuki vaatimusmäärittelytyöhön osallistumiseen. He tunsivat myös, että heillä on vaatimusmäärittelytyössä annettavaa.

Lomakekyselyn mukaan vaatimusmäärittelytyöhön aikaisemmin osallistuneet pitivät osallistumista hyvin harvoin kovin työläänä. Vastaajat, jotka eivät ole aikaisemmin osallistuneet vaatimusmäärittelytyöhön, eivät osanneet sanoa, onko osallistuminen työlästä. Lomakekyselyn tulokset poikkesivat haastattelujen tuloksista, joiden mukaan kaikki haastateltavat olivat sitä mieltä, että vaatimusmäärittelytyöhön osallistumisesta saadaan lopulta kulutettuun aikaresurssiin nähden suurempi hyöty. Huomioitavaa haastattelujen osalta on kuitenkin se, että kaikki haastateltavat molemmista haastatteluryhmistä olivat tietoisia siitä, mitä vaatimusmäärittelytyö tarkoittaa.

Lomakekyselyn ristiintaulukoitu aineisto on esitetty yksityiskohtaisemmin liitteessä C ja siitä on yhteenveto taulukossa C.5. Lomakekyselyn tulosten ristiintaulukoinnin mukaan vastaajan ikä, koulutusaste tai koulutusala eivät selittäneet vastaajan suhtautumista vaatimusmäärittelytyöhön. Tässä tutkimuksessa ei voitu tutkia tietokoneen käyttöaidon vaikutusta vaatimusmäärittelytyöhön asennoitumiseen. Vastaukset olivat painottuneet liitteen taulukon C.3 mukaisesti erittäin paljon vaihtoehtoon hyvä, joten tietokoneen käyttöaitoa ei voinut järkevästi käyttää hyödyksi ristiintaulukoinnissa. Vaihtoehtoja ei pystynyt yhdistämään niin, että nollahypoteesin alle viiden suuruisten oletettujen frekvenssien määrä olisi jäänyt tarpeeksi pie-

neksi. Tuloksista voidaan kuitenkin päätellä, että Lapin AMKin henkilöstön asennoituminen ohjelmistojen vaatimusmäärittelytyöhön on asiantunteva, koska vastaajan henkilökohtaisten ominaisuuksien ja osaamisen ei katsota estävän vaatimusmäärittelytyöhön osallistumista.

6.2 Vaatimusmäärittelytyöhön liittyvät haasteet ja mahdollisuudet

Loppukäyttäjien osallistaminen vaatimusmäärittelytyöhön koettiin erittäin tärkeäksi osaksi ohjelmistohankintoja. Haastateltavat molemmista haastatteluryhmistä kokivat, että loppukäyttäjiä pitäisi ottaa vaatimusmäärittelytyöhön nykyistä enemmän mukaan. Osallistumiseen suhtauduttiin myönteisesti, mutta suurimmaksi haasteeksi koettiin aikaresurssin puute. Ohjelmistojen heikko käytettävyys vaikutti henkilöstön asennoitumiseen negatiivisesti, mutta samalla lisäsi motivaatiota vaatimusmäärittelytyötä kohtaan.

6.2.1 Osallistaminen ja osallistuminen

Haastateltavat olivat selvästi sitä mieltä, että loppukäyttäjien mukaan ottaminen vasta perehdytyksessä on liian myöhäistä, eikä se sitouta loppukäyttäjiä ohjelmiston käyttöön ja opiskeluun. Haastateltavat haastattelyryhmässä 2. kokivat, että valmis ohjelmisto annetaan heille suoraan käyttöön lyhyen perehdytyksen kautta. He toivat esiin mielipiteen, jonka mukaan opettajille vain sanotaan mikä on tuleva järjestelmä, eikä kysytä vaatimuksia sitä kohtaan. Nykytilanteen ongelmallisuudesta huolimatta haastateltavilla oli myönteinen asennoituminen tilanteen parantamista kohtaan.

Loppukäyttäjien osallistamisen riittämättömyys vaatimusmäärittelytyöhön nähtiin ongelmana lomakekyselyn tuloksissa ja ryhmähaastattelujen tuloksissa. Erityisesti lomakekyselyn tuloksissa ilmeni, että vaatimusmäärittelytyöhön suhtaudutaan positiivisemmin silloin, kun vaatimusmäärittelytyön tarkoitus on ymmärretty. Tämä tulos on esitetty liitteen taulukoissa C.25 – C.30. Lomakekyselyyn vastannut henkilöstön jäsen kommentoi tietoisuuden lisäämistä:

”Vaatimusmäärittelyyn osallistuvat tarvitsisivat perustiedot siitä, mitä osallistuminen käytännössä tarkoittaa, mikäli ei ole aikaisemmin osallistunut määrittelyyn.”

Haastateltavat haastattelyryhmässä 2. tunsivat, että ohjelmistohankinnoista ko-

konaisuudessaan päätetään jossakin muualla kuin loppukäyttäjien keskuudessa. Heidän mukaansa nykytilassa toimitaan niin, että tuotantokäyttöön saatettu ohjelmisto perehdytetään loppukäyttäjille, jonka jälkeen ohjelmiston käyttäminen kuuluu osaksi työnkuvaa. Haastateltavat asennoituivat negatiivisesti siihen, että loppukäyttäjää ei osallisteta ohjelmistojen vaatimusmäärittelytyöhön. Toisaalta ohjelmistohankintojen nykytila nähtiin myös siten, että tuotantokäyttöön otettujen ohjelmistojen ei tarvitse kaikkia miellyttääkään. Tilanne, jossa jokainen käyttäjä olisi tyytyväinen käytettäviin ohjelmistoihin, koettiin mahdottomaksi. Haastateltava haastatteluryhmästä 2. totesi:

”Ehdottomasti henkilökuntaa pitäisi osallistaa enemmän.”

Ohjelmistoprojekteihin ja vaatimusmäärittelytyöhön osallistumisessa ongelmaksi nähtiin ryhmähaastattelujen tulosten mukaan työnantajan tuen puute osallistumiseen. Lomakekyselyssä n. 60% ei osannut liitteen taulukon C.4 mukaisesti sanoa tukeeko työnantaja osallistumista. Ryhmähaastattelujen perusteella työnantajan tuen puute tarkoitti aikaresurssin puuttumista vaatimusmäärittelytyöhön osallistumiseen. Erityisesti opettajat olivat sitä mieltä, että heidän on vaikea saada varsinaisen opetustehtävänsä resursoinnin lisäksi resurssia muihin organisaation tehtäviin.

6.2.2 Aikaresurssin puute

Vaikka tutkimuksen molempiin osiin osallistuneet vastaajat olivat kiinnostuneita osallistumaan vaatimusmäärittelytyöhön, he kokivat osittain, ettei heillä ollut työnantajan tukea osallistumiseen. Lomakekyselyn tulosten mukaan kokemus työnantajan tuesta liittyi tietämykseen vaatimusmäärittelytyön sisällöstä ja aiempaan vaatimusmäärittelytyöhön osallistumiseen. Tämä tulos on esitetty liitteen taulukoissa C.29 ja C.36. Ryhmähaastatteluissa työnantajan tuki liittyi tiiviimmin aikaresurssiin. Työnantajan tuki merkitsi tässä tapauksessa aikaresurssin myöntämistä vaatimusmäärittelytyöhön osallistumiseen.

Työnantajan päätöksiä kritisoitiin myös siksi, että haastateltavien mukaan loppukäyttäjää aletaan nykytilanteessa aktivoida vasta ohjelmiston perehdytysvaiheessa. Loppukäyttäjät haluaisivat vaikuttaa heidän omiin työssä käytettäviin ohjelmistoihin jo aikaisemmin, jotta ohjelmistot tukisivat työtä eivätkä vaikeuttaisi sitä.

Erityisesti molempien haastatteluryhmien lehtorit kokivat aikaresurssin olevan riittämätön vaatimusmäärittelytyöhön osallistumiseen. Opettajien työnkuvasta johtuen valtaosa työpäivistä on resursoitu opetukseen ja siihen liittyviin tehtäviin. Opet-

tajat kokivat vaikeaksi myös löytää keskenään yhteistä aikaa yhteisten opintojaksojen suunnitteluun. He kokivat, että työnantajan tukea vaatimusmäärittelytyöhön osallistumiseen ei ole, koska aikaresurssia opetuksen ulkopuoliseen työhön on vaikea saada.

Haastateltavat suhtautuivat hyvin negatiivisesti resurssipulaan, mutta olivat tietoisia, että koko henkilökuntaa ei voida osallistaa resursseja myöntämällä vaatimusmäärittelytyöihin. Organisaation projektimallin selkeyttäminen ja jalkauttaminen vaikuttaisi haastateltavien mukaan myös resurssiongelmiin helpottavasti.

Opettajat kokivat haasteena myös opetuksen ja uusien ohjelmistojen perehdyttämisaikataulujen yhteensovittamisen. Organisaation tukipalvelut auttavat käyttöönotossa jonkin verran, mutta paljon ominaisuuksia jää oppimatta ja hyödyntämättä. Resurssikysymys vaatimusmäärittelytyöhön nähtiin tärkeäksi, koska aikaresurssin myöntäminen kehittää organisaation toimintaa ja mahdollistaa perehtymisen kehitettävään ohjelmistoon ja ongelmaan.

Ainoana esteenä vaatimusmäärittelytyöhön osallistumiseen koettiin siten ajankäyttö. Haastateltavien mukaan tarve ohjelmistohankinnalle pitäisi nähdä organisaation johdossa, koska heidän mielestään suorittava taso ei voi määrätä mitä tehdään ja miten järjestelmiä kehitetään.

6.2.3 Ohjelmistojen käytettävyyteen vaikuttaminen

Keskusteltaessa vaatimusmäärittelytyön sisällöstä ja siitä, millainen mielikuva haastateltavilla siitä oli, korostui käytettävyyden huomioiminen ohjelmistohankinnoissa. Ohjelmistojen vaatimusmäärittelytyö nähtiin sisällöltään etenkin käyttäjäystävälliseen lopputulokseen tähtäävänä. Vaatimusmäärittelytyön tärkeinä huomioina koettiin kehitettävän ohjelmiston käytettävyys ja luotettavuus, sekä tiedon säilyvyys, saatavuus ja eheys. Käytettävyydellä tarkoitettiin myös helposti opittavuutta esimerkiksi sellaisissa tapauksissa, joissa loppukäyttäjä on kokematon tai vähän kokenut tietokoneen käytössä. Ohjelmiston tulisi ohjata käyttöä ja käyttäjää siten, että ohjelmiston käyttö olisi mahdollisimman sujuvaa.

Haastateltavat kokivat heidän työssään käytettävien ohjelmistojen olevan huonosti suunniteltuja ja osittain hankaloittavan heidän työtään. Etenkin käyttöliittymäsuunnittelua ja käytössä olevien oikeiden termien käyttämistä haluttaisiin parantaa jatkossa. Haastateltavien mukaan vaatimusmäärittelytyössä kartoitetuista vaatimuksista pitäisi neuvotella loppukäyttäjien kanssa. Haastateltava haastatteluryhmästä 2. arvosteli nykyistä toimintatapaa:

”Meidän työnkuvaan kuuluu, että uusi järjestelmä pitää ottaa käyttöön, vaikka sille ei näkisi tarvetta.”

Vaatusmäärittelytyö ymmärrettiin selkeästi vaiheena, jonka huolella suorittaminen vaikuttaa vähentävästi muutostöiden tarpeeseen käyttöönoton jälkeen. Vaatusmäärittelytyön tarkoituksena oli haastateltavien mielestä varmistaa, että kehitettävä ohjelmisto vastaa loppukäyttäjän tarvetta. Vaatusmäärittelytyön jälkeen tulisi myös arvioida voidaanko vaatimuksista tuottaa toimiva ohjelmisto ja onko alkuperäinen idea kuitenkin lopulta toimiva. Haastateltava haastatteluryhmästä 2. ehdotti:

”Jo alkuun voitaisiin kehitellä alusta sellaiseksi, että se on helppokäyttöinen ja kaikki löytyy mitä tarvitaan.”

Perinteiset dokumentaatioon vahvasti tukeutuvat mallit koettiin nykyaikaan soveltumattomiksi. Erityisesti raskaan dokumentaation luomista ei nähty enää mielekkääksi. Perinteisen ohjelmistokehityksen tunnistettiin kuluttavan paljon aikaa, jonka seurauksena ohjelmiston valmistuessa asiakkaan ja loppukäyttäjän tarpeet ovat voineet jo muuttua. Ketterät menetelmät korostuivat vaatusmäärittelytyöstä keskusteltaessa. Sen jälkeen kun tavoitteet on määritetty, tulisi työtä tehdä priorisoiden, mikä on tärkeintä ja mitä pitää ensimmäisenä saada aikaan.

6.3 Vastaajien kehitysehdotukset

Tutkimuksen tulosten mukaan parannusehdotuksena vaatusmäärittelytyöhön parempaan suhtautumiseen on henkilöstön tietoisuuden lisääminen vaatusmäärittelytyön sisällöstä ja tarkoituksesta. Lisäksi osallistamista pitäisi lisätä ja varmistaa aikaresurssi osallistumista varten. Lomakekysely tuotti selkeän yhteyden vaatusmäärittelytyön tietämyksestä ja myönteisestä asenteesta sitä kohtaan. Ryhmähaastatteluissa yhteys näkyi myös haastateltavien myönteisenä asenteena.

6.3.1 Loppukäyttäjien osallistaminen vaatusmäärittelytyöhön

Molemmat haastatteluryhmät kokivat, että ohjelmistohankinnan käynnistyttyä loppukäyttäjiä pitäisi osallistaa määrittelytyöhön alusta alkaen. Haastateltavat arvelivat, että vaatusmäärittelytyöhön osallistuminen helpottaisi sitoutumista ohjelmiston käyttöön myöhemmin. Paremman sitoutumisen lisäksi käytettävyyden kehittäminen nähtiin mahdollisuutena vaatusmäärittelytyön kautta. Haastateltavat

kokivat kuitenkin, että vaatimusmäärittelytyöhön osallistumisen täytyisi perustua vapaaehtoisuuteen. Haastateltava haastatteluryhmästä 1. ehdotti:

”Osallistuminen täytyisi perustua vapaaehtoisuuteen. Voisi kysyä, että ketkä ovat vapaaehtoisia ja sitten ottaa sieltä pienen porukan ja selvittää onko mahdollisuuksia osallistua.”

