

**KÄYTTÖLIITTYMÄPROJEKTIEIN VAATIMUSTEN HALLINNAN
KEHITTÄMINEN – CASE KEMPPI**



Ylempi ammattikorkeakoulututkinto

Teknologiaosaamisen johtamisen koulutusohjelma

Syksy, 2022

Jarno Kolkka

Hitsauslaitteissa nopeasti muuttuvat teknologiat ja kasvava kilpailu asettavat paineita tuotteiden kehittämiseksi. Hitsauslaitteiden valmistajien on kilpailukykyä pitääkseen koko ajan kehitettävä hitsauslaitteistoa ja vastattava asiakkaiden tarpeisiin. Asiakkaiden tarpeet ja vaatimukset liittyvät enenevässä määrin käyttöliittymiin ja niiden käytettävyyteen.

Opinnäytetyön tarkoituksena on tehdä Kempille kehitysehdotus vaatimusmäärittelyn toimintamallista. Toimintamallin pyrkimyksenä on parantaa projektien hallintaa, mielekkyyttä sekä ennustettavuutta.

Opinnäytetyössä tarkastellaan vaatimustenhallintaa ohjelmistoprojektissa ja sen kehittämistä vaatimusten priorisointimenetelmää kehittämällä. Aluksi tutustutaan ohjelmistojen vaatimuksiin, ohjelmistokehitysmenetelmiin ja vaatimusten priorisointimenetelmiin kirjallisuuden pohjalta. Kirjallisuuskatsauksen perusteella muodostetaan viitekehys, jossa käsitellään vaatimuksia sidosryhmien ja erilaisten ohjelmistokehityksen toimintatapojen ja johtamisen kautta. Empiirinen osa toteutetaan Delfoi-menetelmällä hyödyntäen konstruktivistista tutkimusotetta, jonka perustana käytetään muodostettua viitekehystä.

Keskeisiä asioita toiminnan laadun parantamiseen vaatimusmäärittelyssä ovat toimenpiteet, jotka koskevat vaatimuksien käsittelyn vakiointia kehitysehdotuksen avulla. Toiminnan laatua voidaan tehostaa parantamalla läpinäkyvyyttä. Läpinäkyvyyttä lisää päätöksen dokumentointi.

Avainsanat Vaatimusten hallinta, Vaatimusten määrittely, Vaatimusten priorisointi

Sivut 46 sivua ja liitteitä 11 sivua

Fast developing technologies and increasing competition in welding equipment put pressure on the development of products. Welding equipment manufacturers need to develop the welding equipment constantly and meet the needs of customers to maintain their competitiveness. Customer needs and requirements are increasingly related to user interfaces and their usability.

The purpose of the thesis is to submit a development proposal to the Kemppi, for operating model about the requirement definition. The aim is to improve project management, meaningfulness and predictability.

The thesis examines the management of requirements in the software project and its development through the development of a prioritization method. At begin, we will learn about software requirements, software development methods and requirements prioritization based on the literature. Based on the literature review, a reference framework will be established to process requirements through stakeholder and different software development method and management. The empirical part shall be carried out using the Delfoi method using a construction study, which is based on the created framework of reference.

The key elements for improving the quality of operations in the definition of requirements are measures, which is concerning the standardisation of the treatment of requirements by means of a method developed. The quality of operations can be improved by improving transparency by investing in the documentation of decisions.

Keywords requirements management, requirements specification, requirement prioritization

Pages 46 pages and appendices 11 pages

Sisällys

Lyhenteet ja merkinnät	2
1 Johdanto	1
1.1 Yritysesittely	2
1.2 Perustelut aiheen valinnalle	3
1.3 Tutkimusaiheen tavoite ja tarkoitus sekä rajaus	4
1.4 Tutkimuskysymykset	5
1.5 Tutkimuksen rakenne.....	5
2 Tutkimuksen tietoperusta	7
2.1 Ohjelmiston vaatimusmäärittely	7
2.1.1 Toiminnalliset ja Ei-toiminnalliset vaatimukset	8
2.2 Ohjelmiston kehitysmenetelmät	8
2.2.1 Lineaarinen ohjelmistokehitysmenetelmä	9
2.2.2 Iteratiivinen ohjelmistokehitys	11
2.2.3 Scrum on yksi ketterä ohjelmistokehitysmenetelmä	12
2.3 Vaatimusten priorisointimenetelmistä.....	14
2.3.1 Value vs. effort-priorisointimenetelmä	15
2.3.2 Kano-malli.....	20
3 Tutkimuksen empiirinen osuus	23
3.1 Tutkimusmenetelmät ja aineistonkeruu.....	24
3.2 Tutkimuksen kulku	25
3.3 Tutkimuksen lähtökohdaksi luotu konstruktio vaatimustenhallinnasta	28
3.4 Tutkimustulokset	30
3.5 Tulosten analysointi ja luotettavuuden pohdinta.....	37
3.6 Yhteenveto tutkimustuloksista	39
4 Pohdinta	40
4.1 Jatkotutkimusehdotukset	43
Lähteet.....	44

Kuvat

Kuva 1. Kemppe globaalisti lähellä (Kemppe, 2022-a)

Kuva 2. Tutkimuksen rakenne havainnollistettu

Kuva 3. Tutkimuksen teoreettinen viitekehys

Kuva 4. Mukailen Roycen esittelemää vesiputousmallia (Royce, 1970, s. 2).

Kuva 5. Iteratiivisen ohjelmistokehityksen prosessikaavio mukailen teoriaa.

Kuva 6. Scrumin yksittäisen sprintin prosessikaavio mukailen teoriaa

Kuva 7. Mukailen Productplan nelikenttämallia. (Productplan, 2022)

Kuva 8. Mukailen Noriaki Kanon Kano-mallia

Kuva 9. Lähtökohdaksi luotu konstruktio vaatimustenhallinnasta vuokaaviolla esitettynä

Kuva 10. Kyselyn tulos vaatimusten säilytyspaikasta

Kuva 11. Kysymys vaatimusten priorisoinnin tarpeellisuudesta osana vaatimuksen käsittelyä.

Taulukot

Taulukko 1. Mukailen Kano-matriisia (Hietikko, 2021-b)

Liitteet

Liite 1: Esimerkkityökalu Value vs. Effort-menetelmästä

Liite 2: Kyselykierros 1

Liite 3: Kyselykierros 2

Lyhenteet ja merkinnät

CONFLUENCE	Atlassian Confluence™ on verkkoselaimessa toimiva organisaatiowikiohjelmisto. Se on verkkosivusto, jonka sisältöä käyttäjät voivat itse muokata verkkoselaimellaan. (Atlassian, n.d.)
JIRA	Atlassian JIRA™ on verkkoselaimessa toimiva tehtävienhallintaohjelmisto, jonka on tehnyt australialainen vuonna 2002 perustettu ohjelmistoyritys Atlassian. (Atlassian, n.d.)
LEAN	Lean on liiketoiminnan johtamisfilosofia, joka perustuu toiminnan järkevöittämiseen turhan tekemisen poistamisella ja jatkuvalla parantamisella. (Airila, 2022)
MAG	Metal Active Gas welding eli metallikaasukaarihitsaus, jossa suojakaasu on aktiivinen, joka reagoi sulassa metallissa olevien aineiden kanssa. Menetelmässä valokaari palaa suojakaasun ympäröimänä hitsattavan lisäainelangan ja työkappaleen välissä. Sula metalli siirtyy pisaroina lisäainelangasta hitsisulaan. (ESAB, n.d., s. 503)
MIG	Metal Inert Gas welding eli metallikaasukaarihitsaus, jossa suojakaasu on inertti eli reagoimaton sulassa metallissa olevien aineiden kanssa. Menetelmässä valokaari palaa suojakaasun ympäröimänä hitsattavan lisäainelangan ja työkappaleen välissä. Sula metalli siirtyy pisaroina lisäainelangasta hitsisulaan. (ESAB, n.d., s. 503)
MMA	Manual Metal Arc welding eli puikkokaarihitsaus menetelmä, jossa lisäaine on hitsauspuikossa, joka sulautuu hitsauksen aikana perusaineeseen. (ESAB, n.d., s. 502)

Scrum	Scrum on kevyt ohjelmistokehityksen menetelmäkehys, joka auttaa ohjelmistokehittäjiä ja tiimejä luomaan arvoa mukautuvilla ratkaisuilla monimutkaisiin ongelmiin. (Schwaber & Sutherland, 2020, s. 2)
Sidosryhmä	Sidosryhmät ovat projektin ulkoisia toimijoita, joilla on mahdollisuus vaikuttaa projektiin tai johon projekti vaikuttaa, kuten myynti, huolto, loppukäyttäjät tai muut suunnittelija ryhmät. (Tieteen termipankki, 2022)
Sprint	Sprintti tarkoittaa, että ohjelmistokehitys tapahtuu 1–4 viikon iteraatioissa. Sprintti on keskeinen elementti scrum menetelmällä tapahtuvassa ohjelmistokehityksessä. (Schwaber & Sutherland, 2020, s. 8)
TIG	Tunsten Inert Gas welding eli kaasukaarihitsausprosessi, jossa valokaari palaa sulamattoman elektrodin ja työkappaleen välissä inertin eli reagoimattoman suojakaasun ympäröimänä. Hitsausta voidaan tehdä lisäaineen kanssa tai ilman lisäainetta. (ESAB, n.d., s. 506)

1 Johdanto

Standish groupin kaaos -raportin mukaan ainoastaan 16,2 % ohjelmistokehitysprojekteista onnistutaan toteuttamaan sovitussa aikataulussa ja budjetissa niin, että ohjelmisto sisältää luvatut ominaisuudet. Suurin osa projekteista ylittää budjetin tai aikataulun ja luvatuissa ominaisuuksissa on puutteita. Ohjelmistoprojekteista jopa 31,1 % epäonnistuu tai keskeytetään ennen valmistumista. (Johnson, 2020)

Hitsauslaitteissa nopeasti muuttuvat teknologiat ja kasvava kilpailu asettavat paineita tuotteiden kehittämiseksi. Hitsauslaitteiden valmistajien on kilpailukykyä säilyttääkseen koko ajan kehitettävä hitsauslaitteistoja ja vastattava asiakkaiden tarpeisiin. Asiakkaiden tarpeet ja vaatimukset liittyvät enenevässä määrin käyttöliittymiin ja niiden käytettävyyteen.

Käyttöliittymien suunnittelussa tämä tarkoittaa muun muassa asiakastarpeen syvällistä ymmärtämistä. Teknologian nopean kehityksen vuoksi hitsauslaitteiden käyttöliittymäohjelmistoja ei voi usein suoraan verrata aikaisempiin tuotteisiin. Tämä on osoittanut, että asiakkaat ja muut sidosryhmät eivät vaatimuksia esittäessään osaa sanoa mitä haluavat tai tarvitsevat. Vaatimusten esittäjien on hyvin vaikea etukäteen pystyä miettimään kaikkea toiminnallisuutta sillä tasolla, että vaatimukset voidaan lukita vaatimuksia kerätessä. Onkin hyvin tavallista, että vaatimuksen esittänyt henkilö alkaa haluta muutoksia tuotteeseen heti kun näkee valmiin lopputuloksen. Yhtenä tähän liittyvänä riskinä on se, että projektitiimi tulkitsee dokumentoituja asiakkaan vaatimuksia väärin, tai asiakas ei tule ymmärretyksi täysin vaatimusten kirjaushetkellä, eli vaatimukset ovat jo lähtökohtaisesti dokumentoitu väärin. On siis tärkeää perehtyä ohjelmistosuunnittelun vaatimustenhallinnan prosessin kehittämiseen ja vaatimusten priorisointimenetelmiin, jos halutaan pysyä kehityksen ja kilpailun kärjessä samalla keskittyen tuomaan todellista asiakasarvoa tuotteisiin.

Tämä opinnäytetyö on toteutettu osittain kirjallisuustutkimuksena ja osittain haastattelututkimuksena. Haastattelututkimus on tehty Delfoi-menetelmällä hyödyntäen konstruktivistista tutkimusotetta. Työn ensimmäisessä osuudessa käsitellään tutkimuksen teoriapohjaa. Siinä on tietoa ohjelmistokehityksen toimintamalleista,

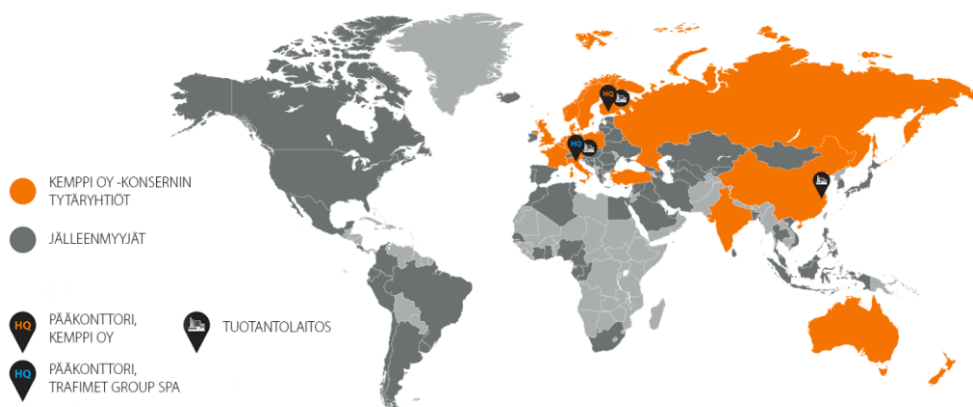
priorisointimenetelmistä ja asioista, joilla on vaikutusta vaatimustenhallintaan ohjelmistokehityksessä. Tämän jälkeen perehdytään tutkimuksessa tehtyihin havaintoihin ja tuloksiin. Lopuksi löytyvät johtopäätökset ja suositukset jatkotoimenpiteille. Käsittelyyn otettiin vain yksi priorisointimenetelmän käyttöönotto, ettei tutkimuksesta tulisi liian laaja. Lisäksi tutkimus rajoitettiin koskemaan vain ainoastaan vaatimusten määrittelyä ja hallintaa projektin aikana.

1.1 Yritysesittely

Kaarihitsusteollisuuden laitevalmistaja Kemppe Oy on vuonna 1949 perustettu suomalainen perheyritys ja yrityksen pääkonttori sijaitsee Lahdessa. Yritys valmistaa MIG/MAG-, TIG- ja MMA- hitsausvirtalähteitä käsin hitsaukseen, mekanisoituun hitsaukseen sekä automatisoituun hitsaukseen. Hitsausvirtalähteiden lisäksi Kemppe Oy valmistaa hitsauspolttimet, lisälaitteet, kuten langansyöttölaitteet, sekä erilaiset suojarusteet. Valmistettavien tuotteiden lisäksi Kemppe Oy tarjoaa kehittyneitä hitsaustuotannon hallintaohjelmistoja ja asiantuntijapalveluita. Yritys tarjoaa ratkaisut vaativan teollisuuden lisäksi myös kuluttajille.

Kemppeillä työskentelee globaalisti 17 eri maassa yhteensä lähes 800 työntekijää. Yrityksen liikevaihto vuonna 2020 oli 140 Meur. Kempin kumppaniverkosto kattaa yli 70 maata, joten sen asiantuntemus on paikallisesti lähellä. (Kemppe, 2022-b)

Kuva 1. Kemppe globaalisti lähellä (Kemppe, 2022-a)



1.2 Perustelut aiheen valinnalle

Motiivina tähän tutkimukseen on tutkijan itsensä pitkäaikainen kiinnostus projektinhallinnan kehittämiseen, projektien johtamiseen ja monimutkaisten projektien läpiviemiseen. Työn tilaajalla oli myös tarve tämän kaltaiselle tutkimukselle ja aihe on mielenkiintoinen hitsauslaittevalmistuksen näkökulmasta, koska laitteiden käyttöliittymät ovat digitaalisessa murroksessa. Megatrendinä hitsauslaitteissakin on yleisesti ollut käyttöliittymien monimutkaistuminen graafisten käyttöliittymäpaneelien yleistyessä. Graafiset käyttöliittymät ovat vauhdittaneet digitalisaation kehittymistä myös hitsauslaitteissa, mikä taas on edistänyt erilaisten pilvipalveluiden kehittymistä. Tämä johtaa helposti nykyisin ohjelmistosuunnitteluprojekteissa siihen, että vaatimuksia tunnistetaan enemmän kuin mitä projektin aikataulu ja resurssit mahdollistavat.

Moni eri taho, kuten markkinointi, myynti, asiakaspalvelu, huolto, tytäryhtiöt ja asiakkaat esittävät vaatimuksia hitsauslaitteiden käyttöliittymän ominaisuuksiin. Myös maantieteellisesti eri paikoissa voi esiintyä hyvinkin erilaisia tarpeita käyttöliittymän ominaisuuksien suhteen. Ohjelmistojen kasvaessa ja monimutkaistuessa projektien ennustettavuus ajallisesti sekä resurssimielessä on haastavaa. Projektien aikana muutoksilta ja lisäyksiltä ominaisuuksien vaatimuksissa ei voi välttyä, joten niitä täytyy pystyä hallitsemaan.

Tähän käyttöliittymien ohjelmistosuunnittelun ongelmaan yhtenä ratkaisuvaihtoehtona voisi olla vaatimusmäärittelyn vaatimusten priorisointi. Priorisoinnilla yritetään valita käyttöliittymään sellaiset ominaisuudet, jotka vastaavat eri tahojen tärkeimmiksi näkemiä vaatimuksia. Tällä tavoin voidaan ennakoida ja hallita hyvissä ajoin projektin haasteita esimerkiksi ylittyvän budjetin, aikataulun tai puutteellisen käyttöliittymän ominaisuudessa.

1.3 Tutkimusaiheen tavoite ja tarkoitus sekä rajaus

Tämän työn tavoite on vastata tutkimusongelmien kysymyksiin sekä tehdä kehitysehdotus toimintamallista vaatimusten dokumentoimiseksi ja niiden käsittelemiseksi. Työssä tutkitaan, mikä käyttöliittymien ohjelmistosuunnittelun vaatimusten hallintamenetelmä olisi paras ratkaisu yrityksen ja hitsauslaitteiden käyttöliittymien kannalta. Toimintamallissa vaatimukset pisteytetään ja priorisoidaan projekteille. Priorisoinneissa pitää olla mukana ne henkilöt yrityksen eri sidosryhmistä, joihin kyseinen vaatimus liittyy. Kun vaatimukset on käsitelty, niiden hallitsemiseksi täytyy olla sovittu toimintamalli. Toimintamallista tulee käydä ilmi, miten vaatimukset ja kaikki niihin liittyvät tiedot säilytetään sellaisenaan. Myös ne vaatimukset, jotka eivät sillä hetkellä tai siinä projektissa ole toteutuskelpoisia, tulee säilyttää.

Vaatimuksiin liittyvät päätökset ja perustelut kaikista valinnoista tulee säilyttää dokumentoinnin mukana. Jos vaatimus päätetään jättää toteuttamatta, niin siihen pitää voida myöhemmin palata. Seuraavassa projektissa on mahdollista, että aikaisemmin hylättyjä ominaisuuksia tullaan toteuttamaan. Tulevissa projekteissa pitää aina haluttujen vaatimuksien osalta kuitenkin priorisointi suorittaa uudelleen. Projekteissa syntyvän valtavan tietomassan hallitsemiseen tarvitaan toimiva järjestelmä, jolloin sitä voidaan hyödyntää aina uudelleen.

Tämän työn tarkoituksena on tehdä kehitysehdotus vaatimusmäärittelyn toimintamallista. Pyrkimyksenä on parantaa projektien hallintaa, mielekkyyttä sekä ennustettavuutta. Työn jalkauttamisesta käytäntöön ei olla sovittu työn tilaajan kanssa.

Tehtävä rajataan käsittelemään vain vaatimusten määrittelyä ja hallintaa projektin aikana. Muihin projektin vaiheisiin ei oteta kantaa. Työssä ei lähdetä selvittämään tarkasti, soveltuuko tämä vaatimusten hallintamenetelmä myös muille yrityksen osastoille käytettäväksi.

1.4 Tutkimuskysymykset

Tämän työn tutkimusongelmana on selvittää, miten vaatimustenhallinta tulee toteuttaa projektin aikana, niin että se parantaa käyttöliittymien laatua sekä nopeuttaa tuotekehityksen läpimenoaikaa käyttöliittymien kehityksen ja toteutuksen osalta.

Täsmentäviä kysymyksiä ovat:

1. Miten vaatimukset saadaan kerättyä yhdenmukaisella tavalla talteen eri sidosryhmiltä ja dokumentoitua ne projektien käytettäväiksi?
2. Miten nämä vaatimukset käsitellään sekä priorisoidaan projektissa?
3. Onko yrityksessä sopiva järjestelmä, mihin työssä saavutettuja toimenpiteitä voisi implementoida jatkossa?

1.5 Tutkimuksen rakenne

Johdannossa esitellään tutkimuksen tausta ja perustelut aiheen valinnalle, yritysesittely, tutkimusaiheen rajaus sekä tutkimuskysymykset ja tavoite, johdannon päättää tutkimuksen rakenteen kuvaus.

Ohjelmistonkehitysmenetelmä esittelee tavan, millä yritys toteuttaa ohjelmistoprojekteja ja minkälaisia vaatimuksia se asettaa vaatimustenhallinnalle ja dokumentaatiolle.

Kirjallisuuskatsauksen avulla tutkimuksessa tarkastellaan toimivaa ohjelmistokehitysmenetelmää, kun vaatimukset muuttuvat ja tarkentuvat projektin aikana.

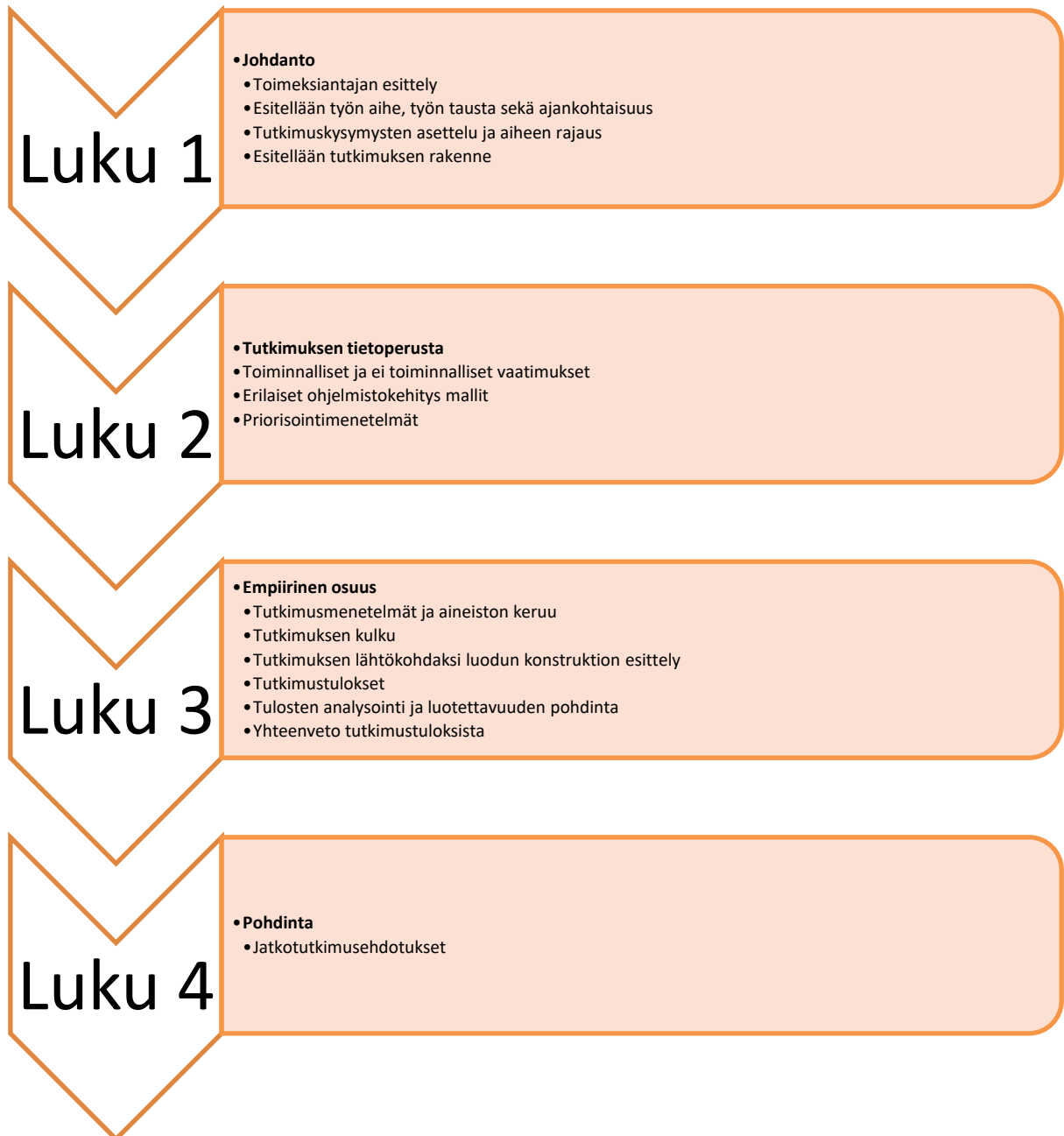
Ohjelmistokehitysmenetelmiin keskittyneiden teorioiden jälkeen tarkastellaan kirjallisuuskatsauksessa seuraavaksi vaatimusten priorisointimenetelmiä.

Priorisointityökaluina vaatimustenhallintaan esitellään Value vs. effort-menetelmä ja Kano-malli.