Ryhmähaastatteluihin osallistuneet haastateltavat ideoivat erilaisia tapoja lisätä loppukäyttäjien tietoisuutta käynnissä olevista ohjelmistohankinnoista ja niihin osallistamisesta. Henkilöstöinfoissa ja osaamisalojen palaverissa voitaisiin kysyä vapaaehtoisia vaatimusmäärittelytyöhön osallistujia, jonka jälkeen muodostettaisiin sopiva ryhmä ja annettaisiin osallistujille aikaresurssi työtä varten. Kasvokkain tapahtuvien tilaisuuksien hyödyntäminen vapaaehtoisten kartoittamiseksi koettiin tehokkaammaksi ja miellyttävämmäksi tavaksi kuin sähköpostitse lähetettävä kysely. Lisäksi erityiset ohjelmistojen kehittämiseen ja eri ohjelmistovaihtoehtojen testaamisen järjestetyt tilaisuudet arveltiin toimiviksi vaihtoehdoiksi.

Vaatimusten kartoittamista kyselyjen avulla esitettiin myös vaihtoehtona loppukäyttäjien kuulemiseen. Kyselyiden avulla voitaisiin selvittää, mikä olisi kaikkein tärkeintä tulevassa ohjelmistossa, ja millaisia ongelmia sen tulisi ratkaista. Kyselyjen suhteen todettiin kuitenkin, että niitä tulee työsähköpostiin aiheesta riippumatta liikaa. Koska erilaiset sähköiset lomakkeet ja kyselyt ovat menettäneet vaikutavuuttaan, ehdotettiin vaihtoehdoksi ohjattuja tilaisuuksia. Haastateltava haastatteluryhmästä 1. pohti kyselyitä:

”Kyselyitä tulee aika paljon, niihin turtuu. Jotenkin nostettaisiin esille, että kun vastaat tähän niin se on tärkeää.”

Ohjatuissa tilaisuuksissa kerrottaisiin ajankohtainen tilanne organisaation ohjelmistohankinnoissa, jonka jälkeen kaikki asiasta kiinnostuneet voisivat äänestää ja kertoa mitä halutaan ja tarvitaan. Tilaisuudet voisivat olla helposti lähestyttäviä kaikille avoimia tapahtumia, joissa tarjottaisiin kahvia ja annettaisiin mahdollisuus vaikuttaa. Haastateltavien näkökulmasta ohjelmistohankinta menisi silloin reilusti loppukäyttäjien ja johdon osalta. Kun loppukäyttäjät saisivat vaikuttaa ohjelmistoon tai vaihtoehtoihin, myös johdon tekemä päätös tukisi loppukäyttäjien tarpeita. Toisaalta myös valinnasta päättävä johto voisi olla varma, että hankinta on tarpeiden mukainen.

Vaatimusmäärittelytyöhön osallistuvilla loppukäyttäjillä ei haastateltavien mielestä tarvitse olla erityistä osaamista tietotekniikasta tai ohjelmistoprojekteista, vaan

riittää, että he hallitsevat oman työprosessinsa. Prosessin tuntemuksen lisäksi vaatimusmäärittelytyöhön osallistuvalla tulisi olla kyky hahmottaa, mikä voisi olla uudessa ohjelmistossa paremmin kuin aikaisemmin. Toimintaprosessin kehittämisen lisäksi vaatimuksia ei pitäisi jättää liian alhaisiksi, vaan rohkeasti miettiä, kuinka työskentelyä olisi mahdollista sujuvoittaa uusien toimintatapojen myötä.

Haastateltavat kuvailivat, että vaatimusmäärittelytyöhön osallistuvan loppukäyttäjän pitäisi ymmärtää, että hänen työprosessiaan ja työkalujaan ollaan kehittämässä. Innovatiiviset luonteenpiirteet edesauttavat näkemään kehityskohtia nykyisissä toimintaprosesseissa. Ohjelmistojen hyödyllisyys työtehtävän hoitamisessa tunnistettiin vaikuttavan omaan työskentelyyn, ja osallistumisen tärkeys vaatimusmäärittelytyöhön koettiin sitä kautta tärkeäksi vaikuttamistavaksi.

Sidosryhmien tunnistamisen tärkeys havaittiin oleellisena osana vaatimusmäärittelytyötä. Haastateltavat totesivat, että ainoastaan opettajien vaatimusten kartoittaminen ei riitä, vaan vaatimukset pitää kerätä myös opiskelijoilta siinä tapauksessa, kun ohjelmiston loppukäyttäjät ovat myös opiskelijat.

Perinteisen vaatimusmäärittelytyöhön osallistumisen sijaan molemmat haastatteluryhmät korostivat eri ohjelmistovaihtoehtojen kokeilemisen tärkeyttä ja prototyyppien kehittämistä. Haastateltavat kokivat riittäväksi osallistamiseksi vaihtoehtoihin vaikuttamisen, jolloin esimerkiksi kolme yhtä hyväksi havaittua ohjelmistoa esitellään johdolle, jonka jälkeen johto valitsee hankittavaksi niistä yhden. Haastateltava haastatteluryhmästä 2. korosti koekäytön tärkeyttä:

”Pitäisi olla esittely, jonka avulla voisi tutustua tuotteeseen. Loppukäyttäjien pitäisi kokeilla, tehdä koekäyttöä.”

Prototyyppityksen etuna nähtiin sen helppous ja erityisesti se, että kaikki ryhmän jäsenet puhuvat samaa kieltä prototyyppin kautta. Yhteisen ymmärryksen saavuttaminen arveltiin olevan mahdollista prototyyppityksen avulla. Haastateltavat pohtivat, että jos valmisohjelmiston hankinnasta on päätetty, voitaisiin siitä järjestää testikäyttö ennen lopullista tuotantokäyttöön siirtymistä. Koekäytöllä arveltiin saavutettavan myös hyötyä sopimuksen suhteen, testikäytön avulla saavutetaan ymmärrys siitä mitä tilattuun kokonaisuuteen sisältyy.

6.3.2 Aikaresurssin merkitsevyys ohjelmistohankinnoissa

Vaatimusmäärittelytyöhön osallistumisen koettiin tuottavan ainutlaatuista tietoa työnantajalle organisaation tilasta ja tarpeista. Vaatimusmäärittelytyöhön panosta-

misen arveltiin maksavan itsensä takaisin ohjelmistojen käyttöönoton jälkeisessä ylläpitovaiheessa. Kustannussäästöjä voisi syntyä, kun ohjelmisto määritettäisiin hankintavaiheessa riittävän hyvin, jonka seurauksena kalliita muutostöitä ei jälkikäteen olisi tarpeen toteuttaa. Loppukäyttäjille projektin alussa annettu aikaresurssi vaatimusmäärittelytyöhön maksaisi siten itsensä takaisin. Haastateltavat arvelivat, että organisaation johto päättäisi henkilöresurssien määrästä ja myöntäisi sitä vaatimusmäärittelytyötä varten.

Haastatteluryhmässä 1. pohdittiin, että jokaisen ohjelmistohankinnan tulisi edetä saman kaavan mukaan. Raharesurssin pitäisi kokonaisuudessaan olla organisaation johdon hallussa, joka ohjaisi oikean toimintamallin hyödyntämiseen. Organisaation määrittämän prosessikuvauksen mukainen eteneminen ohjelmistohankinnassa saavuttaisi johdon tuen ja mahdollistaisi siten resurssin saamisen. Toimintatavan hyödyntämättä jättäminen puolestaan eväisi resurssien saavuttamisen ja siten ohjelmistohankinnan toteutumisen. Aikaresurssi opetuksen ulkopuolisiin tehtäviin osallistumiseen tulisi haastateltavien mielestä kirjata työaikasunnitelmaan ja yhteiset tilaisuudet vaatimusmäärittelytyöhön sekä ohjelmistojen perehdyttämiseen pitäisi sopia hyvissä ajoin.

Koko henkilökunnan osallistumista vaatimusmäärittelytyöhön ei nähty realistisena, vaan ymmärrettiin, että jokaisen henkilön kuuleminen Lapin AMKin kokouksessa organisaatiossa on mahdotonta. Sen sijaan, että yritettäisiin kuulla jokaisen mielipide, tulisi vaatimusmäärittelytyöhön valita huolella ihmiset, jotka edustavat omaa sidosryhmäänsä. Vaatimusmäärittelytyöhön valituilla sidosryhmien edustajilla tulisi olla aikaresurssi osallistumista varten. Haastateltava haastatteluryhmästä 1. pohti:

”Kun vaatimuksia aletaan hiomaan ja miettimään ei voida 500 hengen työpanosta valita, koska työaikaa tarvitaan.”

6.3.3 Ohjelmistohankintojen suunnittelu

Haastatteluryhmä 1. tunnisti laatutyön liittyvän vaatimusmäärittelytyöhön. Ohjelmistohankintojen kuuluisi heidän mielestään sisältyä normaaliin laatutyöhön. Keskusteltaessa ohjelmistohankinnan tarpeen heräämisestä, haastateltavat eivät osanneet sanoa, mistä tai kenen kautta prosessi hankinnalle käynnistetään. Prosessikuvauksen kehittämistä pohdittiin yhdeksi vaihtoehdoksi tietoisuuden lisäämiseen ohjelmistohankintojen menettelytavasta. Esimerkkinä nostettiin opetuksesta nouse-

va tarve jollekin ohjelmistolle. Haastateltavat eivät tienneet, kuinka tilanteessa tulisi toimia, joten he arvelivat että mahdollisesta prosessikuvauksesta voisi katsoa menettelytavan tai yhteyshenkilön keneen ottaa yhteyttä. Lomakekyselyyn vastannut henkilöstön jäsen mietti kommunikaation helpottamista loppukäyttäjän ja ohjelmistohankinnasta vastaavan välillä:

”Varmaan olisi hyvä olla joku tulkki ohjelmiston suunnittelijan ja toisaalta ohjelmiston tulevan käyttäjän välillä. Maailmankuvat ja näkökulmat ovat usein niin erit, että yhteistä säveltä ei meinaa löytyä.”

Eräs haastateltavista toi esiin huolenaiheen ohjelmistohankinnan käynnistämiseen johtavasta esiselvityksestä. Vaatimusmäärittelytyössä pitäisi määrittää ne ongelmat, jotka kehitettävällä ohjelmistolla pyritään ratkaisemaan. Tärkeäksi koettiin etenkin se, että hankintaan pitäisi käynnistyä vasta sen jälkeen, kun on tunnistettu ongelma, joka halutaan ratkaista. Erityisesti nykytilan kuvaus ja tavoitetilan määrittäminen koettiin vaiheiksi, jotka pitäisi suorittaa, mutta jotka jätetään liian usein tekemättä nykyhetken toimintamallissa. Haastateltava haastatteluryhmästä 1. toivoi henkilöstön rohkaisemista vaatimusmäärittelytyössä:

”Pitää olla rohkeutta asettaa tavoitteet riittävän korkealle. Ei pidä himmailla, että tehdään vähän jotakin. Jos tehdään vain ihan vähän paremmaksi eikä uskalleta sanoa, että haluan vaikka automatisoida tämän osan, että käsityötä jäisi pois. Aina ei täysin tavoitteisiin päästä, mutta pitäisi olla visio, se on kaikkein tärkein.”

Tavoitteiden ja maalin määrittämistä korostettiin haastatteluissa onnistumisen edellytyksenä. Tavoitteiden määrittäminen edellyttää ongelmien tunnistamista, jotka ohjelmiston avulla halutaan ratkaista. Haastateltavat totesivat, että nykytilassa ohjelmistoja hankintaan miettimättä syytä hankinnalle. Ohjelmistolle aletaan miettiä käyttötarkoitusta ja -tapoja vasta käyttöönoton jälkeen.

7 Tulosten pohdinta

Henkilökunta kokee selvästi haluavansa osallistua enemmän ohjelmistojen vaatimusmäärittelytyöhön, kuin mihin heille annetaan mahdollisuus. Asenne vaatimusmäärittelytyötä kohtaan on innostunut, vaikka mahdollisuuksia osallistua ei olisi aikaisemmin ollut. Yllättävänä huomiona on suuri kiinnostus testikäyttöä ja eri ohjelmistovaihtoehtojen kokeilua kohtaan. Tutkimukseen osallistuneet vastaajat korostivat loppukäyttäjien osallistamisen tärkeyttä ja osallistamisen määrää toivottiin lisäävän nykyisestä. Vastaajat tunnistivat myös vaatimusmäärittelytyöhön osallistumiseen liittyviä esteitä.

Tässä luvussa keskustellaan tuloksista tutkimuskysymysten näkökulmasta. Alaluvussa 7.1 pohditaan loppukäyttäjien suhtautumista vaatimusmäärittelytyöhön Lapin AMKissa. Alaluvussa 7.2 esitetään havaittuihin ongelmakohtiin kehittämisehdotuksia. Alaluvussa 7.3 tutkimuksen tuloksia verrataan aikaisempien tutkimusten tuottamiin tuloksiin. Viimeisessä alaluvussa 7.4 pohditaan tutkimuksen luotettavuutta ja toistettavuutta.

7.1 Asennoituminen vaatimusmäärittelytyötä kohtaan

Loppukäyttäjien motivaatiota vaatimusmäärittelytyöhön osallistumiseen on tutkittu verrattain vähän. Alaluvussa 4.4 käsiteltiin loppukäyttäjän motivaatioon vaikuttavia tekijöitä. Loppukäyttäjän motivaatiota parantaa se, että hän kokee saavuttavansa jotakin hyötyä vaatimusmäärittelytyöstä. Loppukäyttäjän tietämys vaatimusmäärittelytyön sisällöstä ja vaatimusmäärittelytyöstä koettu hyöty edistävät kokonaisuudessaan myönteistä suhtautumista vaatimusmäärittelytyötä kohtaan.

Yhtenä syynä motivaation puuttumiseen ohjelmistohankintojen suhteen voi olla myös alaluvussa 4.4 tunnistettu epävarmuus omista tietoteknisistä taidoista. Loppukäyttäjät, jotka kokevat omaavansa ei-tekniikan taustan, voivat tuntea osallistumisensa vaatimusmäärittelytyöhön arvottomaksi. Näin ei kuitenkaan koettu tässä tutkimuksessa, vaan haastateltavat kokivat osallistumisen olevan tuloksekas silloin, kun osallistuja tuntee oman työprosessinsa ja on kiinnostunut kehittämään sitä. Kehitettävän ohjelmiston loppukäyttäjä on osaavin henkilö kertomaan oman työpro-

sessinsa sisällön, jonka haastateltavat toivat esiin keskustelun aikana. Toimintaprosessin tuntemus tarkoittaa usein myös ongelmakohtien tunnistamista omassa työssä, joka toimii erinomaisena lähtökohtatietona esiselvitysvaiheessa ohjelmistohankinnan tarvetta selvittäessä.