Luvussa 3 esitellään tutkimusmenetelmät ja aineiston keruu sekä käydään läpi tutkimuksen kulku. Tutkimuksen lähtökohdaksi luotu konstruktio esitellään ennen tutkimustuloksia, luvun

lopuksi on tulosten analysointi ja luotettavuuden pohdinta sekä yhteenveto tutkimustuloksista. Viimeinen luku 4 on pohdinta, jossa esitellään tutkimuksen johtopäätökset ja esitellään tutkimuksen keskeiset tulokset. Pohdinnassa myös arvioidaan työtä ja sen rajoitteita, lopuksi pohditaan, saatiinko vastaus tutkimuskysymyksiin ja esitetään mahdolliset aiheet jatkotutkimuksille.

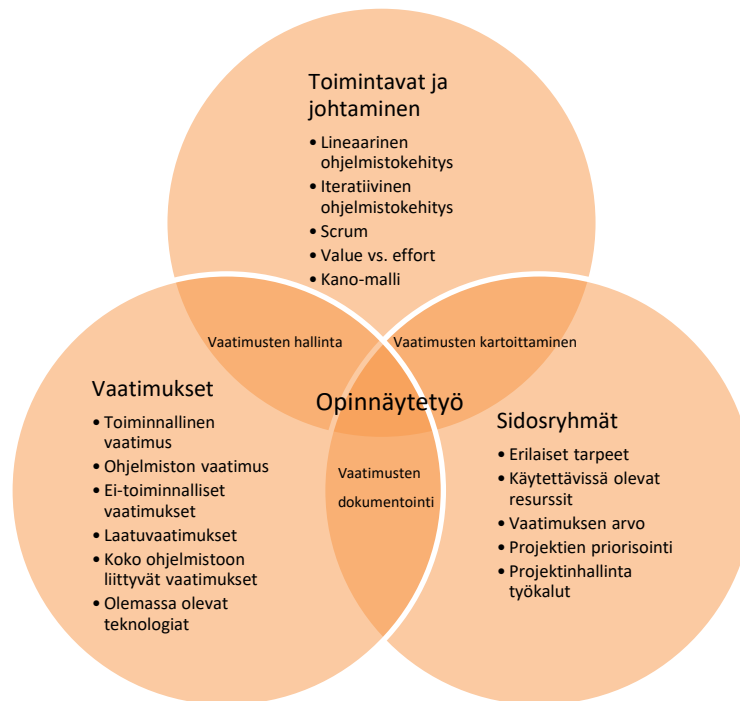
Kuva 2. Tutkimuksen rakenne havainnollistettu



2 Tutkimuksen tietoperusta

Tässä tutkimuksessa tietoperustana teoreettisen tutkimuksen osuudessa käytetään projektihallinnan ja osaamisen johtamisen kirjallisuutta, verkkojulkaisuja ja työn tilaajan muita materiaaleja. Tämän tutkimuksen muoto on ongelman selittämiseen pyrkivä tutkimus, tulosten päättelymenetelmänä on abduktiivinen päättely, joka perustuu siihen, että uuden teorian muodostus on mahdollista vain silloin, kun havaintojen tekoon liittyy teoreettinen johtojatous (Anttila, 2014).

Kuva 3. Tutkimuksen teoreettinen viitekehys



2.1 Ohjelmiston vaatimusmäärittely

Ohjelmiston vaatimusmäärittely voi tarkoittaa eri ihmisille eri asioita. Seuraavat asiat mielletään aina kuuluvaksi ohjelmiston vaatimusmäärittelyyn. Järjestelmälle asetetaan toiminnallisia ja ei-toiminnallisia vaatimuksia, jotka järjestelmän tulee toteuttaa. Osa vaatimuksista koskee järjestelmän ohjelmistolla toteutettavaa osaa, jolloin on kyse ohjelmistovaatimuksista. Osa vaatimuksista on järjestelmän rajoituksia tai teknisiä

ominaisuuksia. Vaatimukset tulevat järjestelmän kanssa tekemisissä olevilta sidosryhmiltä. Vaatimusmäärittelyssä selvitetään, mitä järjestelmältä vaaditaan ja miten esille nousseet vaatimukset saadaan kuvatuksi selkeästi projektille eteenpäin jatkokehitystä varten. (Paakki, 2011, s. 1)

2.1.1 Toiminnalliset ja Ei-toiminnalliset vaatimukset

Toiminnallisilla vaatimuksilla tarkoitetaan kaikkia niitä vaatimuksia mitä ohjelmistolla voi tehdä, eli ohjelmiston tarjoamia toimintoja. Ei-toiminnalliset vaatimukset ovat koko ohjelmistoa koskevat "laatuvaatimukset" esimerkkeinä seuraavia, kuten käytettävyys ja tietoturva sekä ohjelmiston toimintaympäristön sille asettamat rajoitteet. Ketterissä menetelmissä vaatimukset dokumentoidaan usein "user storyina" eli käyttäjätarinoina. Toiminnallisten ja ei-toiminnallisten vaatimusten eroavaisuudet voidaan tiivistää seuraavasti: toiminnalliset vaatimukset, jotka kuvaavat usein järjestelmän yksittäisiä toimintoja, ei-toiminnalliset vaatimukset koskevat useimmiten koko järjestelmää, ja vaikuttavat siihen, miten järjestelmän perusrakenne eli arkkitehtuuri tulee suunnitella. (Luukkainen, 2021-b)

2.2 Ohjelmiston kehitysmenetelmät

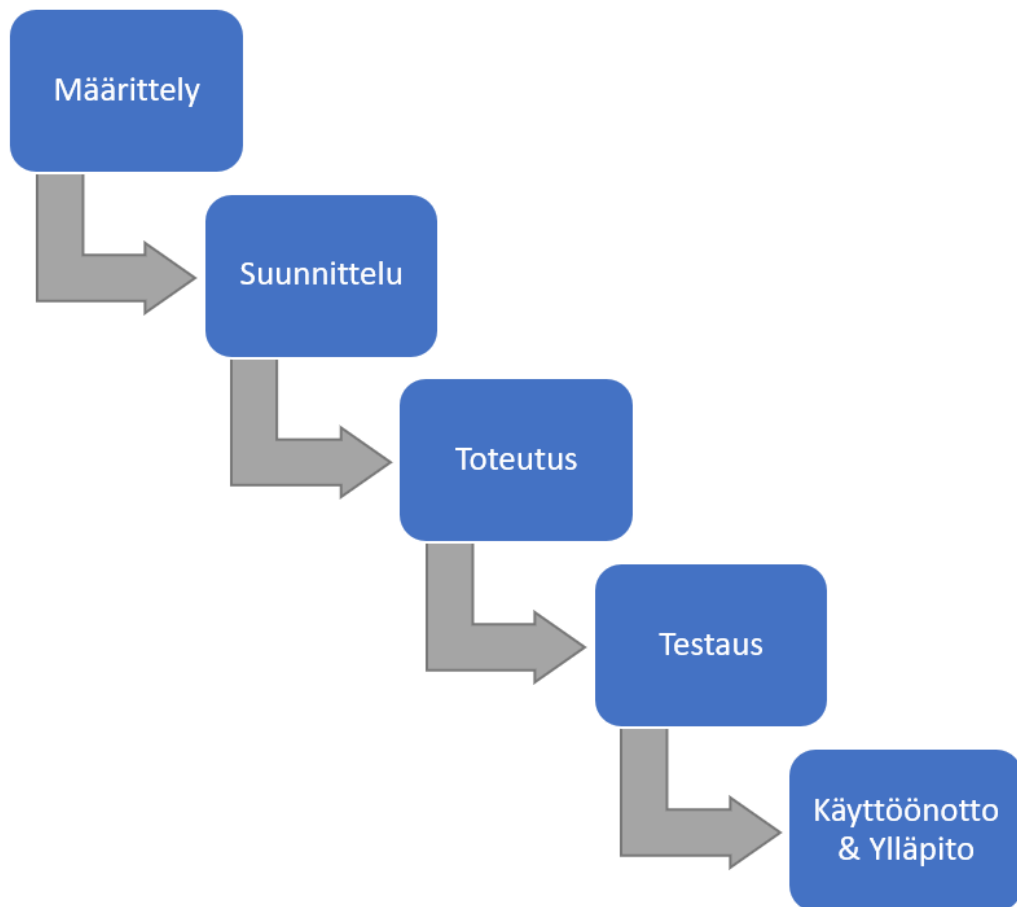
Ohjelmistokehitysmenetelmät voidaan jakaa ylätasolla kahteen eri luokkaan: suunnitelmaohjautuvat menetelmät sekä ketterät ohjelmistokehitysmenetelmät. Kaikkien menetelmien taustalla on kuitenkin seuraava toiminta ajatus: Jos jokin asia on liian monimutkainen hallittavaksi kokonaisuudeksi, hajota se pienempiin osiin ja keskity yhteen osaan kerrallaan. Suunnitelmavetoisessa ohjelmistotuotantomenetelmässä keskiössä on etukäteen tehty tarkka projektisuunnitelma ja sen noudattaminen. Ohjelmisto pyritään suunnittelemaan etukäteen valmiiksi ja toteutuksen ajatellaan olevan triviaalivaihe. Ketterät ohjelmistokehitysmenetelmät lähtevät oletuksesta, että kyseessä ei ole kontrolloitu prosessi, joka voidaan täysin tarkasti etukäteen suunnitella. Ketterät menetelmät ajattelevat ohjelmiston kehittämisen tuotekehitysprojektina, jonka kehitykseen sisältyy epävarmuutta ja tuntemattomia seikkoja. Tämän vuoksi ohjelmiston kehitysprojektiin sopii empiirinen prosessi paremmin kuin tarkkaan etukäteissuunnitelmaan perustuva lähestymistapa.

Ohjelmistokehitysmenetelmät keskittyvät projektin ohjaukseen ja toimintatapoihin, eikä ohjelmointitekniikoihin tai ohjelmiston arkkitehtuuriin. (Luukkainen, 2021-a)

2.2.1 Lineaarinen ohjelmistokehitysmenetelmä

Vesiputousmalli eli suoraviivainen lineaarinen ohjelmistokehitysmenetelmä, jota ruvettiin kutsumaan vesiputousmalliksi, saavutti nopeasti suosiota ohjelmistokehityksessä. Malli sai alkunsa Winston Roycen vuonna 1970 julkaiseman artikkelin perusteella *Management of the development of Large Software*, joka pohdiskelee isojen ohjelmistojen kehittämiseen liittyvää problematiikkaa. Artikkelissa Royce esittelee yksinkertaisen prosessimallin eli ohjeiston työvaiheiden jaksottamiseen, jossa ohjelmistonkehityksen elinkaaren vaiheet suoritetaan lineaarisesti peräkkäin (Royce, 1970, s. 2). Toimintamalli on monella järkeenkäypä menetelmä yleisesti, ensin kannattaa selvittää mitä ollaan tekemässä ja suunnittelu hoidetaan vasta tämän vaiheen jälkeen. Kun suunnitelma tuotteesta on valmis, voidaan itse tuote valmistaa ja sen jälkeen vasta testata, että se toimii kuten haluttiin, näinhän toimitaan monella muullakin tuotannon alalla. Royce kyllä esittelee vesiputousmallin artikkelin sivulla 2, mutta toteaa myöhemmin artikkelissa, ettei se sovellu monimutkaisten ohjelmistoprojektien tekotavaksi (Luukkainen, 2021-a)

Kuva 4. Mukailleen Roycen esittelemää vesiputousmallia (Royce, 1970, s. 2).