Alaluvussa 4.4 esitetyn aikaisemman tutkimustiedon mukaan myönteistä asennetta vaatimusmäärittelytyötä kohtaan lisäävät tilanteet, joissa loppukäyttäjä on kokenut, ettei hänen mielipidettään oltu kysytty. Huonosti käytettävät ohjelmistot voivat lisätä kiinnostusta osallistua ohjelmistojen vaatimusmäärittelytyöhön. Käytettävyys nousi vahvasti ryhmähaastatteluissa esiin, jonka myötä se nousi aineiston analyysissä myös yhdeksi teemaksi. Tutkimuksen tulokset olivat aikaisemman tutkimuksen kanssa hyvin samassa linjassa.

Viitteitä tietoteknisten taitojen yhteydestä vaatimusmäärittelytyöhön asennoitumiseen ei saavutettu tässä tutkimuksessa, koska valtaosa vastaajista arvioi oman tietokoneen käyttötaitonsa hyväksi. Aikaisempi tutkimus ja tämän tutkimuksen tulokset vahvistavat tietoa, jonka mukaan loppukäyttäjän tietämys vaatimusmäärittelytyön sisällöstä ja vaatimusmäärittelytyöstä koettu hyöty edistävät kokonaisuudessaan myönteistä suhtautumista vaatimusmäärittelytyötä kohtaan.

7.2 Kehitysehdotukset vaatimusmäärittelytyöhön osallistamiseen

Ryhmähaastatteluihin osallistuneet haastateltavat toivat esiin useita nykyhetkeen liittyviä haasteita ohjelmistohankinnoissa ja vaatimusmäärittelytyöhön osallistumisessa. Kaikki huomiot eivät liity suoraan vaatimusmäärittelytyöhön, mutta ne vaikuttavat loppukäyttäjien asennoitumiseen vaatimusmäärittelytyötä kohtaan. Havaittuihin huomioihin reagoimalla asenteisiin voitaisiin vaikuttaa myönteisesti.

Ohjelmistohankintojen suunnittelu

Haastatteluissa nousi selkeästi esiin tyytymättömyys ohjelmistohankintoja kohtaan. Nykytilanne ohjelmistohankintojen suhteen oli haastateltavien mukaan se, että ohjelmistoja hankitaan miettimättä syytä hankinnalle ja tekemättä vaatimusmäärittelyä. Ohjelmistohankinnan tarpeen selvittämistä käsiteltiin alaluvussa 2.2.1. Tarpeen selvittäminen ennen vaatimusten keräämistä ja projektin aloittamista vaikuttaa merkittävästi käytettävyyteen ja käyttäjätarpeeseen vastaamiseen käyttöönottavaiheessa. Nykytilan tutkiminen antaa tietoa ratkaistavasta ongelmasta ja sen laajuudes-

ta. Alkuvaiheen esiselvityksessä voidaan myös huomata, että ohjelmistohankinta ei välttämättä ole edes tarpeellinen.

Haastateltavat arvelivat tyytyväisyyden uusia ohjelmistoja kohtaan olevan korkeampi silloin, kun vaatimukset on kerätty suoraan loppukäyttäjiltä. Sitoutumista kehitettyyn ohjelmistoon käsiteltiin alaluvussa 3.2.3. Loppukäyttäjien sitoutumisen on tutkittu olevan suurempi niissä tapauksissa, kun loppukäyttäjää on osallistettu vaatimusmäärittelytyöhön. Ohjelmiston käyttöönoton jälkeinen sitoutumisen määrä on vahvasti sidoksissa projektiin osallistumiseen.

Haastateltavien mukaan osallistumisen ei välttämättä tarvitse liittyä vaatimusmäärittelyprosessiin tai edes ohjelmistoprojektiin, vaan myös eri ohjelmistovaihtoehtojen testikäyttö lisää sitoutumisen ja tyytyväisyyden määrää. Haastateltavat tunnistivat työvälleinä käytettävien ohjelmistojen kautta ongelmakohteet, jotka he olisivat toivoneet toteutettavan heidän ammattiosaamistaan hyödyntäen. Haastateltavat ilmaisivat vahvasti tahtotilansa vaatimusmäärittelytyöhön osallistumiseen jatkossa organisaation hankkiessa uusia ohjelmistoja.

Haastattelujen tulosten mukaan organisaation toiminnassa myös ohjelmistoprojekteissa pitäisi miettiä auttaako ohjelmiston hankinta saavuttamaan yhteiset tavoitteet. Ohjelmistohankinnoissa pitäisi kyetä miettimään innovatiivisesti mikä voisi ohjelmiston avulla olla paremmin kuin aikaisemmin. Liiketoimintatarpeita käsiteltiin alaluvussa 3.2.3. Ohjelmistoprojektin tarpeellisuus tulee käyttäjätarpeiden lisäksi osoittaa myös liiketoiminnan näkökulmasta. Kehitettävän ohjelmiston tulisi tukea organisaation toimintaprosesseja ja strategian toteutumista.

Tyytymättömyyteen ohjelmistohankintoja kohtaan voisi vaikuttaa organisaation yhteisellä ohjelmistohankintojen projektimallilla. Ohjelmistohankintojen suunnittelun ja projektimallin avulla voitaisiin välttää tilanteet, joissa hankittu ohjelmisto ei vastaa asiakastarvetta. Projektimallin myötä tarpeen selvitys ja ohjelmiston suhde organisaation strategiaan tulisivat tutkittua. Malli voisi edistää myös kaikkien sidosryhmien kuulemista vaatimusmäärittelytyössä ja testikäytössä. Sitoutuminen tulevan ohjelmiston käyttöön vahvistuisi osallistumisen myötä luonnollisesti.

Käytettävyys

Etenkin henkilöt, joilla ei ollut aikaisempaa kokemusta vaatimusmäärittelytyöstä, kokivat, että moni nykyisistä organisaation ohjelmistoista on huonosti käytettävä. Käytettävyyden parantamiseksi toivottiin loppukäyttäjien osallistamista jo projektin alkuvaiheessa. Haastateltavat ideoivat heille mieleistä tapaa tuoda esiin omia

vaatimuksiaan, joista prototyyppien kehittäminen nähtiin selkeimpänä toimintamallina. Prototyypitystä käsiteltiin alaluvuissa 2.4.4, 3.2.4 ja 4.5. Prototyyppi on käsitteenä tuttu muistakin käyttöympäristöistä kuin ohjelmistokehityksestä, jonka vuoksi se on helposti ymmärrettävä monialaisissa käyttäjäryhmissä.

Ohjelmistojen käytettävyyteen ja käyttöliittymien selkeyteen voidaan vaikuttaa loppukäyttäjien avulla vaatimusmäärittelytyön ja testauksen kautta. Ohjelmiston loppukäyttäjät ovat sidosryhminä ohjelmistoprojektin tärkein voimavara ja tiedonlähde. Käytettävyyteen liittyvät vaatimukset, kuten kaikki muutkin vaatimukset, kartoitetaan riippumatta menetelmästä aina projektin alkuvaiheessa. Ketterissä menetelmissä kaikkiin vaatimuksiin voidaan vaikuttaa huomattavasti helpommin myös projektin alkuvaiheen jälkeen iteraatiokierrosten avulla.

Ketterien menetelmien hyödyntäminen myös valmisohjelmistojen mukauttamisessa ja käytettävyyden parantamisessa voisi tuottaa apua loppukäyttäjien osallistamiseen ja ohjelmiston parempaan vastaanottoon loppukäyttäjien keskuudessa. Ohjelmistoprojektimalleista ketterät menetelmät pyrkivät kaikessa toiminnassaan tuottamaan mahdollisimman suurta hyötyä asiakkaalle ja loppukäyttäjälle, samalla osallistaen heitä läpi projektin.

Iteroiva työskentelytapa voi olla helpompi ei-teknisille loppukäyttäjille, koska mukautettavaa tai kehitettävää ohjelmistoa voi näyttää heille jokaisen iteraatiokierroksen aikana. Ohjelmiston ja sen mahdollisuuksien näkeminen voi auttaa vaatimusten täsmentämisessä. Ohjelmiston testaaminen ja näkeminen jo kehitysvaiheessa lisää sitoutumista sekä henkilökohtaisesti että sidosryhmän edustajan kautta myös muulle sidosryhmälle.

Käytettävyyden parantamiseen jatkossa voitaisiin vaikuttaa myös organisaation projektimallin avulla. Prototyyppien rakentaminen tai iteratiivinen kehittäminen tukisivat käytettävyyteen liittyvien vaatimusten keräämistä. Prototyypitystä ja iteratiivista kehitystä voi hyödyntää yhdessä tai erikseen.

Tietoisuuden lisääminen vaatimusmäärittelytyöstä

Ryhmähaastattelujen tulosten mukaan johdon sitoutuminen ohjelmistohankintoihin ei ole nykyhetkellä riittävä. Jotta johdon sitoutumista ohjelmistohankintoihin voisi edistää, tulisi ohjelmistohankinnat pystyä esittämään heille vakuuttavasti ja selkeänä kokonaisuutena. Tähän ratkaisuna voisi olla ohjelmistohankintojen koordinoiminen esimerkiksi tietohallinnon toimesta. Yhdenmukaisuuden ja seurattavuuden vuoksi ohjelmistohankinnoissa kannattaisi hyödyntää organisaatiossa sa-

maa projektimallia.

Johdon sitoutuminen ohjelmistoprojekteihin on tunnistettu merkittäväksi tekijäksi projektin onnistumisen kannalta. Johdon sitoutumista käsiteltiin alaluvuissa 3.2.3, 4.2 ja 4.6. Sitoutumisen saavuttamiseksi projektin tulee olla uskottava ja sen tuoma hyöty organisaatiolle tulee olla konkreettinen. Ohjelmistohankintojen tulee myötäillä organisaation strategiaa, jonka myötä johdon tuki on myös saavutettavissa. Organisaation projektimallin noudattaminen puolestaan lisää mahdollisuutta hallita organisaatiossa yhtä aikaa käynnissä olevia projekteja ja niiden resursseja.

Organisaation kannattaisi painottaa vaatimusmäärittelytyön merkitystä ja tarkoitusta johdolle ja loppukäyttäjille. Perehdyttäminen saattaa jo itsessään parantaa suhtautumista selkeyttämällä miksi vaatimusmäärittelytyöhön osallistumisesta on hyötyä loppukäyttäjälle. Alaluvun 4.4 mukaisesti loppukäyttäjät, jotka ymmärtävät mikä vaatimusmäärittelytyö on, ovat innokkaampia osallistumaan vaatimusmäärittelytyöhön myös tulevaisuudessa. Myös lomakekyselyn tulosten mukaan tieto vaatimusmäärittelytyön sisällöstä muuttaa suhtautumista vaatimusmäärittelytyöhön huomattavasti myönteisempään suuntaan.

Perehdyttämisen ydinviestinä tulisi kertoa, että ohjelmistojen vaatimusmäärittelytyöllä pyritään kehittämään ohjelmistoista paremmin tarvetta vastaavia ja helpottamaan loppukäyttäjän tehtävän suorittamista. Erityisesti organisaation johdolle olisi hyödyksi selvittää, mitä hyötyjä aikaresurssin myöntämisestä vaatimusmäärittelytyötä varten saavutetaan. Loppukäyttäjien perehdyttämiseen liittyvä tärkeä viesti on se, että vaatimusmäärittelytyöhön osallistumiseen ei vaadita erityisesti tietoteknistä osaamista. Teknisen osaamisen sijaan tärkeämpää on tuntea se työprosessi, jota ohjelmiston avulla tehostetaan.

7.3 Tulokset ja aikaisempi tutkimus

Luvussa 4 käsiteltiin loppukäyttäjän suhtautumista vaatimusmäärittelytyöhön. Luvussa käytiin läpi aikaisempaan tutkimukseen perustuen tekijöitä, jotka lisäävät kiinnostusta osallistumista kohtaan. Luvussa esitettiin myös tekijöitä, jotka voivat vaikuttaa negatiivisesti loppukäyttäjän kiinnostukseen osallistua vaatimusmäärittelytyöhön. Tämän tutkimuksen tulokset poikkesivat osittain aikaisemmasta tutkimustiedosta. Vastaajat Lapin AMKissa eivät pitäneet teknistä osaamista ehtona vaatimusmäärittelytyöhön osallistumiseen. He tiedostivat, että vaatimusmäärittelytyössä ei tarvita teknistä osaamista, eivätkä siten pitäneet osallistumista pelottavana

tai vaikeana. Vastaajat eivät kokeneet myöskään oman mielipiteensä esiintuomista haasteena tai estävänä tekijänä osallistumiseen. Taustateoriassa esitetty työyksiköiden välinen kilpailu ei noussut esiin tutkimuksen tuloksissa.

Aikaisempaan tutkimustietoon peilaten tässä tutkimuksessa nousi vahvasti esiin aikaresurssin puute vaatimusmäärittelytyöhön osallistumiseen. Aikaresurssin puute havaittiin alaluvussa 3.1 olennaiseksi tekijäksi ohjelmistoprojektien onnistumisessa. Aikaresurssi liittyy taustateoriassa loppukäyttäjien saatavuuteen ja johdon tuen puutteeseen. Johdon tuki ja aikaresurssi liittyvät kiinteästi toisiinsa. Jos ohjelmistoprojekti nauttii johdon luottamuksesta ja se nähdään vakuuttavana, todennäköisesti myös resurssia projektia varten myönnetään suopeammin.

Suurimmat poikkeavuudet aikaisempaan tutkimukseen liittyivät positiiviseen yleisasenteeseen vaatimusmäärittelytyötä kohtaan. Yksilöiden tai yksiköiden välillä ei ollut havaittavissa kilpailua vaatimusten suhteen, vaan kaikki tutkimukseen osallistuneet halusivat yhteisesti parempaa kaikille. Pelkoa omien työtehtävien katoamisesta tietotekniikan myötä ei havaittu, kuten ei vastarintaa vaatimusmäärittelytyötä tai toimintatapojen kehittämistä kohtaan.

7.4 Tutkimuksen luotettavuus ja toistettavuus

Alaluvussa 5.4 käsiteltiin tutkimuksen luotettavuutta ja toistettavuutta. Tämän tutkimuksen perusjoukon monimuotoisuus ja ryhmähaastattelujen suppea otanta vaikuttavat reliabiliteettiin eli luotettavuuteen. Tutkimuksen lomakekysely saavutti tavanomaista suuremman vastausprosentin. Noin joka neljäs henkilöstön jäsen vastasi lomakekyselyyn. Lomakekyselyn tulokset ovat luultavasti paremmin luotettavat ja toistettavissa, kuin ryhmähaastattelujen tulokset. Toisessa ryhmähaastattelussa yksi haastatteluun kutsutuista henkilöistä ei voinut osallistua. Ryhmähaastattelujen tulokset olisivat voineet olla myös poikkeavat, mikäli kaikki kutsutut henkilöt olisivat voineet osallistua.