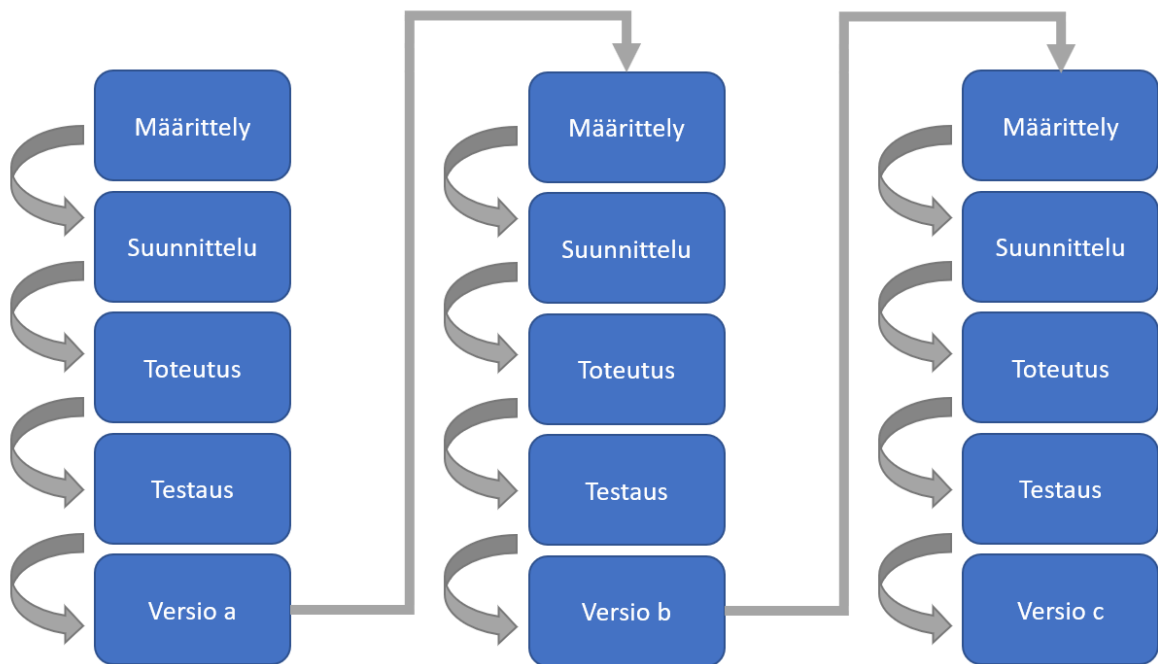


Vesiputousmallin mukaisessa ohjelmistotuotannossa voi tulla isoja haasteita monimutkaisissa ohjelmistoprojekteissa, koska vesiputousmalli olettaa, että ohjelmistokehityksen vaiheet etenevät järjestyksessä koko ajan eteenpäin. Suurin ongelma on se, että tehtiin vaatimusmäärittely, miten huolellisesti tahansa, käytännössä muutoksia tulee aina matkan varrella. On hyvin tyypillistä ohjelmistokehityksessä, että asiakas ei ohjelmistoa tilatessaan tiedä tai osaa sanoa mitä haluaa tarkalleen, mutta alkaa haluta muutoksia heti, kun näkee valmiin lopputuloksen. Vesiputousmalliin perustuvan ohjelmistosuunnittelun takana onkin analogia muihin insinööritieteisiin: Toteutettava projekti tulee ensin määritellä ja suunnitella aukottomasti, tämän jälkeen toteutus on suoraviivainen operaatio. (Luukkainen, 2021-a)

2.2.2 Iteratiivinen ohjelmistokehitys

Iteratiivisesta ohjelmistokehityksestä saatetaan käyttää myös seuraavia nimityksiä: spiraalimalli tai inkrementaalinen ohjelmistokehitys. Tämä ohjelmistokehityksen iteratiivinen tapa tehdä ohjelmistoja alkoi yleistyä 90-luvulla. Iteratiivisessa ohjelmistokehityksessä työ jaetaan pienempiin aikaväleihin, eli iteraatioihin. Jokaisen iteraation aikana tehdään uusi määrittely-, suunnittelu-, toteutus- sekä testausvaihe ohjelmistolle. Toisin kuin vesiputousmallin mukaisessa kehityksessä, iteratiivisesti edetessä ei tehdä aluksi lopullisia suunnitelmia ja määritelmiä, vaan ohjelmisto kehittyy vähitellen. Ennen uuden iteraation aloitusta asiakas näkee sen hetkisen version ohjelmasta ja pystyy vaikuttamaan seuraavien iteraatioiden kulkuun. (Luukkainen, 2021-a)

Kuva 5. Iteratiivisen ohjelmistokehityksen prosessikaavio mukaillen teoriaa.



Iteratiivisessa ohjelmistokehityksessä on myös omat haasteensa, esimerkiksi kehityksen mittaaminen on vaikeaa prosessin ollessa näkymätöntä ja jokaisen iteraatiokierroksen jälkeen ohjelmiston julkaiseminen asiakkaille voi olla työlästä ja kallista. Jatkuva ohjelmiston muokkaaminen voi vaurioittaa ohjelmiston rakennetta ja muutoksista voi tulla haastavia

toteuttaa. Iteratiivinen ohjelmistokehitysmenetelmä korjaa monia vesiputousmenetelmän epäkohtia, mutta se on vielä vahvasti suunnitelmavetoinen ja olettaa, että ohjelmistotuotanto on jossain määrin kontrolloitavissa oleva prosessi. (Luukkainen, 2021-a)

2.2.3 Scrum on yksi ketterä ohjelmistokehitysmenetelmä

Scrum on yksi eniten käytetty ketterän ohjelmistokehityksen menetelmistä (Luukkainen, 2021-a). Scrumin kehittäjät määrittelevät Scrumin olevan menetelmäkehys, jonka avulla monimutkaisten tuotteiden kehitystä voidaan hallita siten, että asiakkaille saadaan korkein mahdollinen asiakasarvo. Menetelmän kehittäjien kirjoittama dokumentti korostaa, ettei kyseessä ole prosessi tai tekniikka, joka yksistään antaa riittävän ohjeistuksen tekemiselle, vaan kyseessä on menetelmäkehys. Menetelmäkehys antaa toteutukselle suuntaviivat, mutta mahdollistaa ja edellyttääkin muita menetelmiä ja tekniikoita, jotka tulee valita tapauskohtaisesti ohjelmistoprojekteissa. (Schwaber & Sutherland, 2020, s. 2)

Scrumin tärkein tavoite on tehdä ohjelmistokehityksen työskentelyn etenemisen suorituskyky näkyväksi ja mahdollistaa tuotteen ja työskentelytapojen jatkuvan parantamisen. Scrumin keskiössä ovat läpinäkyvyys, tarkkailu ja mukautuminen ohjelmistokehityksessä. (Schwaber & Sutherland, 2020, ss. 4–5)

Scrumissa ohjelmistokehitys tapahtuu 1–4 viikon iteraatioissa, joita kutsutaan sprinteiksi. Jokaisen sprintin alussa tiimi valitsee tuotteen kehitysjonosta eli työlistasta sprintin aikana toteutettavat vaatimukset, jotka ovat valmiiksi esikäsitellyjä vaatimusten esisuunnittelukokouksissa. Esisuunnittelukokouksia järjestetään vaatimuksille niin kauan, että kaikki tarvittavat lähtötiedot vaatimuksesta on saatu selville ja vaatimukset on pilkottu niin pieniin osiin, että yksittäisiä vaatimuksia pystytään toteuttamaan yhden sprintin aikana. (Luukkainen, 2021-a)

Ohjelmoitavien tehtävien työlistasta käytetään englannin kielistä termiä product backlog. Työlistalla tarkoitetaan scrum-ohjelmistokehitysmenetelmässä, että projektin tehtävät on priorisoitu tärkeysjärjestykseen valmiiksi tekijöille. Kehitysjonon priorisoinnista vastaa tuotteen omistaja eli product owner. (Luukkainen, 2021-a)

Ennen kuin aloitetaan uusi sprintti, järjestetään sprintin suunnittelupalaveri. Tärkeimpänä aiheena on selvittää mitä sprintin aikana tehdään. Tuotteen omistaja esittelee tuotteen kehitysjonon kärkeen priorisoituja vaatimuksia kehitystiimille. Kehitystiimin tulee arvioida, onko vaatimukselle tehty riittävästi esisuunnittelua ja kuvattu haluttu ominaisuus niin, että sen voi vain ymmärtää yksiselitteisesti ja näin ollen sen toteutus voidaan aloittaa.

Vaatimusten valinnan lisäksi esisuunnittelussa asetetaan sprintille yhteisesti hyväksytty tavoite, joka on yksittäisiä vaatimuksia yleisempi ilmoitus muille sidosryhmille mitä seuraavassa sprintissä on tarkoitus tehdä. Toisena aiheena suunnittelupalaverissa on selvittää, miten sprintin tavoitteet saavutetaan. Tässä varmistetaan, että tehtäviä on realistinen määrä seuraavaan sprinttiin ja niiden testaustalle on myös varattu riittävästi aikaa ennen tuotoksien julkaisua. Sprintin suunnittelu palaverin lopputuloksena käynnistetään yleensä uusi sprintti. (Luukkainen, 2021-a)

Sprintin lopussa järjestetään katselmointitilaisuus, josta käytetään nimitystä sprint review. Katselmointi on tilaisuus, jonka aikana kehitystiimi esittelee käynnissä olevan sprintin tuotoksia kaikille sidosryhmille, jotka ovat kyseisestä kehityksen kohteena olevasta tuotteesta kiinnostuneita. Katselmoinnin tarkoitus on esittää itse toteutettua ohjelmistoa oikeassa ympäristössä tai kehitysympäristön avulla. Katselmoinnin jälkeen kehitystiimi järjestää retrospektiivin, jonka tarkoituksena on tarkastella kehitystiimin omaa työskentelyä, sekä pohtia mennyttä sprinttiä kahden kysymyksen avulla mikä meni hyvin ja missä asioissa on parantamisen varaa. Retrospektiivissä mietitään ratkaisuja haasteisiin, joita pyritään ratkaisemaan tulevissa sprinteissä. (Luukkainen, 2021-a)

Kuva 6. Scrumin yksittäisen sprintin prosessikaavio mukaillen teoriaa



2.3 Vaatimusten priorisointimenetelmistä

Projektin kolme peruspilaria ovat aikataulu, kustannukset ja sisältö. Peruspilarit ovat kytköksissä toisiinsa, sillä jos esimerkiksi aikataulu pettää, yleensä kustannukset kasvavat myös. Uuden tuotteen kehitysprosessin perustana ovat vaatimukset, jotka määrittelevät projektiin liittyvien sidosryhmien edustajat eli käyttäjien, asiakkaiden ja suunnittelijoiden vaatimukset tulevalle järjestelmälle ja mitä järjestelmän tulee tehdä tyydyttääkseen nämä tarpeet. Vaatimusten pohjalta arvioidaan myös mahdollista aikataulua ja resurssitarpeita. Vaatimusten toteutusta suunniteltaessa voidaan havaita esimerkiksi, että tietyn vaatimuksen toteutus ei onnistu saatavilla olevilla resursseilla tai vaaditussa aikataulussa. Tällöin voidaan joutua harkitsemaan toteutettavan projektin laajuutta tai päädytään julkaisemaan tuotteen ominaisuuksia vaiheittain. Hyvällä muutoshallinnalla voidaan projektin sisältöä hallitusti rajata niin, että aikataulu tai kustannukset eivät ylitä. Tämän vuoksi on ensiarvoisen tärkeää uuden projektin alussa määritellä, mikä projektissa on tärkeintä: tuotteen julkaiseminen aluksi sovitussa tavoite aikataulussa, budjetissa pysyminen vai se, että projektin sovitusta sisällöstä ei tingitä. Tehokas vaatimusmäärittely ja priorisointi auttaa yrityksiä hallitsemaan riskejä jo mahdollisimman aikaisessa vaiheessa projektin suunnittelua, sekä helpottaa yrityksen kykyä selviytyä koko ajan lisääntyvästä monimutkaisuudesta ohjelmistokehityksessä. (Juvonen, 2018)

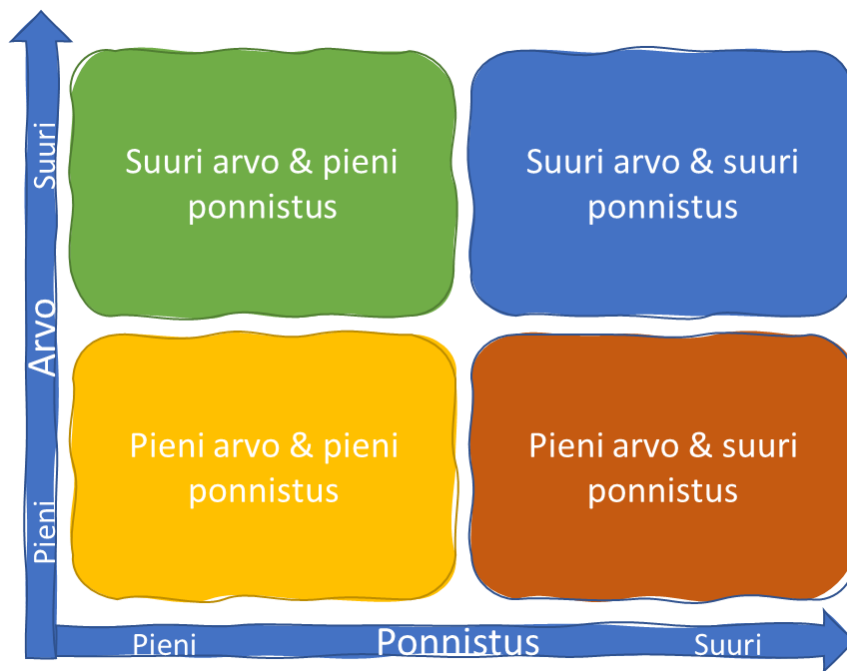
Firesmith (2004) kirjoittaa kolumnissaan vaatimusten priorisoinnin tärkeydestä suurissa ja monimutkaisissa projekteissa. Suurissa toteutuksissa voi olla valtavan paljon yksittäisiä vaatimuksia. Asiakkaat ja muut sidosryhmät odottavat usein projektin toteuttavan ne kaikki. Tällaiset projektit eivät voi välttyä haasteilta ja samalla vastata kaikkien tuotetta käyttävien sidosryhmien edustajien odotuksiin. Siksi tarvitaan priorisointia, kaikki vaatimukset eivät ole yhtä tärkeitä, tavoitteet voivat vaihdella, näkökulmat voivat muuttua ympäristön mukaan, jossa sidosryhmien edustajat työskentelevät. Sidosryhmien tavoitteet eivät ole samat tai yksittäisillä henkilöillä voi olla vaikeuksia ymmärtää toisiaan. Vaatimukset ovat luonteeltaan pakollisia. Tämä saa jotkin sidosryhmät uskomaan siihen, että kaikilla vaatimuksilla tulee olla sama korkein prioriteetti. Tärkeää on muistaa vaikka kaikki vaatimukset, olisivat pakollisia tietyllä ajanhetkellä projektissa, ne eivät voi olla täysin samanarvoisia. Jos liian pieni osa vaatimuksista priorisoidaan, menetetään projektissa priorisoinnin tuomat edut. Vaatimukset

voidaan priorisoida erilaisten menetelmien avulla, voi olla useampi menetelmä, jotka liittyvät toisiinsa tai priorisoivat vastakkaisten seikkojen perusteella. Yksi yksinkertainen tapa projektissa, jossa on tiukka aikataulu, voidaan vaatimukset priorisoida tietyn sidosryhmän mukaisesti, jossa toteutetaan heidän vaatimuksensa ensin ennen kuin aloitetaan toteuttamaan muiden omia.