Lomakekyselyn validiteettiin liittyvä suurin ongelma oli, että vastaajat eivät välttämättä vastanneet loppukäyttäjän näkökulmasta. Lomakekysely osoitettiin kaikille Lapin AMKin päätoimisille henkilöstön jäsenille, joista osa toimii tai on toiminut muussakin kuin loppukäyttäjän roolissa. Vastauksista olisi voinut rajata analysoitavaksi vain vastaukset, joissa vastaaja oli osallistunut vaatimusmäärittelytyöhön ainoastaan loppukäyttäjänä. Tällöin otanta olisi kuitenkin jäänyt liian pieneksi määrälliseen tutkimukseen. Tutkimusongelmien kannalta olisi ollut olennaista saa-

da vastaukset nimenomaan loppukäyttäjän näkökulmasta. Sitä olisi voinut painottaa lomakekyselyn saatteessa enemmän.

Lomakekyselyn vastausvaihtoehtoja olisi kannattanut yksinkertaistaa, sillä nyt joidenkin vaihtoehtojen frekvenssit jäivät hyvin alhaisiksi. Myös "En osaa sanoa" -vaihtoehto keräsi aika paljon vastauksia. Kohdat jotka sisälsivät tämän vaihtoehdon olisi voinut muotoilla paremmin niin, että "En osaa sanoa" olisi sisältynyt "Kyllä" tai "Ei" vaihtoehtoon. Lomakekyselyn laatimisessa näkyi tutkimuksen tekijöiden kokemattomuus lomakekyselyjen toteuttamisesta.

Lomakekysely kannattaisi suorittaa suuremmalla otannalla, jotta vastauksista voitaisiin suodattaa esimerkiksi loppukäyttäjänä osallistumisen suhteen tai jättämällä pois "Ei osaa sanoa" -vastaukset. Suuremmalla otannalla saataisiin riittävän suuria frekvenssejä ristiintaulukointia varten jopa suodatetuista vastauksista.

Tutkimuksen toistettavuudessa on huomioitavaa, että tutkimuksen laadullisessa osassa suoritettiin vain kaksi ryhmähaastattelua. Haastateltavat valittiin niiden henkilöiden joukosta, jotka olivat ilmaisseet suostumuksen haastattelua varten. Suostumuksen antaneet henkilöt voivat itsessään olla jo ryhmä, joka poikkeaa perusjoukosta. Jos tutkimus suoritettaisiin uudelleen, voisivat tulokset olla poikkeavat tämän tutkimuksen tuloksista. Tuloksiin voi vaikuttaa haastateltavien valinta ja tutkijan suhde haastateltaviin.

Ryhmähaastatteluja tai yksilöhaastatteluja olisi voinut pitää enemmänkin kuin tutkimuksessa suoritettut kaksi ryhmähaastattelua. Erityisesti johtavassa asemassa olevien henkilöiden haastattelusta olisi voitu saada kiinnostavaa vertailukohtaa muihin ryhmähaastatteluihin. Haastattelut olisivat voineet tuottaa tietoa johdon tietoisuuden määrästä koskien ohjelmistoprojekteja ja niiden vaatimusmäärittelytyötä. Loppukäyttäjien osallistamisen tavoista vaatimusmäärittelytyöhön olisi ollut kiinnostavaa kuulla ajatuksia myös johdolta.

Kiinnostavia jatkotutkimuksen aiheita ovat

- Kuinka ohjelmistohankintojen ja vaatimusmäärittelytyön sisältöjen tietoisuutta voitaisiin lisätä?
- Miten monialaista henkilöstöä voitaisiin osallistaa tehokkaammin vaatimusmäärittelytyöhön?
- Millainen ohjelmistoprojektin prosessimalli tukisi loppukäyttäjien osallistamista vaatimusmäärittelytyöhön?

Tulokset tuottivat tietoa Lapin AMKin henkilöstön suhtautumisesta vaatimusmäärittelytyöhön. Tuloksia ei voi yleistää laajemmin, mutta tuloksia voidaan hyödyntää tutkimuksen suunnitteluun toisissa vastaavissa oppilaitoksissa. Tutkimuksen tuloksia on mahdollista hyödyntää jatkossa myös Lapin AMKin toimintatapojen kehittämiseen ohjelmistohankinnoissa. Tulokset antavat tietoa hyvin moninaisesti henkilöstön toiveista ohjelmistohankintoja ja vaatimusmäärittelytyötä kohtaan.

8 Yhteenveto ja loppupäätelmät

Tämä pro gradu -tutkielma sai innoituksensa kiinnostuksestamme tutkia asenteita vaatimusmäärittelytyötä kohtaan. Tutkimuksen aiheena oli selvittää, miten ohjelmistojen vaatimusmäärittelytyöhön suhtaudutaan ja millainen asenne loppukäyttäjillä on siihen osallistumista kohtaan. Tutkielmaa ohjasi siten tutkimuskysymykset: ”Miten loppukäyttäjät asennoituvat ohjelmistojen vaatimusmäärittelytyöhön?” ja ”Kuinka loppukäyttäjien osallistamista vaatimusmäärittelytyöhön voitaisiin lisätä?”.

Työn alkuosa koostuu kirjallisuuskatsauksesta. Kirjallisuuskatsauksessa esiteltiin vaatimusmäärittelytyö, syitä ohjelmistoprojektien epäonnistumiseen ja loppukäyttäjän roolia ohjelmistoprojektissa. Tutkimukseen kuului määrällisen tutkimuksen osuus, jonka avulla pyrittiin luomaan yleiskuva Lapin AMKin henkilöstön suhtautumisesta vaatimusmäärittelytyötä kohtaan. Määrällinen tutkimus toteutettiin sähköisen lomakekyselyn avulla, joka osoitettiin Lapin AMKin päätoimiselle henkilöstölle. Lomakekyselyn vastausprosentti oli kiitettävä. Joka neljäs Lapin AMKinlainen vastasi kyselyyn.

Tutkimukseen kuului myös laadullinen osuus. Laadullinen tutkimus toteutettiin kahtena erillisenä ryhmähaastatteluna tutkimukseen valituille Lapin AMKin henkilöstön jäsenille. Määrällisen lomakekyselyn tulokset toimivat pohjana laadulliselle haastattelututkimukselle, jonka vastausten avulla selvitettiin tarkemmin syitä loppukäyttäjän suhtautumiseen vaatimusmäärittelytyöhön. Tulosten analysoinnissa pyrittiin jälkipositivistiseen lähestymiseen ottamalla huomioon omat puutteet havainnoinnissa ja kerätyssä aineistossa. Huomioitavaa erityisesti ryhmähaastattelujen osalta oli se, että toinen tutkimuksen tekijöistä työskenteli tutkimuksen aikana tutkittavassa organisaatiossa.

Loppukäyttäjien osallistuminen ohjelmistoprojekteihin edistää monin tavoin sekä ohjelmistoprojektin onnistumista, että loppukäyttäjien sitoutumista ohjelmiston käyttöön. Taustateoriassa esitettiin syitä, miksi loppukäyttäjä ei välttämättä aina ole motivoitunut osallistumaan vaatimusmäärittelytyöhön. Olettamuksemme Lapin AMKin henkilökunnan suhtautumisesta ohjelmistojen vaatimusmäärittelytyöhön oli taustateorian kanssa hyvin samankaltainen.

Tutkimusta suunnitellessamme ja vielä empiirisen osion aikanakin mielessämme oli ennakko-oletamus tulosta kohtaan. Epäilimme, että henkilöstön jäsenet eivät ole välttämättä tietoisia vaatimusmäärittelytyön tarkoituksesta ja sisällöstä. Arvelimme tapaustutkimuksen tuottavan tuloksen, jonka mukaan vastaajat eivät ole motivoituneita osallistumaan ohjelmistojen vaatimusmäärittelytyöhön. Teoriaosuus loppukäyttäjien motivaatioon vaikuttavista tekijöistä tuki määrällisen ja laadullisen aineistonkeruun suunnittelua ja toteutusta.

Tutkimuksen tulokset osoittautuivat päinvastaisiksi olettamuksiimme nähden. Ryhmähaastattelujen mukaan henkilöstö haluaisi osallistua vaatimusmäärittelytyöhön enemmän, jos heille tarjottaisiin siihen mahdollisuus. Lomakekyselyn tulosten mukaan vaatimusmäärittelytyöhön suhtaudutaan pääasiassa positiivisesti silloin, kun henkilöllä on aikaisempaa kokemusta vaatimusmäärittelytyöhön osallistumisesta tai tietoa vaatimusmäärittelytyön sisällöstä. Lomakekyselyn ja ryhmähaastattelujen tulosten perusteella asennetta voitaisiin parantaa edelleen perehdytysten avulla. Perehdytyksissä tulisi kertoa, että vaatimusmäärittelytyössä riittää, kun tuntee oman työprosessinsa ja osaa kuvata sen. Prosessituntemuksen lisäksi tarvitaan vain avointa mieltä ja kiinnostusta kehittää omassa työssä käytettäviä ohjelmistoja.

Vaatimusmäärittelytyöhön osallistumisen mielekkyyttä koettiin vähentävän aikaresurssin puuttuminen ja johdon tuen puute. Lisäksi negatiivisena asiana nykyhetkessä koettiin ohjelmistohankintojen suunnittelun puutteellisuus ja ohjelmistojen huono käytettävyys. Tutkimukseen osallistuneet kritisivat sitä, että loppukäyttäjät aletaan osallistaa vasta käyttöönottovaiheessa, kun uuden ohjelmiston perehdytykset järjestetään. Ohjelmistohankintoihin toivottiin selkeämpää ja yhdenmukaisempaa projektimallia ja johdon tuen näkymistä aikaresurssin myöntämisenä osallistumista varten. Ryhmähaastatteluihin osallistuneet olivat sitä mieltä, että loppukäyttäjien osallistaminen vaatimusmäärittelytyöhön maksaa itsensä takaisin ohjelmiston käyttöönottovaiheessa asiakastyytyväisyyden kautta.

Tutkimuksen toisen tutkimuskysymyksen mukaisesti ryhmähaastatteluihin osallistuneet Lapin AMKin henkilöstön jäsenet pohtivat keinoja loppukäyttäjien osallistamiseen. Henkilöstöinfot ja tiimipalaverit nähtiin tilaisuuksina, joissa henkilöstöstä voitaisiin kartoittaa mielipiteitä ja vapaaehtoisia jäseniä osallistumaan vaatimusmäärittelytyöhön. Ohjelmistohankintojen prosessimallin tai prosessikuvauksen hyödyllisyyttä mietittiin. Haastateltavat olivat yhdenmielisiä siitä, että kaikkien ohjelmistohankintojen tulisi noudattaa samaa projektimallia. Suurimpana osallistamisen lisäämisen keinona nähtiin resurssiongelman ratkaiseminen ja huomioiminen

organisaation johdon toimesta.

Tutkimuksen tuloksissa yllättävää oli se, että yleisesti tutkimukseen osallistuneiden asenne vaatimusmäärittelytyötä kohtaan oli myönteinen, vaikka suhtautuminen nykyhetken tilanteeseen oli monella osa-alueella tyytymätön. Ongelmakohtiin tarttumisella ja toimintatapojen uudistamisella valtaosa havaituista epäkohdista voitaisiin saada korjattua. Ongelmien käsitteleminen ja korjaaminen voisi edistää jatkossa loppukäyttäjien tyytyväisyyttä ohjelmistoja kohtaan ja pienentää ohjelmistoprojektien kustannuksia Lapin AMKissa.

Tämä tutkimus tuotti kiinnostavan kuvan Lapin AMKin henkilöstön asennoitumisesta ohjelmistojen vaatimusmäärittelytyöhön. Empiirinen osio vastasi kattavasti työn alussa määritettyihin tutkimuskysymyksiin. Kuten sanotaan, että jokaisen tutkimuksen tulisi jollakin tapaa yllättää tutkijansa, niin voimme todeta käyneen tässäkin tutkimuksessa. Vaikka tulokset olivat yllättävän positiiviset tutkimuskysymysten kannalta, olivat henkilöstön havaitsemat epäkohdat yhdenmukaisia työssä esitetyn taustateorian kanssa.

Jälkikäteen katsottuna tutkimuksen suunnittelussa olisi voinut huomioida kyselylomakkeen suunnittelun paremmin. Organisaation henkilöstörakenne vaikutti jonkin verran tuloksiin siten, että kaikkiin vastauskategorioihin ei saatu riittävää määrää vastauksia ristiintaulukointia ajatellen. Ryhmähaastatteluja tai yksilöhaastatteluja olisi voinut pitää enemmänkin kuin tutkimuksessa suoritettuihin kaksiryhmähaastatteluihin. Erityisesti johtavassa asemassa olevien henkilöiden haastattelusta olisi voitu saada kiinnostavaa vertailukohtaa muihin ryhmähaastatteluihin.

Tutkimuksen tuloksia on mahdollista hyödyntää jatkossa organisaation toimintatapojen kehittämiseen ohjelmistohankinnoissa. Tulokset antavat tietoa hyvin moninaisesti henkilöstön toiveista ohjelmistohankintoja ja vaatimusmäärittelytyötä kohtaan. Tuloksista on poimittavissa myös esiin nousseita epäkohtia vaatimusmäärittelytyöhön osallistumiseen liittyen.

Tämä tapaustutkimus ja sen tulokset sellaisenaan antavat myös muihin tutkimuksiin lisää ymmärrystä loppukäyttäjien suhtautumisesta ohjelmistojen vaatimusmäärittelytyöhön. Tutkimuksessa havaittuja haasteita ja niiden kehitysehdotuksia on mahdollista peilata toisten oppilaitosorganisaatioiden ohjelmistohankintaprosesseihin ja niiden parantamiseen.

Lähteet

- [1] *The CHAOS Report*. Tekninen raportti, The Standish Group International, 1995.
- [2] ABBAS, N., GRAVELL, A. M., JA WILLS, G. B. Historical roots of agile methods: where did “agile thinking” come from? Julkaisusarjassa *9th International Conference on Agile Processes and Extreme Programming in Software Engineering* (June 2008), Limerick, ss. 94–103.
- [3] ALLADI, A., JA VADARI, S. Systemic approach to project management: A stakeholders perspective for sustainability. Julkaisusarjassa *2011 Annual IEEE India Conference (INDICON)* (Dec. 2011), Hyderabad, ss. 1–4.
- [4] ALLY, M., DARROCH, F., JA TOLEMAN, M. *A Framework for Understanding the Factors Influencing Pair Programming Success*. Springer-Verlag, Berlin, 2005.
- [5] AURUM, A., JA WOHLIN, C. *Engineering and managing software requirements*. Springer-Verlag, Berlin, 2005.
- [6] BANO, M., JA ZOWGHI, D. A systematic review on the relationship between user involvement and system success. *Information and Software Technology* 58 (2015), 148–169.
- [7] BAROUDI, J. J., OLSON, M. H., JA IVES, B. An empirical study of the impact of user involvement on system usage and information satisfaction. *Communications of the ACM* 29, 3 (1986), 232–238.
- [8] BATOOL, A., MOTLA, Y. H., HAMID, B., ASGHAR, S., RIAZ, M., MUKHTAR, M., JA AHMED, M. Comparative study of traditional requirement engineering and agile requirement engineering. Julkaisusarjassa *15th International Conference on Advanced Communications Technology (ICACT)* (Jan. 2013), PyeongC-hang, ss. 1006–1014.
- [9] BECK, K., BEEDLE, M., VAN BENNEKUM, A., COCKBURN, A., CUNNINGHAM, W., FOWLER, M., GRENNING, J., HIGHSMITH, J., HUNT, A., JEFFRIES, R.,

- KERN, J., MARICK, B., MARTIN, R. C., MELLOR, S., SCHWABER, K., SUTHERLAND, J., JA THOMAS, D. Manifesto for agile software development, 2001. URL <http://agilemanifesto.org/iso/fi/manifesto.html>, viitattu 17.2.2017.
- [10] BEECHAM, S., HALL, T., JA RAINER, A. Software process improvement problems in twelve software companies: An empirical analysis. *Empirical software engineering* 8, 1 (2003), 7–42.
- [11] BERANDER, P., JA ANDREWS, A. Requirements prioritization. Kirjassa *Engineering and managing software requirements*, A. Aurum ja C. Wohlin, Eds. Springer-Verlag, Berlin, 2005, ss. 69–94.
- [12] BERENBACH, B., Ed. *Software & systems requirements engineering : in practice*. McGraw-Hill, New York, 2009.
- [13] BOEHM, B. W. A spiral model of software development and enhancement. *Computer* 21, 5 (May 1988), 61–72.
- [14] BROWNE, G. J., JA RAMESH, V. Improving information requirements determination: a cognitive perspective. *Information & Management* 39, 8 (2002), 625–645.
- [15] BUBENKO, J. A. Challenges in requirements engineering. Julkaisusarjassa *Proceedings of the Second IEEE International Symposium on Requirements Engineering* (Mar. 1995), York, ss. 160–162.
- [16] CANFORA, G., CIMITILE, A., JA VISAGGIO, C. A. *Empirical Study on the Productivity of the Pair Programming*. Springer-Verlag, Berlin, 2005.
- [17] CHEMUTURI, M. *Requirements Engineering and Management for Software Development Projects*. Springer-Verlag, New York, 2013.
- [18] CORBIN, J. M., JA STRAUSS, A. L. *Basics of qualitative research : techniques and procedures for developing grounded theory*, 3rd ed. Sage Publications, Los Angeles, California, 2008.
- [19] COTTERELL, M., JA HUGHES, B. *Software project management*. International Thomson Computer Press, London, 1999.
- [20] COUGHLAN, J., JA MACREDIE, R. D. Effective communication in requirements elicitation: A comparison of methodologies. *Requirements Engineering* 7, 2 (2002), 47–60.

- [21] CURTIS, B., KRASNER, H., JA ISCOE, N. A field study of the software design process for large systems. *Communications of the ACM* 31, 11 (1988), 1268–1287.
- [22] DINGER, C., JA OLSSON, T. Quality assurance in requirements engineering. Kirjassa *Engineering and managing software requirements*, A. Aurum ja C. Wohlin, Eds. Springer-Verlag, Berlin, 2005, ss. 163–185.
- [23] DOLL, W. J., JA TORKZADEH, G. A congruence construct of user involvement. *Decision Sciences* 22, 2 (Mar. 1991), 443.
- [24] DORFMAN, M., JA THAYER, R. H. *Software requirements engineering*. John Wiley & Sons, Inc., Los Alamitos, California, 2000.
- [25] EBERT, C., JA WIERINGA, R. J. Requirements engineering: Solutions and trends. Kirjassa *Engineering and managing software requirements*, A. Aurum ja C. Wohlin, Eds. Springer-Verlag, Berlin, 2005, ss. 453–476.
- [26] EISENHARDT, K. M. Building theories from case study research. *Academy of Management Review* 14, 4 (1989), 532–550.
- [27] EL EMAM, K., JA KORU, A. G. A replicated survey of it software project failures. *IEEE Software* 25, 5 (Sept. 2008), 84–90.
- [28] EL EMAM, K., JA MADHAVJI, N. H. A field study of requirements engineering practices in information systems development. Julkaisusarjassa *Proceedings of the Second IEEE International Symposium on Requirements Engineering* (Mar. 1995), York, ss. 68–80.
- [29] EL EMAM, K., QUINTIN, S., JA MADHAVJI, N. H. User participation in the requirements engineering process: An empirical study. *Requirements Engineering* 1, 1 (1996), 4–26.
- [30] ESKOLA, J., JA SUORANTA, J. *Johdatus laadulliseen tutkimukseen*. Vastapaino, Tampere, 1998.
- [31] GILLHAM, B. *Case study research methods*. Real world research. Continuum, London, 2000.
- [32] GOGUEN, J. A., JA LINDE, C. Techniques for requirements elicitation. Julkaisusarjassa [1993] *Proceedings of the IEEE International Symposium on Requirements Engineering* (Jan. 1993), San Diego, California, ss. 152–164.

- [33] GRUDIN, J. Obstacles to user involvement in software product development, with implications for cscw. *International Journal of Man-Machine Studies* 34, 3 (1991), 435–452.
- [34] GRÜNBAKER, P., JA SEYFF, N. Requirements negotiation. Kirjassa *Engineering and managing software requirements*, A. Aurum ja C. Wohlin, Eds. Springer-Verlag, Berlin, 2005, ss. 143–162.
- [35] HARRIS, M. A., JA WEISTROFFER, H. R. A new look at the relationship between user involvement in systems development and system success. *Communications of the Association for Information Systems* 24, 1 (2009), 42.
- [36] HAUGHEY, D. Smart goals, 2017. URL <https://www.projectsmart.co.uk/smart-goals.php>, viitattu 26.2.2017.
- [37] HEIKKILÄ, T. *Tilastollinen tutkimus*, 9th ed. Edita, Helsinki, 2014.
- [38] HEINBOKEL, T., SONNENTAG, S., FRESE, M., STOLTE, W., JA BRODBECK, F. C. Don't underestimate the problems of user centredness in software development projects there are many! *Behaviour & Information Technology* 15, 4 (1996), 226–236.
- [39] HOFMANN, H. F., JA LEHNER, F. Requirements engineering as a success factor in software projects. *IEEE Software* 18, 4 (July 2001), 58–66.
- [40] HULL, E. *Requirements Engineering*. Springer-Verlag, London, 2011.
- [41] IEEE STANDARD 1220-2005. *IEEE Standard for Systems engineering — Application and management of the systems engineering process*, 2005.
- [42] IEEE STANDARD 1233, 1998 EDITION. *IEEE Guide for Developing System Requirements Specifications*, 1998.
- [43] IEEE STANDARD 610.12-1990. *IEEE Standard Glossary of Software Engineering Terminology*, 1990.
- [44] IEEE STANDARD 830-1998. *IEEE Recommended Practice for Software Requirements Specifications*, 1998.
- [45] JOHNSON, J., JA CREAR, J. Success redefined, 2017. URL <http://blog.standishgroup.com/post/23>, viitattu 26.2.2017.

- [46] JÖNSSON, P., JA LINDVALL, M. Impact analysis. Kirjassa *Engineering and managing software requirements*, A. Aurum ja C. Wohlin, Eds. Springer-Verlag, Berlin, 2005, ss. 117–142.
- [47] KARLSSON, L., DAHLSTEDT, Å. G., REGNELL, B., OCH DAG, J. N., JA PERS-
SON, A. Requirements engineering challenges in market-driven software
development—an interview study with practitioners. *Information and Software
technology* 49, 6 (2007), 588–604.
- [48] KEIL, M., CULE, P. E., LYYTINEN, K., JA SCHMIDT, R. C. A framework for
identifying software project risks. *Communications of the ACM* 41, 11 (1998),
76–83.
- [49] KHAIYUM, S., KUMARASWAMY, Y. S., JA KARIBASAPPA, K. Significance of fai-
lure avoidance in software development process. Julkaisusarjassa *2014 Interna-
tional Conference on Intelligent Computing Applications* (Mar. 2014), Coimbatore,
ss. 340–344.
- [50] KOCH, A. S. *Agile software development : evaluating the methods for your organiza-
tion*. Artech House computing library. Artech House, Boston, Massachusetts,
2005.
- [51] KUJALA, S. User involvement: a review of the benefits and challenges. *Beha-
viour & information technology* 22, 1 (2003), 1–16.
- [52] KUJALA, S., KAUPPINEN, M., LEHTOLA, L., JA KOJO, T. The role of user invol-
vement in requirements quality and project success. Julkaisusarjassa *13th IEEE
International Conference on Requirements Engineering* (Aug. 2005), Paris, ss. 75–84.
- [53] LANKHORST, MARC, T., Ed. *Agile Service Development : Combining Adaptive Met-
hods and Flexible Solutions*. The Enterprise Engineering Series. Springer-Verlag,
Berlin, 2012.
- [54] LARMAN, C., JA BASILI, V. R. Iterative and incremental development: A brief
history. *Computer* 36, 6 (June 2003), 47–56.
- [55] LAWRENCE, B., WIEGERS, K., JA EBERT, C. The top risk of requirements engi-
neering. *IEEE Software* 18, 6 (Nov. 2001), 62–63.

- [56] MACHADO, R. J., RAMOS, I., JA FERNADES, J. M. Specification of requirements models. Kirjassa *Engineering and managing software requirements*, A. Aurum ja C. Wohlin, Eds. Springer-Verlag, Berlin, 2005, ss. 47–68.
- [57] MEASEY, P. *Agile Foundations : principles, practices and frameworks*. BCS, Wiltshire, England, 2015.
- [58] METSÄMUURONEN, J. *Tutkimuksen tekemisen perusteet ihmistieteissä : e-kirja opiskelijalaitos*. International Methelp, Booky.fi, Helsinki, 2011.
- [59] MOHAPATRA, P. K. J. *Software engineering : (a lifecycle approach)*. New Age International, New Delhi, 2010.
- [60] MORRIS, P. W., ET AL. Managing project interfaces: key points for project success. *Project management handbook 2* (1983), 16–55.
- [61] NASIR, M. H. N., JA SAHIBUDDIN, S. Critical success factors for software projects: A comparative study. *Scientific Research and Essays* 6, 10 (2011), 2174–2186.
- [62] NUSEIBEH, B., JA EASTERBROOK, S. Requirements engineering: a roadmap. Julkaisusarjassa *Proceedings of the Conference on the Future of Software Engineering* (2000), Limerick, ss. 35–46.
- [63] PAETSCH, F., EBERLEIN, A., JA MAURER, F. Requirements engineering and agile software development. Julkaisusarjassa *Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises* (June 2003), Linz, ss. 308–313.
- [64] PATTON, M. Q. *Qualitative research and evaluation methods*, 3rd ed. Sage Publications, Thousand Oaks, California, 1990.
- [65] PINTO, J. K., JA SLEVIN, D. P. Critical factors in successful project implementation. *IEEE Transactions on Engineering Management* EM-34, 1 (Feb. 1987), 22–27.
- [66] POHL, K., JA RUPP, C. *Requirements engineering fundamentals : a study guide for the Certified Professional for Requirements Engineering exam : foundation level, IREB compliant*. Rocky Nook, Santa Barbara, California, 2011.

- [67] RACHEVA, Z., DANEVA, M., SIKKEL, K., HERRMANN, A., JA WIERINGA, R. Do we know enough about requirements prioritization in agile projects: Insights from a case study. *Julkaisusarjassa 18th IEEE International Requirements Engineering Conference* (Sept. 2010), Sydney, ss. 147–156.
- [68] REGNELL, B., HÖST, M., OCH DAG, J. N., BEREMARK, P., JA HJELM, T. An industrial case study on distributed prioritisation in market-driven requirements engineering for packaged software. *Requirements Engineering* 6, 1 (2001), 51–62.
- [69] ROBSON, C. *Real world research: a resource for users of social research methods in applied settings*, 3rd ed. Wiley-Blackwell, Chichester, 2011.
- [70] ROYCE, W. W., ET AL. Managing the development of large software systems. *Julkaisusarjassa proceedings of IEEE WESCON* (1970), vol. 26, Los Angeles, California, ss. 1–9.
- [71] SADDINGTON, P. *Agile pocket guide : a quick start to making your business agile using Scrum and beyond*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2013.
- [72] SCHARER, L. Pinpointing requirements. *Kirjassa Software requirements engineering*, M. Dorfman ja R. H. Thayer, Eds. John Wiley & Sons, Inc., Los Alamitos, California, 2000, ss. 30–35.
- [73] SCHWABER, K., JA SUTHERLAND, J. V. *Software in 30 days : how Agile managers beat the odds, delight their customers, and leave competitors in the dust*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2012.
- [74] SHARP, H., FINKELSTEIN, A., JA GALAL, G. Stakeholder identification in the requirements engineering process. *Julkaisusarjassa Tenth International Workshop on Database and Expert Systems Applications. DEXA 99* (1999), Florence, ss. 387–391.
- [75] SILLITTI, A., JA SUCCI, G. Requirements engineering for agile methods. *Kirjassa Engineering and managing software requirements*, A. Aurum ja C. Wohlin, Eds. Springer-Verlag, Berlin, 2005, ss. 309–326.
- [76] SILVERMAN, D. *Doing qualitative research : a practical handbook*. Sage Publications, London, 2000.

- [77] SIOUKAS, A. V. User involvement for effective customization: An empirical study on voice networks. *IEEE Transactions on Engineering Management* 42, 1 (1995), 39–49.
- [78] SOMMERVILLE, I. Integrated requirements engineering: a tutorial. *IEEE Software* 22, 1 (Jan. 2005), 16–23.
- [79] SOMMERVILLE, I., JA SAWYER, P. Viewpoints: principles, problems and a practical approach to requirements engineering. *Annals of Software Engineering* 3, 1 (1997), 101–130.
- [80] SOMMERVILLE, I., JA SAWYER, P. *Requirements engineering: a good practice guide*. John Wiley & Sons, Inc., New York, 2000.
- [81] STEPHENS, R. *Beginning software engineering*. Wrox, Indianapolis, Indiana, 2015.
- [82] TARAWNEH, M. M. I., AL-TARAWNEH, H., JA ELSHEIKH, A. Software development projects: An investigation into the factors that affect software project success/ failure in jordanian firms. *Julkaisusarjassa 2008 First International Conference on the Applications of Digital Information and Web Technologies (ICADIWT)* (Aug. 2008), Ostrava, ss. 246–251.
- [83] TUOMI, J., JA SARAJÄRVI, A. *Laadullinen tutkimus ja sisällönanalyysi*, 8th ed. Tammi, Helsinki, 2009.
- [84] VISKOVIC, D., VARGA, M., JA CURKO, K. Bad practices in complex it projects. *Julkaisusarjassa ITI 2008 - 30th International Conference on Information Technology Interfaces* (June 2008), Dubrovnik, ss. 301–306.
- [85] WIEGERS, K. Karl wiegers describes 10 requirements traps to avoid. *Software Testing and Quality engineering* 2, 1 (2000), 34–41.
- [86] WILSON, S., BEKKER, M., JOHNSON, P., JA JOHNSON, H. Helping and hindering user involvement - a tale of everyday design. *Julkaisusarjassa Proceedings of the ACM SIGCHI Conference on Human factors in computing systems* (1997), Atlanta, Georgia, ss. 178–185.
- [87] YIN, R. K. *Case study research : design and methods*, 3rd ed. Applied social research methods series. Sage Publications, Thousand Oaks, California, 2003.