2.3.1 Value vs. effort-priorisointimenetelmä

Value vs. effort on priorisointimenetelmä, jota käytetään ominaisuuksien priorisoimiseen ja tehokkaan tiekartan laatimiseen projektin toteutukselle, menetelmä toimii hyvin osana lean-metodologiaa. Value vs. effort-priorisointimenetelmä onkin suunnattu enemmän tuotehallinnalle sekä projektipäälliköille, joiden pitää saada suuren projektin vaatimukset priorisoitua optimaalisella tavalla. Priorisointimenetelmässä projektiryhmän tai valittujen sidosryhmien jäseniä pyydetään määrittämään kullekin ominaisuudelle arvo (esim. tulopotentiaali) ja vaadittu ponnistus (esim. työmääräarvio). Tämän perusteella projektiryhmä ja projektinjohto näkee nopeasti, mitkä ominaisuudet ovat arvokkaita ja vaativat vain vähän ponnistuksia, ja mitkä taas ovat vähän arvoa tuottavia, mutta paljon vaivaa vaativia ominaisuuksia, joista kannattaa luopua mahdollisesti kokonaan kyseisessä projektissa. Menetelmä on nopea ja yksinkertainen, koska siinä on ainoastaan kaksi erilaista muuttujaa: arvo ja vaadittu ponnistus, prioriteetti saadaan jakamalla arvo vaaditulla ponnistuksella. Prioriteetit jaetaan 4 eri luokkaan. (Andreev, 2022; Productplan, 2022)

Kuva 7. Mukailien Productplan nelikenttämallia. (Productplan, 2022)



Arvokysymykset määrittää tuotepäällikkö tai tuoteomistaja. Kysymysten määrittämiseen ei ole tiettyä kaavaa. Alla yleisimpiä tekijöitä, joita selvitetään arvomuuttujaan (Andreev, 2022):

- Asiakkaan hyöty tästä ominaisuudesta
- Asiakkaiden tyytyväisyys ja sitoutuminen tuotteeseen tämän ominaisuuden vuoksi
- Kuinka suureen osaan asiakkaista tämä ominaisuus vaikuttaa. (Andreev, 2022)

Vaadittu ponnistus: Matriisin toinen muuttuja menetelmässä on ponnistus. Projektitiimi arvioi ominaisuuden toteutuksen kustannukset ja arvioi tarvittavan työmäärän vaatimuksen toteuttamiseksi. Jotkut käyttävät toisena muuttujana monimutkaisuutta ponnistuksen sijaan. On tärkeää huomata, että monimutkaisuuden käyttö muuttujana saattaa olla harhaanjohtavaa, koska monimutkaisuus on yksi tekijöistä, jotka on otettava huomioon arvioitaessa ponnistuksen vaivaa halutun ominaisuuden toteuttamiseksi. Mutta se ei ole ainoa tekijä, on myös muita osatekijöitä, kuten työmäärä, toteutuksen epävarmuus ja siihen liittyvät riskit. Haluttuun ominaisuuteen vaadittavan työmäärän voi arvioida päivissä tai sanallisesti luokiteltuna esim. pieni tai suuri ja toteutuksen epävarmuuden tiedosta onko vaatimuksen toteuttajaksi useita vaihtoehtoja vai vain yksi vaihtoehto. (Andreev, 2022)

Kun vaatimukset on pisteytetty ensin niiden arvon ja toiseksi niiden toteuttamiseen tarvittavan vaivan perusteella, lopputulokseksi saadaan prioriteetti vaatimukselle, kun yhteenlaskettu arvo -muuttuja jaetaan yhteenlasketulla ponnistus -muuttujalla. Prioriteetit jaetaan neljään luokitukseen. (Productplan, 2022)

1. Suuri arvo ja pieni ponnistus -prioriteettia kutsutaan usein "Quick wins" prioriteetiksi. Tämä luokitus sisältää houkuttelevia vaatimuksia, jotka vaativat vain vähän vaivaa ja joilla voi olla suuri vaikutus tuotteeseen. Tähän luokkaan projektin täytyy keskittyä ensin, koska toteuttamalla nämä vaatimukset on loistava tapa voittaa asiakkaita nopeasti puolelleen sekä nämä vaatimukset kertovat miksi käynnissä olevaa projektia tehdään. Tuotteesta tai projektista riippuen ei ehkä tähän kategoriaan päädy monia ominaisuuksia, mutta ne vaatimukset, jotka saavat tämän prioriteetin, kannattaa toteuttaa projektissa. (Productplan, 2022)
2. Suuri arvo ja suuri ponnistus -prioriteettia kutsutaan myös nimellä "Big Bets" tai "Strategiset aloitteet". Näillä aloitteilla tai ominaisuuksilla on suuri arvo, mutta niiden toteuttaminen vaatii myös paljon vaivaa tai toteutus sisältää epävarmuutta. Suuret uudistukset ja uudet suuret ideat kuuluvat usein tähän prioriteettiluokitukseen. Yleisesti nämä vaatimukset ovat ehdottomasti tekemisen arvoisia ja kannattaa priorisoida toteutuksen kannalta "Quick wins" vaatimusten jälkeen toteutettavaksi. Niissä vaatimuksissa missä vaatimuksen toteutukseen liittyy paljon epävarmuutta, on paras lähestymistapa olla valikoiva ja arvioida uudelleen, kun vaatimuksen toteutukseen tarvittava ponnistus tarkentuu projektin edetessä. Suunnittele huolellisesti ja toteuta tehokkaasti, jos mahdollista, jaa nämä suuret työmäärät vähemmän monimutkaisiksi kokonaisuuksiksi ja tarkenna vaatimusten määrittelyjä aina riittävästi ennen toteutusta. (Productplan, 2022)
3. Pieni arvo ja pieni ponnistus -prioriteetti saattaa edustaa projektin yleisintä osaa. "Nice to have" ominaisuuksia, tai ominaisuuksia, joita asiakkaat odottavat olevan vakiona tuotteessa. Näitä vaatimuksia ei suositella toteutettavaksi ensin projektissa, vaan niitä voidaan jättää odottamaan projektin valmistumista, jolloin voidaan vielä täydentää tuotetta tai jättää tuleviin päivitysversioihin julkaistavaksi. Koska nämä vaatimukset

edustavat mahdollisuutta saada pieniä voittoja kuluttamatta paljoa resursseja, niin näitä vaatimuksia voidaan toteuttaa, kun projektitiimillä on vapaa-aikaa. (Productplan, 2022)

4. Pieni arvo ja suuri ponnistus -prioriteetilla luokitelluista vaatimuksista sanotaan usein, etteivät ne ole toteuttamisen arvoisia. Ellei projektissa ole erityistä syytä käyttää aikaa sellaisen vaatimuksen tekemiseen, joka ei tarjoa paljon liikearvoa sekä on haastava toteuttaa, niin tällaiset vaatimukset pitäisi jättää toteuttamatta projektissa. (Productplan, 2022)

Menetelmän heikkoutena voidaan pitää subjektiivista luonnetta. Tällä tarkoitetaan priorisointimenetelmää käyttävien omakohtaista tulkintaa tai käsitystä ja usein myös puolueellista näkemystä, viitaten käsiteltävien vaatimusten näkökulmaan. Koska tarkkaa pisteytyskaavaa ei ole olemassa, priorisointimenetelmä on edelleen melko avoin keskustelulle ja sitä voidaan sen vuoksi myös pitää subjektiivisena. Tästä syystä tällä menetelmällä priorisoitaessa on tärkeää tehdä siitä läpinäkyvää kaikille sidosryhmille ja selittää sekä dokumentoida selkeästi, miksi vaatimus tai ominaisuus määritellään niin arvokkaaksi tai suureksi ponnistukseksi. (Andreev, 2022)

Hyvänä puolena Value vs. effort-priorisointimenetelmässä on, ettei se vie paljon aikaa tai vaadi monimutkaisia laskelmia. Menetelmän voi räätälöidä yrityksen tai projektin tarpeisiin sopivaksi ja menetelmä on määrällisesti mitattavissa. Menetelmässä kunkin vaatimuksen potentiaalinen hyöty punnitaan sen vaatimiin ponnisteluihin nähden ja pyritään tunnistamaan, mitä voidaan realistisesti tarjota nykyisillä ja käytettävissä olevilla resursseilla. Tämä viitekehys voi olla hyödyllinen muissakin priorisointiskenaarioissa. Tässä muutama esimerkki siitä, missä Value vs. effort-menetelmä voi olla erityisen hyödyllinen. (Productplan, 2022):

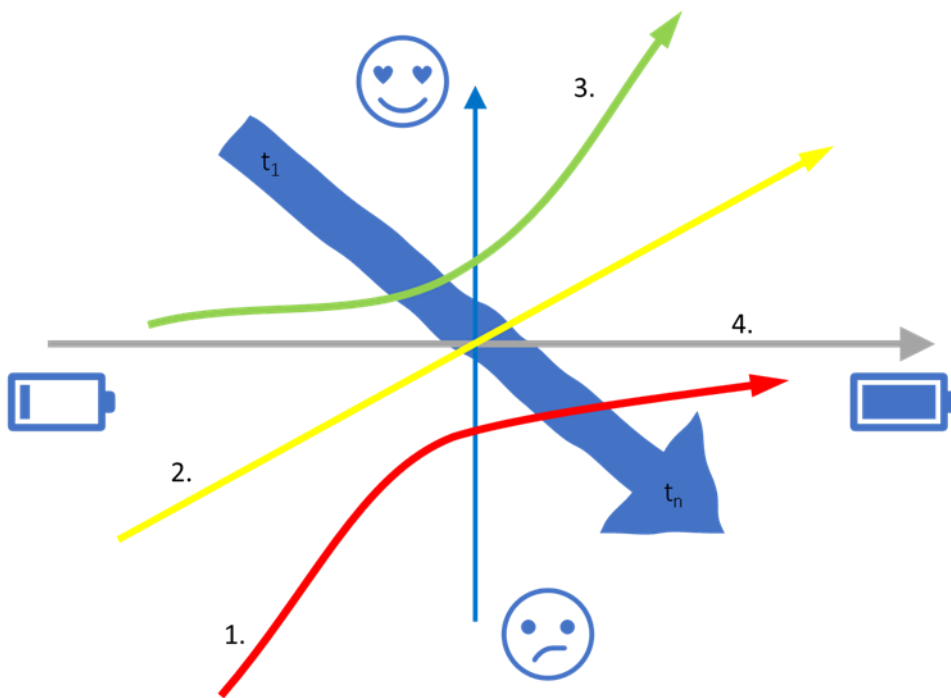
- Kun tehdään uutta tuotetta
- Priorisoinnille halutaan yksinkertainen ja nopea käsittely
- Käytännöllinen, kun projektin julkaisupäivä on etukäteen lukittu
- Projektissa on käytössä rajalliset resurssit. (Productplan, 2022)

Parhaimpia syitä käyttää Value vs. effort-priorisointimenetelmää on ensisijaisesti auttaa projektia tunnistamaan nämä vaatimukset, jotka tuottavat vain vähän arvoa ja ovat vaativia toteuttaa. Tällä menetelmällä voidaan varmistaa, ettei projekti tuhlaa tärkeää kehitysaikaa sekä resursseja. Menetelmä on myös tehokas tunnistamaan nopeasti kaikki paljon arvoa tuottavat vaatimukset, jotka eivät ole vaativia toteuttaa ja perustelevat siten koko projektin olemassaolon. Mallin käyttö voi olla hyödyllinen strategia projektitiimeille, jotka yrittävät muuttaa pitkän listan ehdotetuista ominaisuuksista, parannuksista ja muista kohteista strategisesti järkeväksi prioriteettilistaksi. (Productplan, 2022)

2.3.2 Kano-malli

Japanilainen Noriaki Kano julkaisi vuonna 1984 tutkimuksestaan artikkelin Attractive Quality and Must-be Quality, joka sisälsi joukon ideoita ja tekniikoita, jotka auttavat määrittämään asiakastytyvyyden tuotteen ominaisuuksiin perustuen. Tätä priorisointi menetelmää kutsutaan Kano-malliksi. (Zacarias, n.d.)