- [88] YOUNG, R. R. *The requirements engineering handbook*. Artech House technology management and professional development library. Artech House, Boston, Massachusetts, 2004.
- [89] YU, E. S. K. Towards modelling and reasoning support for early-phase requirements engineering. Julkaisusarjassa *Proceedings of the Third IEEE International Symposium on Requirements Engineering* (Jan. 1997), Annapolis, Maryland, ss. 226–235.
- [90] ZHU, J., LIANG, Y., JA GU, Y. The requirements change analysis for different level users. Julkaisusarjassa *2008 International Symposium on Intelligent Information Technology Application Workshops* (Dec. 2008), Shanghai, ss. 987–989.
- [91] ZOWGHI, D., JA COULIN, C. Requirements elicitation: A survey of techniques, approaches, and tools. Kirjassa *Engineering and managing software requirements*, A. Aurum ja C. Wohlin, Eds. Springer-Verlag, Berlin, 2005, ss. 19–46.

A Sähköinen lomakekysely

Hei!

Olemme tekemässä ohjelmistotekniikan maisteriopintoihin (Jyväskylän yliopisto) liittyvää pro gradu –tutkimusta, jonka tavoitteena on selvittää suhtautumista ohjelmistojen vaatimusmäärittelyyn.

Ohjelmistojen hankintaan sisältyy olennaisesti käyttäjiltä kerättävät vaatimukset, jotka voivat liittyä esimerkiksi ohjelmiston käytettävyyteen, toimintoihin tai ulkoasuun. Vaatimusmäärittelyn tavoitteena on siis saada tietoon kaikki tarpeet ja vaatimukset, jotka ohjelmiston tulisi toteuttaa.

Toivomme, että voisit käyttää pari minuuttia kysymyksiin vastaamiseen ja edesauttaa näin arvokkaan tutkimusaineiston kerryttämistä. Vastauksesi on erityisen tärkeä, vaikka et olisi koskaan osallistunutkaan vaatimusmäärittelyyn! Vastaamalla voit vaikuttaa myös jatkossa Lapin AMKissa toteutettavien ohjelmistohankintojen onnistumiseen.

Pääset vastaamaan kyselyyn tämän sähköpostin alaosasta löytyvän Webropol-linkin kautta. Vastaukset käsitellään luottamuksellisesti, vastauksia ei käytetä henkilöiden yksilöintiin tai tunnistamiseen.

Pyydämme vastauksia pikaisesti, kuitenkin viimeistään perjantaihin 17.3. mennessä.

Kiitos jo etukäteen yhteistyöstä!

Terveisin,
Laura Kinnunen ja
Petri Niemi

Kyselytutkimus asenteista ohjelmiston vaatimusmäärittelyä kohtaan

Kyselyn tavoitteena on selvittää monialaisten henkilöiden suhtautumista ohjelmistoprojekteihin ja erityisesti ohjelmistojen vaatimusmäärittelyyn. Ohjelmistojen hankintaan sisältyy olennaisesti käyttäjiltä kerättävät vaatimukset, jotka voivat liittyä esimerkiksi ohjelmiston käytettävyyteen, toimintoihin tai ulkoasuun. Vaatimusten keräämistä kutsutaan vaatimusmäärittelyksi. Vaatimusmäärittelyn tavoitteena on kerätä käyttäjien vaatimukset tulevaa ohjelmistoa varten ja siten helpottaa käyttäjän työtehtävien suorittamista ohjelmiston avulla.

Tämä kysely on ensimmäinen osa pro gradu -tutkielman empiirisestä osiosta. Seuraavassa vaiheessa tutkittavaan ilmiöön perehdytään haastattelujen avulla. Tämän lomakkeen täyttäminen ei sido sinua tutkimuksen seuraavaan vaiheeseen. Vastaamalla kyselyyn autat kerryttämään arvokasta tietoa vaatimusmäärittelyyn suhtautumisesta. Suuret kiitokset ajastasi!

Kysely toteutetaan Lapin ammattikorkeakoulun henkilöstölle. Taustatietoja ei käytetä henkilöiden yksilöintiin tai tunnistamiseen.

1. Ikä *

2. Mikä on korkein suoritettu tutkintosi *

3. Koulutusalasi *

4. Millaiseksi arvioit omat tietokoneen käyttötaitosi? *

- Erinomainen
 Hyvä
 Keskinertainen
 Välttävä

5. Tiedätkö mitä vaatimusmäärittely tarkoittaa? *

- Kyllä (Kirjoita lyhyesti oma käsityksesi/määritelmä vaatimusmäärittelystä)
 En
 En osaa sanoa

6. Oletko osallistunut aikaisemmin vaatimusmäärittelyyn? *

- Kyllä
 En
 En osaa sanoa

7. Tunnetko olevasi pätevä osallistumaan vaatimusmäärittelyyn? *

- Kyllä
 En
 En osaa sanoa

8. Koetko että sinulla olisi annettavaa vaatimusmäärittelyissä? *

- Erittäin paljon
- Melko paljon
- Jonkin verran
- Vähän
- Ei lainkaan

9. Koetko että sinulla on työnantajan tuki vaatimusmäärittelyyn osallistumiseen? *

- Kyllä
- En
- En osaa sanoa

10. Koetko vaatimusmäärittelyyn osallistumisen olevan vaivalloista? *

- Kyllä
- En
- En osaa sanoa

11. Missä rooleissa olet osallistunut vaatimusmäärittelyihin? *

- Tulevan ohjelmiston käyttäjänä
- En ole koskaan osallistunut
- Muu rooli, mikä?

12. Koetko että vaatimusmäärittelyyn osallistuminen lisää mahdollisuuksiasi vaikuttaa kehitettävän ohjelmiston ominaisuuksiin? *

- Kyllä
- En
- En osaa sanoa

13. Oletko kiinnostunut osallistumaan jatkossa vaatimusmäärittelyihin? *

- Kyllä
- Ei
- En osaa sanoa

14. Jos vastasit kysymykseen 13. kieltävästi, niin mitä pitäisi muuttaa, jotta vaatimusmäärittelyyn osallistuminen olisi mielekkäämpää ja kokisit osallistumisen tärkeäksi?

15. Tutkimuksemme jakautuu kahteen osaan: kyselytutkimukseen ja haastattelututkimukseen. Molemmat osat ja vastauksesi ovat tärkeitä pro gradu –tutkielmamme onnistumisen kannalta. Toivoisimme, että voisit osallistua myös haastatteluun. Jos haluat ilmoittautua vapaaehtoiseksi haastattelututkimukseen, voit ilmoittaa työsähköpostiosoitteesi tässä.

Sähköposti

B Ryhmähaastattelut

Kutsu haastatteluryhmälle 1.

Hei!

Kiitos vastauksestasi pro gradu –tutkielmaan liittyen viime viikolla. Vastausprosentti oli erinomaisen korkea, kaikkiaan joka neljäs Lapin AMKilainen vastasi kyselyyn! Kyselyssä kartoitettiin myös vapaaehtoisia tutkimuksen toiseen osaan, haastattelututkimukseen.

Olemme käyneet läpi vastaukset ja haluaisimme pyytää sinut mukaan haastatteluun. Kyseessä on lyhyt ryhmähaastattelu, johon kutsutaan kolme henkilökunnan jäsentä jotka ovat joskus osallistuneet vaatimusmäärittelyyn. Aikaisemmat kokemuksesi vaatimusmäärittelyistä tuottaisivat arvokasta aineistoa tutkimukseemme.

Alustavien aikataulusovittelujen mukaan haastattelu voitaisiin järjestää torstaina 23.3. klo. 10.00. Aikaa keskustelua varten on tarpeen varata korkeintaan tunti. Olisiko sinulla mahdollisuus osallistua ja sopiiko aikatauluehdotus?

Terveisin

Laura Kinnunen ja

Petri Niemi

Kutsu haastatteluryhmälle 2.

Hei!

Kiitos vastauksestasi pro gradu –tutkielmaan liittyen viime viikolla. Vastausprosentti oli erinomaisen korkea, kaikkiaan joka neljäs Lapin AMKilainen vastasi kyselyyn! Kyselyssä kartoitettiin myös vapaaehtoisia tutkimuksen toiseen osaan, haastattelututkimukseen.

Olemme käyneet läpi vastaukset ja haluaisimme pyytää sinut mukaan haastatteluun. Kyseessä on lyhyt ryhmähaastattelu, johon kutsutaan kolme henkilökunnan jäsentä, joilla ei ole aikaisempaa kokemusta vaatimusmäärittelyyn osallistumisesta. Ajatuksesi vaatimusmäärittelystä tuottaisivat arvokasta aineistoa tutkimukseemme.

Alustavien aikataulusovittelujen mukaan haastattelu voitaisiin järjestää keskiviikkona 22.3. klo. 14.30. Aikaa keskustelua varten on tarpeen varata korkeintaan tunti. Olisiko sinulla mahdollisuus osallistua ja sopiiko aikatauluehdotus?

Terveisin

Laura Kinnunen ja

Petri Niemi

Aihealueet haastatteluryhmälle 1.

- Aikaisemmat kokemukset vaatimusmäärittelystä
- Motivaatio ohjelmistohankintaan osallistumiseen
- Kuvailu hyvästä vaatimusmäärittelystä

Aihealueet haastatteluryhmälle 2.

- Ajatuksia vaatimusmäärittelystä ja sen sisällöstä
- Ennakkoasenteet vaatimusmäärittelytyötä kohtaan
- Kokemukset mahdollisuudesta osallistua vaatimusmäärittelyyn

Taulukko B.1: Haastatteluryhmä 1.

	Haastateltava 1.	Haastateltava 2.	Haastateltava 3.
Ikä	40-49	50-59	60-
Tutkinto	Toinen aste	Ylempi korkeakouluaste	Alempi korkeakouluaste
Koulutusala	Yhteiskuntatieteiden, liiketalouden ja hallinnon ala	Luonnontieteiden ala	Tekniikan ja liikenteen ala
Tietokoneen käyttötaidot	Erinomainen	Erinomainen	Keskinkertainen
Tunnetko olevasi pätevä osallistumaan vaatimusmäärittelyyn	Kyllä	Kyllä	En
Koetko että sinulla olisi annettavaa vaatimusmäärittelyssä	Erittäin paljon	Melko paljon	Jonkin verran
Koetko että sinulla on työnantajan tuki vaatimusmäärittelyyn osallistumiseen	Kyllä	En	Kyllä
Koetko vaatimusmäärittelyyn osallistumisen olevan vaivalloista	Kyllä	En	En
Missä rooleissa olet osallistunut olet osallistunut vaatimusmäärittelyyn	Tulevan ohjelmiston käyttäjänä, kehittäjänä, teknisenä pääkäyttäjänä, sisällöllisenä asiantuntijana	Tulevan ohjelmiston käyttäjänä, ohjelmiston suunnittelijana	Tulevan ohjelmiston käyttäjänä
Koetko että osallistuminen lisää mahdollisuuksiasi vaikuttaa kehitettävän ohjelmiston ominaisuuksiin	Kyllä	Kyllä	Kyllä
Oletko kiinnostunut osallistumaan jatkossa vaatimusmäärittelyihin	Kyllä	Kyllä	Kyllä

Taulukko B.2: Haastatteluryhmä 2.

	Haastateltava 1.	Haastateltava 2.	Haastateltava 3.
Ikä	30-39	40-49	60-
Tutkinto	Alempi korkeakouluaste	Ylempi korkeakouluaste	Ylempi korkeakouluaste
Koulutusala	Sosiaali-, terveys ja liikunta-ala	Tekniikan ja liikenteen ala	Humanistinen ja kasvatustieteet
Tietokoneen käyttötaidot	Hyvä	Hyvä	Hyvä
Tunnetko olevasi pätevä osallistumaan vaatimusmäärittelyyn	Kyllä	En osaa sanoa	En
Koetko että sinulla olisi annettavaa vaatimusmäärittelyssä	Melko paljon	Vähän	Erittäin paljon
Koetko että sinulla on työnantajan tuki vaatimusmäärittelyyn osallistumiseen	En osaa sanoa	En osaa sanoa	En
Koetko vaatimusmäärittelyyn osallistumisen olevan vaivalloista	En osaa sanoa	En osaa sanoa	En
Koetko että osallistuminen lisää mahdollisuuksiasi vaikuttaa kehittävän ohjelmiston ominaisuuksiin	Kyllä	Kyllä	Kyllä
Oletko kiinnostunut osallistumaan jatkossa vaatimusmäärittelyihin	Kyllä	Kyllä	Kyllä

Taulukko B.3: Esimerkki teemoittelusta

Haastattelu	Raakadata	Ydinsisältö	Tärkein tekijä	Teema
H1	Kun prosessi käynnistetään, siihen pitäisi alusta lähtien osallistaa.	Osallistaminen, vaatimusmäärittely	Osallistaminen	Loppukäyttäjien osallistaminen vaatimusmäärittelytyöhön
H1	Osallistuminen täytyisi perustua vapaaehtoisuuteen. Voisi kysyä että ketkä ovat vapaaehtoisia ja sitten ottaa sieiltä pienen porukan ja selvittää onko mahdollisuuksia osallistua.	Vaatimusmäärittely, motivaatio, resurssi		
H2	Jos saisi vaikuttaa jo ennen järjestelmän hankintaa niin käytettävyydestä tulisi selkeämpi.	Vaatimusmäärittely, osallistaminen, käytettävyys		
H2	Ehkä se helpottaisi jos olisi alusta asti mukana	Vaatimusmäärittely, osallistaminen, sitoutuminen		
H2	Ehdottomasti henkilö-kuntaa pitäisi osallistaa enemmän	Osallistaminen, vaatimusmäärittely, henkilökunta		
H1	Kun asia on auki ja jollakin on jotakin annettavaa niin voisi kertoa.	Palaute, vaatimusmäärittely		

C Sähköisen lomakekyselyn tuloksia ja vastausten ristiintaulukoiteja

Vastausten jakaumia

Taulukko C.1: Ikä

<30	3	2,7%
30-39	23	20,7%
40-49	28	25,2%
50-59	42	37,8%
60-	15	13,5%
Kaikki	111	100%

Taulukko C.2: Koulutusaste

Toinen aste (ylioppilastutkinnot, 1-3 -vuotiset ammatilliset tutkinnot ja ammatilliset perustutkinnot)	2	1,8%
Opistoaste (esim. teknikon, merkonomin tutkinnot, jotka eivät ole ammattikorkeakoulututkintoja)	10	9,0%
Alempi korkeakouluaste ja ammatillinen korkea-aste (ammattikorkeakoulututkinto, alempi korkeakoulututkinto)	30	27,0%
Ylempi korkeakouluaste (ylempi ammattikorkeakoulututkinto, ylempi korkeakoulututkinto, lääkärien erikoistumiskoulutus)	62	55,9%
Tutkijakoulutusaste (lisansiaatin ja tohtorin tutkinnot)	7	6,3%
Kaikki	111	100%

Taulukko C.3: Tietokoneen käyttötaito

Erinomainen	35	31,5%
Hyvä	61	55,0%
Keskinkertainen	14	12,6%
Välttävä	1	0,9%
Kaikki	111	100%

Taulukko C.4: Kokeeko henkilö, että hänellä on organisaation tuki osallistua

Kyllä	40	36%
Ei	9	8,1%
Ei osaa sanoa	62	55,9%
Kaikki	111	100%

Ristiintaulukoinnin yhteenveto

Taulukko C.5: Kyselyn tulosten yhteenveto (p -arvot)

	Ikä	Oma arvio tietokoneen käyttötaidosta	Koulutusaste	Koulutusala	Oma arvio siitä, ymmärtääkö mitä vaatimusmäärittelytyö on	Aiempi osallistuminen vaatimusmäärittelytyöhön
Kokeeko olevansa pätevä vaatimusmäärittelytyöhön	0,8035	-	0,6383	<u>0,02337</u>	<u>0,004</u>	<u>0,001</u>
Kokeeko että on annettavaa vaatimusmäärittelytyöhön	0,4184	-	0,2287	0,1233	<u>0,000316</u>	<u>0,0001164</u>
Kokeeko voivansa vaikuttaa vaatimusmäärittelytyön avulla	0,5209	-	0,7579	0,3067	<u>0,0001811</u>	<u>0,003787</u>
Kokeeko vaatimusmäärittelytyön vaivalloiseksi	0,4549	-	0,569	0,4863	<u>0,000</u>	<u>0,000</u>
Kokeeko että oma organisaatio tukee osallistumista vaatimusmäärittelytyöhön	0,06399	-	0,4522	0,5566	<u>0,043</u>	<u>0,025</u>
Haluaako osallistua jatkossa vaatimusmäärittelytyöhön	0,2719	-	0,1079	0,436	<u>0,014</u>	0,0661
Taulukossa on alleviivattu p -arvot, jotka ovat pienempiä kuin 5%, ja ovat siten tässä tutkimuksessa merkitseviä						

Taulukossa C.5 esitetään yhteenveto selittevien ja selitettävien tekijöiden ristiintaulukoinnilla saaduista p -arvoista. Taulukossa on alleviivattu tilastollisesti merkitsevät riippuvuudet.

Ikä selittävänä tekijänä

Taulukon C.1 mukaisia ikäryhmiä jouduttiin analyysissä yhdistelemään, jotta ristiintaulukoinnissa saatiin nollahypoteesiin riittävän suuria frekvenssejä. Lisäksi kyselyn vastausvaihtoehtojen epätarkkuuden vuoksi 30-vuotiailla on voinut olla ongelmia sijoittaa itsensä vaihtoehtojen <30 ja 30-39 välillä. Uusiksi ryhmiksi muodostettiin -39, 40-49 ja 50-, jolloin nollahypoteesiin saatiin riittävän suuret frekvenssit. Samalla yhdistyivät ryhmät, joiden välillä 30-vuotiailla saattoi olla hankaluuksia valita oikea kategoria.

Taulukko C.6: Onko henkilö osallistunut vaatimusmäärittelytyöhön * Henkilön ikä

		Ikä			
		-39	40-49	50-	Kaikki
Onko osallistunut aiemmin	Kyllä	10	20	32	62
	Ei	16	8	25	49
	Kaikki	26	28	57	111
$\chi^2_{5,95}; df = 2; p = 0,051$					
Odotetut frekvenssit: pienin 11,5; alle 5 suuruisia 0,0%					

Taulukko C.7: Kokeeko henkilö olevansa pätevä vaatimusmäärittelytyöhön * Henkilön ikä

		Ikä			
		-39	40-49	50-	Kaikki
Kokeeko olevansa pätevä vaatimusmäärittelytyöhön	Kyllä	13	17	28	58
	Ei	3	2	9	14
	Ei osaa sanoa	10	9	20	39
	Kaikki	26	28	57	111
Fischerin tarkka testi $p = 0,8035$					

Taulukko C.8: Kokeeko henkilö, että hänellä on annettavaa vaatimusmäärittelytyöhön * Henkilön ikä

		Ikä			
		-39	40-49	50-	Kaikki
Kokeeko että on annettavaa vaatimusmäärittelytyöhön	Erittäin paljon	5	3	8	16
	Melko paljon	3	11	19	33
	Jonkin verran	14	12	24	50
	Vähän	2	2	4	8
	Ei lainkaan	2	0	2	4
Kaikki		26	28	57	111
Fischerin tarkka testi $p = 0,4184$					

Taulukko C.9: Kokeeko henkilö voivansa vaikuttaa vaatimusmäärittelyn avulla * Henkilön ikä

		Ikä			
		-39	40-49	50-	Kaikki
Kokeeko voivansa vaikuttaa vaatimusmäärittelytyön avulla	Kyllä	22	24	49	95
	Ei	0	2	1	3
	Ei osaa sanoa	4	2	7	13
	Kaikki	26	28	57	111
Fischerin tarkka testi $p = 0,5209$					

Taulukko C.10: Kokeeko henkilö vaatimusmäärittelyn vaivalloiseksi * Henkilön ikä

		Ikä			
		-39	40-49	50-	Kaikki
Kokeeko vaatimusmäärittelytyön vaivalloiseksi	Kyllä	6	4	5	15
	Ei	10	12	30	52
	Ei osaa sanoa	10	12	22	44
	Kaikki	26	28	57	111
Fischerin tarkka testi $p = 0,4549$					

Taulukko C.11: Kokeeko henkilö, että hänellä on organisaation tuki osallistua vaatimusmäärittelytyöhön * Henkilön ikä

		Ikä			Kaikki
		-39	40-49	50-	
Kokeeko että oma organisaation tukee osallistumista vaatimusmäärittelytyöhön	Kyllä	7	8	25	40
	Ei	2	0	7	9
	Ei osaa sanoa	17	20	25	62
Kaikki		26	28	57	111
Fischerin tarkka testi $p = 0,06399$					

Taulukko C.12: Haluaako henkilö osallistua jatkossa * Henkilön ikä

		Ikä			Kaikki
		-39	40-49	50-	
Haluaako osallistua jatkossa vaatimusmäärittelytyöhön	Kyllä	18	18	33	69
	Ei	3	0	4	7
	Ei osaa sanoa	5	10	20	35
Kaikki		26	28	57	111
Fischerin tarkka testi $p = 0,2719$					

Koulutusaste selittävänä tekijänä

Taulukon C.2 mukainen jako viiteen ryhmään koulutusasteen mukaan oli ongelmallinen, sillä frekvenssit jäivät hyvin mataliksi toiseen asteeseen ja tutkijakoulutusasteeseen. Analyysissa käytettiin Fischerin täsmällistä testiä, jolla voidaan laskea riippuvuuksia, vaikka taulukossa olisikin pieniä frekvenssejä.

Taulukko C.13: Kokeeko henkilö olevansa pätevä vaatimusmäärittelytyöhön * Henkilön koulutusaste

		Koulutusaste					Kaikki
		Toinen aste (ylioppilastutkinnot, 1-3 -vuotiset ammatilliset tutkinnot ja ammatilliset perustutkinnot)	Opistoaste (esim. teknikon, merkonomien tutkinnot, jotka eivät ole ammattikorkeakoulututkintoja)	Alempi korkeakouluaste ja ammatillinen korkea-aste (ammattikorkeakoulututkinto, alempi korkeakoulututkinto)	Ylempi korkeakouluaste (ylempi ammattikorkeakoulututkinto, ylempi korkeakoulututkinto, lääkärin erikoistumiskoulutus)	Tutkijakoulutusaste (lisansiaatin ja tohtorin tutkinnot)	
Kokeeko olevansa pätevä vaatimusmäärittelytyöhön	Kyllä	1	4	15	34	4	58
	Ei	0	1	2	11	0	14
	Ei osaa sanoa	1	5	13	17	3	39
	Kaikki	2	10	30	62	7	111
Fischerin tarkka testi $p = 0,6383$							

Taulukko C.14: Kokeeko henkilö, että hänellä on annettavaa vaatimusmäärittelytyöhön * Henkilön koulutusaste

		Koulutusaste					Kaikki
		Toinen aste (ylioppilastutkinnot, 1-3 -vuotiset ammatilliset tutkinnot ja ammatilliset perustutkinnot)	Opistoaste (esim. teknikon, merkonomin tutkinnot, jotka eivät ole ammattikorkeakoulututkintoja)	Alempi korkeakouluaste ja ammatillinen korkea-aste (ammattikorkeakoulututkinto, alempi korkeakoulututkinto)	Ylempi korkeakouluaste (ylempi ammattikorkeakoulututkinto, ylempi korkeakoulututkinto, lääkärin erikoistumiskoulutus)	Tutkijakoulutusaste (lensiaatin ja tohtorin tutkinnot)	
Kokeeko että on annettavaa vaatimusmäärittelytyöhön	Erittäin paljon	1	0	5	10	0	16
	Melko paljon	0	2	7	19	5	33
	Jonkin verran	0	7	16	25	2	50
	Vähän	1	1	2	4	0	8
	Ei lainkaan	0	0	0	4	0	4
	Kaikki	2	10	30	62	7	111
Fischerin tarkka testi $p = 0,2287$							

Taulukko C.15: Kokeeko henkilö voivansa vaikuttaa vaatimusmäärittelyn avulla *
Henkilön koulutusaste

		Koulutusaste					
		Toinen aste (ylioppilastutkinnot, 1-3 -vuotiset ammatilliset tutkinnot ja ammatilliset perustutkinnot)	Opistoaste (esim. teknikon, merkonomin tutkinnot, jotka eivät ole ammattikorkeakoulututkintoja)	Alempi korkeakouluaste ja ammatillinen korkea-aste (ammattikorkeakoulututkinto, alempi korkeakoulututkinto)	Ylempi korkeakouluaste (ylempi ammattikorkeakoulututkinto, ylempi korkeakoulututkinto, lääkärin erikoistumiskoulutus)	Tutkijakoulutusaste (lisansiaatin ja tohtorin tutkinnot)	Kaikki
Kokeeko voivansa vaikuttaa vaatimusmäärittelytyön avulla	Kyllä	2	8	24	54	7	95
	Ei	0	0	2	1	0	3
	Ei osaa sanoa	0	2	4	7	0	13
	Kaikki	2	10	30	62	7	111
Fischerin tarkka testi $p = 0,7579$							

Taulukko C.16: Kokeeko henkilö vaatimusmäärittelyn vaivalloiseksi * Henkilön koulutusaste

		Koulutusaste					
		Toinen aste (ylioppilastutkinnot, 1-3 -vuotiset ammatilliset tutkinnot ja ammatilliset perustutkinnot)	Opistoaste (esim. teknikon, merkonomin tutkinnot, jotka eivät ole ammattikorkeakoulututkintoja)	Alempi korkeakouluaste ja ammatillinen korkea-aste (ammattikorkeakoulututkinto, alempi korkeakoulututkinto)	Ylempi korkeakouluaste (ylempi ammattikorkeakoulututkinto, ylempi korkeakoulututkinto, lääkärin erikoistumiskoulutus)	Tutkijakoulutusaste (lisansiaatin ja tohtorin tutkinnot)	Kaikki
Kokeeko vaatimusmäärittelytyön vaivalloiseksi	Kyllä	1	1	4	9	0	15
	Ei	0	7	12	29	4	52
	Ei osaa sanoa	1	2	14	24	3	44
	Kaikki	2	10	30	62	7	111
Fischerin tarkka testi $p = 0,569$							

Taulukko C.17: Kokeeko henkilö, että hänellä on organisaation tuki osallistua vaatimusmäärittelytyöhön * Henkilön koulutusaste

		Koulutusaste						
		Toinen aste (ylioppilastutkinnot, 1-3-vuotiset ammatilliset tutkinnot ja ammatilliset perustutkinnot)	Opistoaste (esim. teknikon, merkonomien tutkinnot, jotka eivät ole ammattikorkeakoulututkintoja)	Alempi korkeakouluaste ja ammatillinen korkea-aste (ammattikorkeakoulututkinto, alempi korkeakoulututkinto)	Ylempi korkeakouluaste (ylempi ammattikorkeakoulututkinto, ylempi korkeakoulututkinto, lääkärin erikoistumiskoulutus)	Tutkijakoulutusaste (lisensiaatin ja tohtorin tutkinnot)	Kaikki	
Kokeeko että oma organisaation tukee osallistumista vaatimusmäärittelytyöhön	Kyllä	2	5	12	18	3	40	
	Ei	0	1	1	7	0	9	
	Ei osaa sanoa	0	4	17	37	4	62	
	Kaikki	2	10	30	62	7	111	
Fischerin tarkka testi $p = 0,4522$								

Taulukko C.18: Haluaako henkilö osallistua jatkossa * Henkilön koulutusaste

		Koulutusaste					Kaikki
		Toinen aste (ylioppilastutkinnot, 1-3 -vuotiset ammatilliset tutkinnot ja ammatilliset perustutkinnot)	Opistoaste (esim. teknikon, merkonomin tutkinnot, jotka eivät ole ammattikorkeakoulututkintoja)	Alempi korkeakouluaste ja ammatillinen korkea-aste (ammattikorkeakoulututkinto, alempi korkeakoulututkinto)	Ylempi korkeakouluaste (ylempi ammattikorkeakoulututkinto, ylempi korkeakoulututkinto, lääkärin erikoistumiskoulutus)	Tutkijakoulutusaste (lisansiaatin ja tohtorin tutkinnot)	
Haluaako osallistua jatkossa vaatimusmäärittelytyöhön	Kyllä	1	5	18	43	2	69
	Ei	1	0	2	4	0	7
	Ei osaa sanoa	0	5	10	15	5	45
	Kaikki	2	10	30	62	7	111
Fischerin tarkka testi $p = 0,1079$							

Koulutusala selittävänä tekijänä

Kyselyn vastauksista oli selvät piikit yhteiskuntatieteiden, liiketalouden ja hallinnon koulutusosalalla sekä tekniikan ja liikenteen koulutusosalalla, jotka kattoivat vastauksista yhteensä yli 60%. Koulutusaloja ei voinut järkevästi yhdistää analyysiä varten. Yritimme myös tehdä ristiintaulukointia karsien pois kaikki muut paitsi nämä kaksi yleisintä vastausta sisältävät vastaukset, ja ristiintaulukoida saatua osajoukkoa, mutta tämäkään ei tuonut hyödynnettävissä olevia tuloksia. Tässä yhteydessä hylkäsimme χ^2 testin, ja käytimme sen sijaan Fischerin tarkkaa testiä.