Kuva 8. Mukailleen Noriaki Kanon Kano-mallia



Ominaisuudet voidaan mallissa jakaa neljään kategoriaan.

1. Tuotteen minimiominaisuudet. ominaisuudet ja asiat, joiden asiakas olettaa olevan tuotteessa. Näiden asioiden tuottaminen asiakkaalle ei riitä positiivisen kokemuksen muodostumiseen tuotteesta, mutta jos jokin puuttuu, asiakas on tyytymätön. Näitä kutsutaan pakollisiksi ominaisuuksiksi ja tästä kategoriasta käytetään yleisesti nimitystä Must be.
2. Enemmän on parempi -ominaisuudet. Tähän kategoriaan kuuluu ominaisuudet mitä asiakas sanoo tarvitsevänsä ja arvostavansa. Mitä enemmän näitä ominaisuuksia voidaan tuotteesta asiakkaalle tarjota, sitä tyytyväisempi hän on. Tästä kategoriasta

käytetään joskus virheellisesti pelkästään nimitystä tehokkuusominaisuudet, jolla tarkoitetaan, että näihin ominaisuuksiin liittyy jokin mitattava suure ja sen avulla asiakkaat vertailevat tuotteita keskenään ja arvottavat asiakastyytyvyyttä tuotteen hintaan nähden. Tehokkuusominaisuudet ovat vain yksi osa tämän kategorian kokonaisuutta ja ylätasolla tähän kategoriaan kuuluvat ne ominaisuudet, mitä asiakas osaa vaatia ja nimetä tuotteeseen. Tästä kategoriasta käytetään yleisesti nimitystä Performance.

3. Houkuttelevat ominaisuudet. On olemassa odottamattomia ominaisuuksia, joita asiakas ei osaa suoraan vaatia, mutta ne aiheuttavat positiivisen reaktion. Näitä ominaisuuksia kutsutaan usein vau-tekijöiksi ja tästä kategoriasta käytetään yleisesti nimeä Attractive.
4. Ei väliä -ominaisuudet. On myös ominaisuuksia ja vaatimuksia, joiden läsnäolo tai puuttuminen ei vaikuta todelliseen reaktioon tuotetta kohtaan. Tästä kategoriasta käytetään yleisesti nimitystä Indifferent. (Zacarias, n.d.)

Kuvassa 8. näkyvä sininen nuoli kuvaa Kano-mallissa ajan mittaan tapahtuvaa ilmiötä, kun ominaisuudet vaihtavat tilaansa siten, että houkuttelevista ominaisuuksista tulee, Enemmän on parempi -vaatimuksia ja vastaavasti näistä ominaisuuksista tulee tuotteen minimivaatimuksia. Esimerkkinä voisi mainita auton turvatyynyjärjestelmät, jotka ovat aikojen kuluessa muuttuneet houkuttelevista ominaisuuksista minimivaatimuksiksi uusissa auto malleissa. Kano-malliin kuuluu myös kategoriat vastakkainen ja skeptinen, näitä kuitenkin harvemmin hyödynnetään analyysissä. (Hietikko, 2021-b)

Asiakkaan tuntemusten määrittämiseksi pohjatiedot saadaan Kano-matriisi kyselyn avulla, joka määrittää ominaisuus -kategorian vaatimukselle. Kysely koostuu kahdesta kysymyksestä erikseen jokaiselle ominaisuudelle.

1. Miltä asiakkaasta tuntuu, jos heillä on kyseinen ominaisuus?
2. Miltä heistä tuntuu, jos heillä ei olisi tätä ominaisuutta?

Tätä kutsutaan toimiva ja toimimaton -kyselymuodoksi, kumpaankin vastaukseen ainoat mahdolliset vastaus vaihtoehdot ovat: 1) Pidän siitä, 2) Odotan sitä, 3) Minulle ei ole väliä, 4) Voin hyväksyä sen ja 5) En pidä siitä. (Zacarias, n.d.)

Taulukko antaa vastausten perusteella vaatimukselle oikean kategorian. Taulukossa 1. kategorioita merkitään seuraavilla kirjaimilla:

- A on houkuttelevat ominaisuudet luokitus
- O on enemmän on parempi ominaisuudet
- M on minimiominaisuudet
- I on ei väliä ominaisuudet
- R on vastakkainen luokitus
- S on skeptiset ominaisuudet

Taulukko 1. Mukailen Kano-matriisia (Hietikko, 2021-b)

Kirjaa tähän ominaisuus:		Mikä on asiakkaan reaktio, jos kyseinen tuoteominaisuus puuttuu?				
Miltä asiakkaasta tuntuu jos tuotteessa on kyseinen ominaisuus?		Mielissään	Edellyttää	Neutraali	Voi elää	Pettynyt
	Mielissään	S	A	A	A	O
	Edellyttää	R	I	I	I	M
	Neutraali	R	I	I	I	M
	Voi elää	R	I	I	I	M
	Pettynyt	R	R	R	R	S

Taulukon kolme ylintä riviä ja kolme oikeanpuolimmaista saraketta ovat yleensä taulukon merkittävimmät solut. Näitä soluja voidaan arvottaa tuotekehityksessä esimerkiksi seuraavasti.

- Kyselyn tuloksena luokituksen O saava Enemmän on parempi -vaatimus, on optimaalinen tuotteelle ja tulee ottaa erityisen vakavasti huomioon tuotekehityksessä.

- Kyselyn tuloksena luokituksen A saa Houkutteleva ominaisuus -vaatimus voi saada aikaan erottautuvuutta kilpailevista tuotteista. Ominaisuuden toteuttamiseen kannattaa kiinnittää huomiota tuotekehityksessä, mutta sen aiheuttamia mahdollisia kustannuksia on tarkasteltava kriittisesti.
- Kyselyn tuloksena luokituksen M saava ominaisuus on tuotteelle minimivaatimus. Siihen tulee kiinnittää huomiota, mutta ei ole tarpeen kohdistaa erityisiä resursseja.
- Kyselyn tuloksena luokituksen I saava vaatimus, koska sillä ominaisuudella ei ole asiakkaalle väliä, niin se voidaan usein jättää toteuttamatta.

Kun vastaajia kyselyyn on useita, tavallisin tapa selvittää ominaisuuden luokka vastausten perusteella on moodi. Moodilla tarkoitetaan sitä, että luokitus valitaan ominaisuudelle sen perusteella mikä luokka on saanut eniten kannatusta kyselyn vastaajien kesken. (Hietikko, 2021-b)

Vaikka tämä menetelmä antaa tuotekehitykselle tietoa asiakkaiden tarpeista ja luokittelee vaatimukset, liittyy sen käyttöön haasteita. Menetelmä ei ota vaatimusten toteutukseen riittävästi kantaa, toteutus vaati oman käsittelyn tai menetelmän ja näiden yhteistuloksena saadaan lopulliset prioriteetit vaatimuksille. Suurimmat haasteet liittyvät kyselyn toteutukseen ja vastaajien määrään. Jos luokiteltavia vaatimuksia on paljon, kasvaa kyselyyn vastaamiseen tarvittava aika. Tämä voi aiheuttaa sen, että vastaajat eivät jaksu keskittyä ja suorittaa kyselyä loppuun asti. Vastaajien on myös ymmärrettävä kyselystä selkeästi, mitä kyselyn jokainen vaatimus tarkoittaa ja miten se vaikuttaa tuotteeseen. Suurempi haaste on riittävä vastaajien lukumäärä. Sitä suurempi mahdollisuus on virheellisiin luokitus tulkintoihin eri ominaisuuksille, mitä vähemmän vastaajia on kyselyssä. (Hietikko, 2021-a)

3 Tutkimuksen empiirinen osuus

Tässä tutkimuksen empiirisessä osuudessa kuvataan aluksi tutkimusmenetelmät ja aineiston keruun toteutus. Tarkoituksena on kertoa, miten koko empiirinen osuus tutkimuksesta on toteutettu. Tutkimuksen kulku kappaleessa kuvataan empiirisen tutkimuksen etenemisen

lisäksi, myös lyhyesti kohdeorganisaatiota ja sen asiantuntijoita sekä perustellaan ja arvioidaan tehtyjä tutkimuksen valintoja. Tutkimuksen lähtökohdaksi luotu konstruktio vaatimustenhallinnasta esitellään ennen tutkimustulosten käsittelyä.

Koska tutkimustulokset luvussa esitellään sekä kyselyjen tuloksia että niistä tehtyjä johtopäätöksiä, on suoraan kyselyistä saadut tulokset ja toisaalta niistä johdetut johtopäätökset eroteltava toisistaan. Suoraan kyselyistä saatujen tuloksien kohdalla mainitaan, että ne ovat tulleet vastauksena kyselyistä. Johtopäätökset erotetaan kyselyn vastauksista niin, että vastauskohdissa esitellään ensin kyselyissä ilmenneet asiat ja niiden jälkeen mainitaan alkavan johtopäätökset kyselyn käsiteltävästä asiasta.

Kirjallisuuskatsauksen perusteella on saatu viitteitä siitä, että vaatimusten priorisoinnilla on merkitystä ohjelmistoprojektien hallinnassa. Sen vuoksi kyselytutkimuksessa tutkitaan vaatimusten hallintaa priorisoinnin kautta ja selvitetään, kuinka vaatimukset tulisi priorisoida yhteismitallisesti ja niin, että niiden tulokset ovat kaikille sidosryhmille läpinäkyviä.

3.1 Tutkimusmenetelmät ja aineistonkeruu

Tutkimus toteutettiin perinteisen tutkimuksen periaatteita noudattaen. Siinä käytettiin sekundäärisenä tietolähteenä kirjallisuuskatsausta. Primäärinä tietolähteenä on kvalitatiivinen haastattelututkimus. Tutkimusstrategia, jolla tutkimusongelmiin pyritään löytämään vastaukset, on kvalitatiivinen tutkimusote, jossa hyödynnetään eri kvalitatiivisen tutkimuksen metodeja. Metodit ja aineistonkeruun suunnitelmana on konstruktiiivinen tutkimusote sekä Delfoi-menetelmä.

Tutkimuskysymykset ja lähtökohta tutkimukselle luotiin kirjallisuuskatsauksen avulla käyttäen konstruktiiivista tutkimusotetta. Konstruktiiivinen tutkimusote on kekseliäitä soveltavia tutkimuksia tuottava metodologia, jolla pyritään ratkaisemaan reaalimaailman ongelmia ja tällä tavoin tuottamaan uutta sisältöä olemassa olevaan sille tieteenalalle, jossa sitä sovelletaan. Tämän tutkimusotteen ydinkäsite on uusi konstruktio, joka on abstrakti käsite, jolla on suuri määrä mahdollisia toteutumia. Kaikki ihmisen luomat mallit, kaaviot, suunnitelmat, organisaatorakenteet, kaupalliset tuotteet ja tietojärjestelmämallit, ovat

konstruktioita. Niille on tunnusomaista se, että ne eivät ole löydettyjä, vaan ne keksitään ja kehitetään. Kehittämällä konstruktion, joka poikkeaa kaikesta jo olemassa olevasta, luodaan jotain aivan uutta. Konstruktiivisessa menetelmässä on aina päätettävä, millaista lopputulosta halutaan rakentaa. Tuloksen ei tarvitse aina olla konkreettinen tuote, vaan se voi olla myös prototyyppi tai pelkkä suunnitelma. Tärkeää on, että konstruktioita täytyy kuitenkin aina voida arvioida. (Lukka, 2001)

Tutkimuksen paradigma on Delfoi-menetelmä, jolla myös empiiriset haastattelututkimuskierrokset tehtiin. Delfoi-menetelmä soveltuu käyttöön silloin, kun on tarkoitus saavuttaa ryhmän konsensus eli yksimielisyys jostakin asiasta. Jos siihen pyritään avoimella keskustelulla eri sidosryhmien välillä eli paneelilla, silloin tulos helposti vaarantuu tai vinoutuu mahdollisesti eri syistä. Tutkimusmenetelmänä pyritään Delfoi-menetelmällä saamaan selkoa joistakin suuremmasta kokonaisuudesta, tärkeistä ja hämärän peitossa olevista asioista. Delfoi-menetelmä on tyypillisesti alku tulevaisuuteen suuntautuneelle prosessille, jossa tavoitellaan yhteisiä päämääriä. Delfoi-menetelmällä saadut tulokset eivät ole lopullinen tulos, vaan prosessin edetessä kehittyvä malli. (Anttila, 2014)

Delfoi-menetelmässä kuullaan nykyisin tyypillisesti kahdella tai kolmella kierroksella 15–40 asiantuntijaa. Keskeinen piirre menetelmässä on vastauksien antaminen ilman nimiä eli anonymisti. Päinvastoin kuin gallupeissa, mielipiteitä ei kerätä vain analysoitavaksi, vaan vastaustieto kierrätetään takaisin asiantuntijoille. Palautetiedon avulla vastaajia ohjataan perustelemaan valintojaan. Tiedon muodostus etenee kierroksittain niin, että edellinen kyselykierros muodostaa pohjan seuraavalle. Delfoi-tutkimus käynnistetään usein jostain ajankohtaisesta riita- tai kiistakysymyksestä. (Kuusi, 1999)

3.2 Tutkimuksen kulku

Opinnäytetyön aineisto kerättiin tekemällä kaksi kierrosta kyselytutkimuksia Delfoi-menetelmällä Kempin asiantuntijoille. Kysely lähetettiin yhteensä 18 henkilölle. Kyselyyn osallistuneet työskentelevät teknologiapäälliköinä, päänsinööreinä tai projektipäälliköinä tuotekehityksen eri teknologiaosastoilla. Kysely lähetettiin myös usealle tuotepäällikölle tuotehallintaosastolle, sekä kysyttiin tuotekehityksen johdolta ja liiketoimintayksikön

vetäjältä. Kyselyyn osallistuneet vastaavat ja päättävät nykytoimintamallissa vaatimusten keräämisestä, toteuttamispäätöksistä ja niiden priorisoinnista eri projekteissa. Tavoitteena kyselyissä oli saada mahdollisimman laaja näkemys Kempin tuotekehityksen ja tuotehallinnan toimintatavoista vaatimustenhallintaan liittyen uusien tuotteiden kehitysprojekteissa.

Kyselytutkimus toteutettiin lähettämällä linkki sähköpostilla erilliseen Microsoft Forms työkalulla toteutettuun kyselyyn. Ensimmäisen kyselykierroksen sähköpostiviestissä kerrottiin tutkimuksen taustoista ja sen lisäksi liitteenä oli erillinen Power Point-esitys, missä oli tarkemmin kerrottu tutkimusmenetelmistä ja siellä oli kuvattu tutkimuksen lähtökohdaksi luotu konstruktio mahdollisesta uudesta toimintamallista, jonka perusteella kysyttiin kysymykset ensimmäisellä kierroksella tutkimuksessa.

Kysely laadittiin tutkittavan ilmiön ympärille ja tutkimuksen kiista kysymykseksi oli asetettu: Miten Kempillä tulisi tuotteiden vaatimukset käsitellä ja priorisoida? Pääosa kysymyksistä rakennettiin niin, että vastauksia on helppo vertailla keskenään. Ensimmäisen kierroksen kysymykset alkoivat monivalintakysymyksillä, joita oli mahdollisuus perustella tai kommentoida jokaista kysymystä erikseen. Monivalintakysymyksillä kartoitettiin vastaajien mielipidettä nykytilanteesta. Monivalintakysymysten asteikko oli 5-portainen, missä vaihtoehdot olivat asteikoilla 1–5, jossa 1 tarkoitti "Välttävästi" ja 5 tarkoitti "Erinomaisesti". Seuraavaksi oli avoimia kysymyksiä, joilla kartoitettiin vastaajien mielipidettä, miten asiat pitäisi tehdä vastaajan mielestä. Lopuksi kysyttiin kyllä tai ei kysymyksiä, joilla haettiin vahvistusta konstruktiossa esitettyihin asioihin. Kyselyn lopuksi vastaajalla oli mahdollisuus antaa palautetta tai kommentoida, jos jokin asia jäi vastaajan mielestä puuttumaan.

Toisen kysely kierroksen sähköpostin liitteessä toimitettiin kaikille ensimmäisen kierroksen vastaukset sekä esimerkkityökalu vaatimusten priorisointiin. Toinen kysely kierros tehtiin ensimmäisen kierroksen pohjalta, jossa kysyttiin ensimmäisellä kierroksella vastaajien ehdottamia asioita. Vastausvaihtoehdot olivat pääosin kyllä ja ei tyyppisiä. Ainoastaan liitteenä toimitetusta vaatimusten priorisointityökalusta oli mahdollista antaa avoin vastaus sekä kyselyn lopussa oli mahdollista antaa avointa palautetta tai kommentoida jotain kyselyn kohtaa. Toinen kysely kierros oli strukturoidumpi kysely valmiine vastausvaihtoehtoineen,

koska tarkoitus oli nähdä missä kysymyksissä oli asiantuntijoiden välillä konsensus saavutettu.

Konstruktiiivinen tutkimusote soveltui hyvin tähän työhön lähtökohdaksi, koska työssä kehitetään uusia ratkaisuja ohjelmistokehityksen vaatimusten hallinnan ongelmiin, käyttäen tukena jo olemassa olevaa teoriaa. Pragmaattiseen toiminnan kehittämiseen tähtäävä lopputulos oli hyvä aloittaa kirjallisuuskatsaukseen perustuvalla konstruktiolla. Koska lähtökohdaksi kehitetyn konstruktion oli tarkoitus toimia asiantuntijoille keskustelun katalyyttina. Konstruktio aktivoi asiantuntijat ottamaan henkilökohtaisen kannan kysymyksiin ja herätti tarpeen myös perustella näkemyksiään.

Delfoi-menetelmä valikoitui empiiriseksi tutkimusmenetelmäksi, koska tutkimuskysymyksiä ei voida tutkia täsmällisillä analyttisillä menetelmillä. Eri asiantuntijoiden subjektiiviset arvioinnit ovat tutkimuskysymyksiä ratkaisun kannalta hyödyllisiä. Tutkimuskysymyksiä ongelma on kompleksinen ja yksittäiseltä asiantuntijalta puuttuu keinot ratkaista ongelma.

Tarkoitus oli saada asiantuntijajaneelin keskuudessa perusteluja omille näkemyksille ja keskustelua mahdollisista ratkaisuvaihtoehdoista, kuitenkin niin, että muiden osallistujien mielipiteisiin oli mahdollista vaikuttaa vain hyvillä argumenteilla kyselyn vastauksissa. Anonymiteetillä tutkimuksella haluttiin välttää tutkimuksen vinoutuminen yksittäisen asiantuntijan näkemyksiin mukaan, mikä olisi voinut tapahtua avoimella keskustelulla. Koska työssä pyritään saamaan selkoa suuremmasta kokonaisuudesta ja yksimielisyyden muutoksesta ohjelmistoprojektin vaatimustenhallinnan toimintamallista. Myös positiivisena asiana Delfoi-menetelmässä on tutkimuksen toteutus useammalla kierroksella, jossa vastaustieto kierrätetään takaisin asiantuntijoille, tämä auttaa vahvistamaan vastaajien asennetta tutkimuksen tärkeydestä ja mahdollisesti saattoi herättää uusia ajatuksia vastaajissa tutkimuksen kysymyksistä.

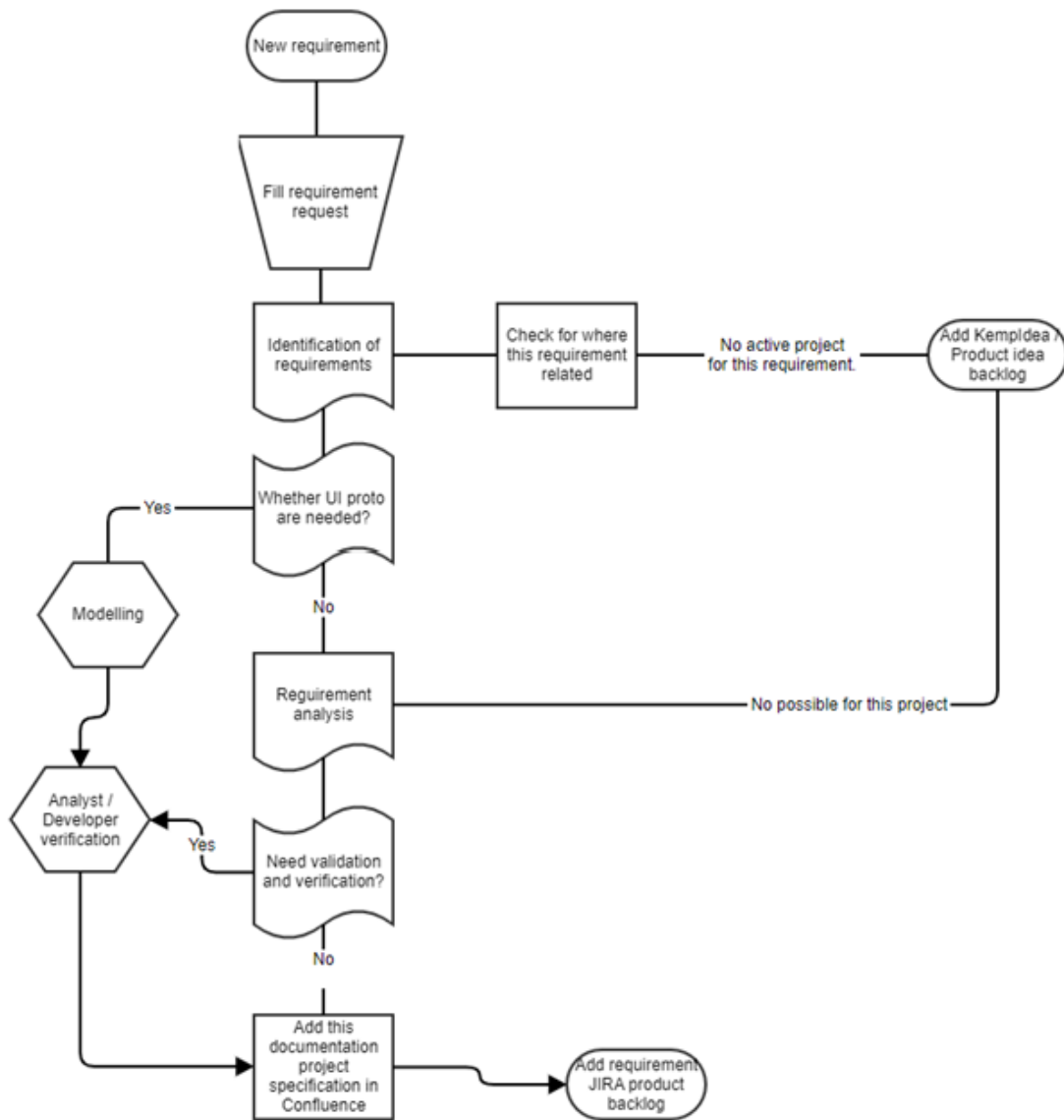
Delfoi-menetelmä soveltui hyvin tähän työhön, koska työssä koitetaan löytää konsensus nykytilanteen kehittämiseksi tunnistetuissa ohjelmistokehitykseen liittyvissä vaatimustenhallinnan haasteissa. Työn lopputulos on konstruktio Delfoi-menetelmällä saaduista tuloksista ja ehdotus toimintamallin kehittämiseksi.

3.3 Tutkimuksen lähtökohdaksi luotu konstruktio vaatimustenhallinnasta

Uusi toimintaehdotus konstruktio vaatimusten käsittelyyn luotiin ensimmäistä kyselykierrosta varten keskustelun katalyytiksi. Vuokaaviossa ja lyhyessä vaihekuvauksessa oli hyvin yleisellä tasolla muutosehdotuksia nykyiseen toiminta malliin. Ennen vuokaavion esittämistä Power Point-esityksessä esitettiin nykytilanteesta tutkijan omia havaintoja ja näkemyksiä.

- Vaatimukset nykytilanteessa tuotteisiin on aina pelkästään projektikohtaisia.
- Vaatimuksia ei nykytilanteessa priorisoida, toteutuksen osalta vaatimuksia käydään päällikkötasolla läpi ovatko ne teknisesti toteutettavissa, ja tähän ei ole vakiintuneita käytänteitä.
- Vaatimukset tulevat projekteille tuotepäällikön kautta.
- Yhtenä haasteena nykymallissa on vaatimuksia tai niiden sisältöä ei lyödä lukkoon siinä kohdassa kuin käytössä oleva lineaarinen projektitoimintamalli vaatisi.
- Priorisoinnin puuttuminen, koska paneeliteknologia hitsauslaitteissa ei rajoita muutospyyntöjä samalla tavalla kuin aikaisemmin. Esimerkiksi aikaisemmissa laitteissa yksi ohjelmoitsija ohjelmoi koko paneelin ja sen painikkeet. Jos käyttöpaneeliin haluttiin muutoksia tai uusia toimintoja, koko paneeli meni uusiksi. Nykyisissä graafisissa käyttöliittymissä ei ole tällaisia rajoitteita, mutta ohjelmistokehittäjiä voi olla 12 henkilöä käyttöliittymien monimutkaistuesssa.
- Ominaisuuksia ja vaatimuksia on joka projektissa enemmän kuin edellisessä projektissa. Tämä johtaa helposti siihen, että vaatimuksia tunnistetaan enemmän kuin mitä projektin aikataulu tai resurssit mahdollistavat.