Taulukko C.19: Kokeeko henkilö olevansa pätevä vaatimusmäärittelytyöhön * Henkilön koulutusala

		Koulutusala									
		Humanistinen ja kasvatustieteiden ala	Kulttuuriala	Yhteiskuntatieteiden, liiketalouden ja hallinnon ala	Luonnontieteiden ala	Tekniikan ja liikenteen ala	Luonnonvara- ja ympäristöala	Sosiaali-, terveys- ja liikunta-ala	Matkailu-, ravitsemis- ja talousala	Muu koulutus	Kaikki
Kokeeko olevansa pätevä vaatimusmäärittelytyöhön	Kyllä	4	2	2	7	15	0	2	0	2	58
	Ei	2	1	4	0	4	2	1	0	0	14
	Ei osaa sanoa	1	0	10	2	12	4	5	3	2	39
	Kaikki	7	3	40	9	31	6	8	3	4	111
Fischerin tarkka testi $p = 0,02337$											

Taulukko C.20: Kokeeko henkilö, että hänellä on annettavaa vaatimusmäärittelytyöhön * Henkilön koulutusala

		Koulutusala									
		Humanistinen ja kasvatustieteiden ala	Kulttuuriala	Yhteiskuntatieteiden, liiketalouden ja hallinnon ala	Luonnontieteiden ala	Tekniikan ja liikenteen ala	Luonnonvara- ja ympäristöala	Sosiaali-, terveys- ja liikunta-ala	Matkailu-, ravitsemis- ja talousala	Muu koulutus	Kaikki
Kokeeko että on annettavaa vaatimusmäärittelytyöhön	Erittäin paljon	1	1	6	3	3	0	0	0	2	16
	Melko paljon	1	0	12	4	10	0	4	1	1	33
	Jonkin verran	4	2	19	1	16	4	2	2	0	50
	Vähän	0	0	2	1	1	1	2	0	1	8
	Ei lainkaan	1	0	1	0	1	1	0	0	0	4
	Kaikki	7	3	40	9	31	6	8	3	4	111
Fischerin tarkka testi $p = 0,1233$											

Taulukko C.21: Kokeeko henkilö voivansa vaikuttaa vaatimusmäärittelyn avulla *
Henkilön koulutusala

		Koulutusala									
		Humanistinen ja kasvatustieteiden ala	Kulttuuriala	Yhteiskuntatieteiden, liiketalouden ja hallinnon ala	Luonnontieteiden ala	Tekniikan ja liikenteen ala	Luonnonvara- ja ympäristöala	Sosiaali-, terveys- ja liikunta-ala	Matkailu-, ravitsemis- ja talousala	Muu koulutus	Kaikki
Kokeeko voivansa vaikuttaa vaatimusmäärittelytyön avulla	Kyllä	6	2	33	8	28	3	8	3	4	95
	Ei	0	1	1	0	0	1	0	0	0	3
	Ei osaa sanoa	1	0	6	1	3	2	0	0	0	13
	Kaikki	7	3	40	9	31	6	8	3	4	111
Fischerin tarkka testi $p = 0,3067$											

Taulukko C.22: Kokeeko henkilö vaatimusmäärittelyn vaivalloiseksi * Henkilön koulutusala

		Koulutusala									
		Humanistinen ja kasvatustieteiden ala	Kulttuuriala	Yhteiskuntatieteiden, liiketalouden ja hallinnon ala	Luonnontieteiden ala	Tekniikan ja liikenteen ala	Luonnonvara- ja ympäristöala	Sosiaali-, terveys- ja liikunta-ala	Matkailu-, ravitsemis- ja talousala	Muu koulutus	Kaikki
Kokeeko vaatimusmäärittelytyön vaivalloiseksi	Kyllä	1	0	9	0	2	0	1	1	1	15
	Ei	3	2	19	6	15	3	1	1	2	52
	Ei osaa sanoa	3	1	12	3	14	3	6	1	1	44
	Kaikki	7	3	40	9	31	6	8	3	4	111
Fischerin tarkka testi $p = 0,4863$											

Taulukko C.23: Kokeeko henkilö, että hänellä on organisaation tuki osallistua vaatimusmäärittelytyöhön * Henkilön koulutusala

		Koulutusala									
		Humanistinen ja kasvatustieteiden ala	Kulttuuriala	Yhteiskuntatieteiden, liiketalouden ja hallinnon ala	Luonnontieteiden ala	Tekniikan ja liikenteen ala	Luonnonvara- ja ympäristöala	Sosiaali-, terveys- ja liikunta-ala	Matkailu-, ravitsemis- ja talousala	Muu koulutus	Kaikki
Kokeeko että oma organisaation tukee osallistumista vaatimusmäärittelytyöhön	Kyllä	3	0	15	3	10	2	2	1	4	40
	Ei	1	0	4	2	2	0	0	0	0	9
	Ei osaa sanoa	3	3	21	4	19	4	6	2	0	62
	Kaikki	7	3	40	9	31	6	8	3	4	111
Fischerin tarkka testi $p = 0,5566$											

Taulukko C.24: Haluaako henkilö osallistua jatkossa * Henkilön koulutusala

		Koulutusala									
		Humanistinen ja kasvatustieteiden ala	Kulttuuriala	Yhteiskuntatieteiden, liiketalouden ja hallinnon ala	Luonnontieteiden ala	Tekniikan ja liikenteen ala	Luonnonvara- ja ympäristöala	Sosiaali-, terveys- ja liikunta-ala	Matkailu-, ravitsemis- ja talousala	Muu koulutus	Kaikki
Haluaako osallistua jatkossa vaatimusmäärittelytyöhön	Kyllä	4	3	25	7	18	2	5	2	3	69
	Ei	0	0	2	0	1	1	1	1	1	7
	Ei osaa sanoa	3	0	13	2	12	3	2	0	0	35
	Kaikki	7	3	40	9	31	6	8	3	4	111
Fischerin tarkka testi $p = 0,436$											

Tieto vaatimusmäärittelystä selittävänä tekijänä

Taulukko C.25: Kokeeko henkilö olevansa pätevä vaatimusmäärittelytyöhön * Tietääkö henkilö mitä on vaatimusmäärittelytyö

		Tietääkö mitä on vaatimusmäärittelytyö		
		Kyllä	Ei	Kaikki
Kokeeko olevansa pätevä vaatimusmäärittelytyöhön	Kyllä	51	7	58
	Ei	7	7	14
	Ei osaa sanoa	27	12	39
	Kaikki	85	26	111
χ^2 10,85; df = 2; p = 0,004				
Odotetut frekvenssit: pienin 3,3; alle 5 suuruisia 16,7%				

Taulukko C.26: Kokeeko henkilö, että hänellä on annettavaa vaatimusmäärittelytyöhön * Tietääkö henkilö mitä on vaatimusmäärittelytyö

		Tietääkö mitä on vaatimusmäärittelytyö		
		Kyllä	Ei	Kaikki
Kokeeko että on annettavaa vaatimusmäärittelytyöhön	Erittäin paljon	15	1	16
	Melko paljon	27	6	33
	Jonkin verran	40	10	50
	Vähän	3	5	8
	Ei lainkaan	0	4	4
Kaikki		85	26	111
Fischerin tarkka testi $p = 0,000316$				

Taulukko C.27: Kokeeko henkilö voivansa vaikuttaa vaatimusmäärittelyn avulla * Tietääkö henkilö mitä on vaatimusmäärittelytyö

		Tietääkö mitä on vaatimusmäärittelytyö		
		Kyllä	Ei	Kaikki
Kokeeko voivansa vaikuttaa vaatimusmäärittelyn avulla	Kyllä	79	16	95
	Ei	2	1	3
	Ei osaa sanoa	4	9	13
	Kaikki	85	26	111
Fischerin tarkka testi $p = 0,0001811$				

Taulukko C.28: Kokeeko henkilö vaatimusmäärittelyn vaivalloiseksi * Tietääkö henkilö mitä on vaatimusmäärittelytyö

		Tietääkö mitä on vaatimusmäärittelytyö		
		Kyllä	Ei	Kaikki
Kokeeko vaatimusmäärittelytyön vaivalloiseksi	Kyllä	12	3	15
	Ei	50	2	52
	Ei osaa sanoa	23	21	44
	Kaikki	85	26	111
χ^2 25,70; df = 2; p = 0,000				
Odotetut frekvenssit: pienin 3,5; alle 5 suuruisia 16,7%				

Taulukko C.29: Kokeeko henkilö, että hänellä on organisaation tuki osallistua vaatimusmäärittelytyöhön * Tietääkö henkilö mitä on vaatimusmäärittelytyö

		Tietääkö mitä on vaatimusmäärittelytyö		
		Kyllä	Ei	Kaikki
Kokeeko että oma organisaation tukee osallistumista vaatimusmäärittelytyöhön	Kyllä	36	4	40
	Ei	6	3	9
	Ei osaa sanoa	43	19	62
	Kaikki	85	26	111
χ^2 6,31; df = 2; p = 0,043				
Odotetut frekvenssit: pienin 2,1; alle 5 suuruisia 16,7%				

Taulukko C.30: Haluaako henkilö osallistua jatkossa * Tietääkö henkilö mitä on vaatimusmäärittelytyö

		Tietääkö mitä on vaatimusmäärittelytyö		
		Kyllä	Ei	Kaikki
Haluaako osallistua jatkossa vaatimusmäärittelytyöhön	Kyllä	59	10	69
	Ei	5	2	7
	Ei osaa sanoa	21	14	35
	Kaikki	85	26	111
χ^2 8,53; df = 2; p = 0,014				
Odotetut frekvenssit: pienin 1,6; alle 5 suuruisia 16,7%				

Aiempi osallistuminen selittäväenä tekijänä

Taulukko C.31: Tietääkö henkilö mikä vaatimusmäärittely on * Onko henkilö osallistunut vaatimusmäärittelytyöhön

		Onko osallistunut aiemmin vaatimusmäärittelytyöhön		
		Kyllä	Ei	Kaikki
Tietääkö mitä on vaatimusmäärittelytyö	Kyllä	59	26	85
	Ei	3	23	26
	Kaikki	62	49	111
$\chi^2 27,04; df = 1; p = 0,000$				
Odotetut frekvenssit: pienin 11,5; alle 5 suuruisia 0,0%				

Taulukko C.32: Kokeeko henkilö olevansa pätevä vaatimusmäärittelytyöhön * Onko henkilö osallistunut vaatimusmäärittelytyöhön

		Onko osallistunut aiemmin vaatimus- määrittelytyöhön		
		Kyllä	Ei	Kaikki
Kokeeko olevansa pätevä vaatimus- määrittelytyöhön	Kyllä	42	16	58
	Ei	4	10	14
	Ei osaa sanoa	16	23	39
	Kaikki	62	49	111
$\chi^2_{14,15}; df = 2; p = 0,001$				
Odotetut frekvenssit: pienin 6,2; alle 5 suuruisia 0,0%				

Taulukko C.33: Kokeeko henkilö, että hänellä on annettavaa vaatimusmäärittelytyöhön * Onko henkilö osallistunut vaatimusmäärittelytyöhön

		Onko osallistunut aiemmin vaatimus- määrittelytyöhön		
		Kyllä	Ei	Kaikki
Kokeeko että on annettavaa vaatimusmäärittely- työhön	Erittäin paljon	14	2	16
	Melko paljon	23	10	33
	Jonkin verran	24	25	50
	Vähän	1	7	8
	Ei lainkaan	0	4	4
Kaikki		62	49	111
Fischerin tarkka testi $p = 0,0001164$				

Taulukko C.34: Kokeeko henkilö voivansa vaikuttaa vaatimusmäärittelyn avulla *
Onko henkilö osallistunut vaatimusmäärittelytyöhön

		Onko osallistunut aiemmin vaatimus- määrittelytyöhön		
		Kyllä	Ei	Kaikki
Kokeeko voivansa vaikuttaa vaatimusmäärittelyn avulla	Kyllä	58	37	95
	Ei	2	1	3
	Ei osaa sanoa	2	11	13
	Kaikki	62	49	111
Fischerin tarkka testi $p = 0,003787$				

Taulukko C.35: Kokeeko henkilö vaatimusmäärittelyn vaivalloiseksi * Onko henkilö
osallistunut vaatimusmäärittelytyöhön

		Onko osallistunut aiemmin vaatimus- määrittelytyöhön		
		Kyllä	Ei	Kaikki
Kokeeko vaatimus- määrittelytyön vaivalloiseksi	Kyllä	12	3	15
	Ei	38	14	52
	Ei osaa sanoa	12	32	44
	Kaikki	62	49	111
$\chi^2_{24,38}; df = 2; p = 0,000$				
Odotetut frekvenssit: pienin 6,6; alle 5 suuruisia 0,0%				

Taulukko C.36: Kokeeko henkilö, että hänellä on organisaation tuki osallistua vaatimusmäärittelytyöhön * Onko henkilö osallistunut vaatimusmäärittelytyöhön

		Onko osallistunut aiemmin vaatimus- määrittelytyöhön		Kaikki
		Kyllä	Ei	
Kokeeko että oma organisaation tukee osallistumista vaatimus- määrittelytyöhön	Kyllä	29	11	40
	Ei	5	4	9
	Ei osaa sanoa	28	34	62
	Kaikki	62	49	111
$\chi^2 7,37; df = 2; p = 0,025$				
Odotetut frekvenssit: pienin 4,0; alle 5 suuruisia 0,0%				

Taulukko C.37: Haluaako henkilö osallistua jatkossa * Onko henkilö osallistunut vaatimusmäärittelytyöhön

		Onko osallistunut aiemmin vaatimus- määrittelytyöhön		Kaikki
		Kyllä	Ei	
Haluaako osallistua jatkossa vaatimus- määrittelytyöhön	Kyllä	44	25	69
	Ei	4	3	7
	Ei osaa sanoa	14	21	35
	Kaikki	62	49	111
Fischerin tarkka testi $p = 0,0661$				