Kuva 9. Lähtökohdaksi luotu konstruktio vaatimustenhallinnasta vuokaaviolla esitetynä



Power Point-esityksessä esitettiin vuokaavio kuvaaja ja kerrottiin sanallisesti vaiheista, joissa pyrittiin tekemään aikaisemmin esitettyihin nykytilanteen haasteisiin ratkaisuja yleisellä tasolla.

- Kuka tahansa voi täyttää Jira ohjelmistoalustalla uuden vaatimuksen esitietolomakkeen. Esitietolomakkeessa on vaatimuksen tunnistus mihin projektiin uusi vaatimus liittyy, jos valintaa ei tehdä tai ei osata määritellä vaatimus menee ideapankkiin.

- Ennen vaatimusten käsittelyä hankitaan tarvittavat lisätiedot esimerkiksi vaatimuksen toiminnasta käyttöliittymässä, toiminnosta mallikuvat tai suunnittelijoilta karkea työmääräarvio.
- Esitietojen jälkeen voidaan suorittaa vaatimusten käsittely ja priorisointi toimintamallin mukaisesti.
- Vaatimusten käsittelyn ja priorisoinnin jälkeen vaatimus laitetaan projektille toteutettavaksi prioriteetin mukaisesti tai odottamaan toteutuksen vuoroa.
- Vaatimus mikä on päätetty olla toteuttamatta, dokumentoidaan myös, koska se voidaan käsitellä tulevaisuudessa toisessa projektissa uudelleen tai siihen voidaan palata esimerkiksi siinä tapauksessa, jos joku kyselee myöhemmin vaatimuksen perään. Näin voidaan osoittaa perusteet, miksi kyseinen vaatimus on päätetty jättää toteuttamatta.

Tällä konstruktiolla pyrittiin aktivoimaan kyselytutkimukseen osallistuvat asiantuntijat pohtimaan ja perustelemaan näkemyksiään kysymyksestä, miten Kempillä tulisi tuotteiden vaatimukset käsitellä ja priorisoida?

3.4 Tutkimustulokset

Koska kyseessä on kehittävä toiminnan tutkimus, tutkimuksen tuloksissa on päädytty kuvaamaan asiantuntijoiden kuvaamia keinoja ja ehdotuksia toivottuun kehitykseen. Metodivariaatioksi on valittu Konsensus-Delfoi, jossa työn lopputuloksena tuotettavaan toimintamalliehdotukseen valitaan suoraan ehdotukset, joista asiantuntijapaneeli oli täysin samaa mieltä. Vaihtoehtoina tai erilaisina skenaariokuvauksina käsiteltiin ne ehdotukset mitkä ovat kyselyssä nousseet esille, mutta asiantuntijat eivät ole olleet asiasta täysin samaa mieltä. Menetelmä on valittu, koska työ on pyritty tekemään pragmaattisena toiminnan kehittämisenä, jossa pyritään tuottamaan välittömään tarpeeseen uusia toimenpide- sekä päätösehdotuksia. (Linturi, 2019)

Kuitenkin tulosten päättelymenetelmänä oli tutkimuksen aiheen perusteella abduktiivinen päättely, joka perustuu siihen, että uuden teorian muodostus on mahdollista vain silloin, kun havaintojen tekoon liittyy teoreettinen johtoajatus. Tutkija pyrkii todentamaan tämän

johtajatuksen aineistonsa avulla ja sen avulla havainnot voidaan keskittää joihinkin tiettyihin seikkoihin, joiden uskotaan tuottavan uusia näkemyksiä ja teoriaa kyseisestä ilmiöstä. (Anttila, 2014)

Kysyttäessä tutkimuksen aluksi, miten hyvin vaatimustenhallinta tällä hetkellä toimii tuotekehitysprojekteissa vastaajien mielestä, vastaukset jakautuivat ensimmäisellä kysely kierroksella välillä 1–3 keskiarvon ollessa 2.33. Toisella kierroksella tämä kysymys toistettiin ja keskiarvo oli silloin 2.78. Tällä kysymyksellä kartoitettiin vastaajien näkemystä nykytilanteesta yleisesti ja vaatimusten hallinnan toimivuudesta tuotekehitysprojekteissa Kempillä. Ensimmäisellä kierroksella oli mahdollista myös perustella avoimella vastauksella numeroarviota kysymyksestä. Perusteluiksi kirjattiin, että vastuualueet ovat epäselvät ja on epäselvyyttä kenen vastuulla mikäkin asia vaatimusten hallintaan liittyen on. Vastattiin myös, että systemaattinen menetelmä puuttuu ja lisäksi eri osapuolilla on erilaiset omat tavoitteet, joiden vuoksi voi syntyä jännitteitä.

Johtopäätöksenä yleisesti voi sanoa tuloksen toisen kierroksen jälkeenkin viittaavan siihen, että kehittämistoimenpiteitä tarvitaan tällä alueella ja kyseinen opinnäytetyötutkimus on aiheellinen. Ensimmäisen kyselykierroksen jälkeen johtopäätöksenä toiselle kyselykierrokselle ehdotettiin asiantuntijoille tilanteen parantamiseksi, että olisiko vaatimustenhallinnan kannalta projektin alussa hyvä tehdä sidosryhmäanalyysi? Sidoryhmäanalyysillä tarkoitetaan analyysiä sidosryhmien tunnistamiseen ja kommunikoimiseen sekä heidän tarpeiden ja motivaatiotekijöiden ymmärtämiseen. Vastausvaihtoehdoiksi annettiin vain kyllä tai ei vaihtoehdot ja tämä ehdotus sai yksimielisen kannatuksen vastaajien kesken.

Ensimmäisellä kyselykierroksella kysyttäessä, kuinka hyvin uuden muutoksen käsittely toimii jo aiemmin hyväksytyyn vaatimukseen, oli vastauksen keskiarvo 2.67, kun vastausvaihtoehdot oli mahdollista valita välillä 1–5. Yhtenä perusteluna avoimessa kentässä oli kirjattu valmiista tuotteista löytyvän ominaisuuksia, joita ei ole vaadittu ja puuttuu ominaisuuksia, jotka olivat vaatimuksissa. Nämä projektin sisällä tehdyt muutokset paljastuvat vaatimusten omistajille liian usein vasta, kun asiakkailta tulee valituksia tuotteesta.

Johtopäätöksenä kysymykseen, joka myös muista perusteluista käy ilmi, on että vaatimusten hallintaan liittyvässä viestinnässä, dokumentoinnissa ja sitä myöden läpinäkyvyydessä on kehitettävää. Sidosryhmäanalyysin tekeminen mahdollisesti parantaisi kommunikointia myös aiemmin hyväksytyjen vaatimusten muutoskäsittelyprosessissa.

Muutoksenhallintaan liittyen lisäkysymyksenä kysyttiin ensimmäisellä kierroksella, kuinka hyvin vaatimukseen liittyvä muutoshistoria on dokumentoitu ja saatavilla? Vastauksen keskiarvo oli 2.33, kun vastausvaihtoehdot oli mahdollista valita välillä 1–5. Kysymykseen jätetyistä perusteluista käy ilmi, että muutoshistorian dokumentaatiota ei usein ole saatavilla ja muutokseen liittyviä dokumentteja saattaa olla eri paikoissa ja usein myös yksittäisten henkilöiden sähköposteissa.

Johtopäätöksenä voidaan todeta, että ohjelmistokehityksen vaatimusmäärittelyn muutoksenhallinnassa on tärkeää vaatimukseen liittyvien päätösten läpinäkyvyys kaikille sidosryhmille. Laajoissa ja monimutkaisissa projekteissa päädytään helposti käsittelemään uudelleen samoja vaatimuksia ja niihin liittyviä muutoksia, jos muutoshistoriaa ei ole kunnolla dokumentoitu.

Vaatimuksen toteuttamiseen liittyviin päätöksiin liittyen, kysyttiin ensimmäisellä kyselykierroksella, kuinka hyvin päätökset ja päätöshistoria on dokumentoitu ja saatavilla vaatimukseen liittyen? Tällä kysymyksellä kartoitettiin, kuinka läpinäkyvää päätöksenteko on ja voidaanko myöhemmin palata tarkistamaan, miksi päätettiin esimerkiksi jokin vaatimus olla toteuttamatta, vaikka se oli todettu tärkeäksi. Vastauksen keskiarvo oli 2.17, kun vastausvaihtoehdot oli mahdollista valita välillä 1–5. Perusteluiksi kysymykseen kirjattiin, että päätökset jäävät usein hämäräksi, päätökset sekä päätöshistoria ei ole kaikkien saatavilla tai sitä ei ole ollenkaan.

Kirjallisuuskatsauksen perusteella on saatu viitteitä siitä, että empiirisessä ohjelmistokehitysmenetyksessä on tärkeää asioiden läpinäkyvyys, koska päätökset ovat usein subjektiivisia. Johtopäätöksenä myös vaatimuksien päätöshistoriaan liittyvässä kysymyksessä on, että läpinäkyvyyden parantaminen systemaattisen toimintamallin avulla

toisi kehitystä vaatimustenhallintaan. Systemaattisen menetelmän puuttuessa on ollut projektipäällikkö kohtaista, onko vaatimukseen liittyvää päätöshistoriaa dokumentoitu vai ei.

Ensimmäisellä kierroksella avoimena kysymyksenä kysyttiin vaatimusten priorisointiin liittyen, miten vaatimuksessa pitäisi huomioida mielestäsi taloudellinen näkökulma, käyttäjänäkökulma ja teknologianäkökulma? Kysymyksellä haettiin mielipiteitä, mitä esitetyn vaatimuksen toteutuksen priorisoinnissa tulisi painottaa. Tulisiko priorisoinnissa painottaa vaatimuksen omakustannus hintaa vai teknologisia haasteita verrattuna asiakkaille tuotettavasta lisäarvosta. Tämä kysymys toistettiin myöhemmin toisella kyselykierroksella. Perusteluina kysymykseen ensimmäisellä kierroksella vastattiin useamassa vastauksessa, että jokainen osa-alue täytyy huomioida, mutta kuitenkin suurin painoarvo on käyttäjänäkökulmassa ja ilman asiakasarvoa olevaa vaatimusta ei kannata toteuttaa ollenkaan.

Toisella kyselykierroksella haluttiin saada selkeä vastaus asiantuntijoilta, millä toimintatavalla vaatimukset pitäisi priorisoida jatkossa huomioiden vaatimukseen liittyvät eri näkökulmat. Tämän vuoksi kyselykierroksella oli annettu vain kaksi vastausvaihtoehtoa, joista ensimmäinen vaihtoehto samaan kysymykseen oli, että asiantuntijat päättävät palaverissa ilman sovittua menetelmää, huomioiden nämä asiat vaatimuksen toteutuksesta. Toisena vaihtoehtona oli, että käytetään jotain analysointi työkalua tai menetelmää asiantuntijoiden päätöksen tukena. Enemmistö asiantuntijoista 75 % valitsi jälkimmäisen vaihtoehdon, jossa ehdotetaan analysointityökalun tai -menetelmän käyttöä asiantuntijoiden päätöksen tukena.

Johtopäätöksenä, vaikka konsensusta kysymykseen ei saavutettu, oli enemmistön antaman vastauksen perusteella, että yhdessä kehitettävää vaatimusten arvioimismenetelmää suositellaan käytettäväksi jatkossa vaatimusten priorisoimiseen. Vaatimusten saaminen yhteismitallisiksi perustelee projektiin liittyville sidosryhmille vaatimukseen liittyvät päätökset ja näin ollen parantaa läpinäkyvyyttä, tarkkailua ja helpottaa mukautumiseen tarvittavia päätöksiä projekteissa.

Kun ensimmäisellä kierroksella kysyttiin avoimella kysymyksellä vaatimusmäärittelyn dokumentointiin liittyen, missä mielestäsi projekteissa esille nousseet vaatimukset tulisi säilyttää, vastauksissa oli samankaltaisia vaatimuksia kaikilla vastaajilla. Vastausten mukaan tulisi olla paikka, josta ne on helppo löytää, dokumentaation pitää olla kaikkien nähtävillä ja ei saa olla useassa eri paikassa riippuen projektista. Tämä kysymys esitettiin uudestaan vastausvaihtoehdoilla toisella kyselykierroksella, jolloin haluttiin vastaajien näkemys ohjelmistosta, jonne vaatimukset halutaan dokumentoida. Vaihtoehdot oli valittu ohjelmistoista, joita vastaajat olivat ehdottaneet ensimmäisellä kierroksella. Seitsemän vastaajaa yhdeksästä oli sitä mieltä, että vaatimusten dokumentointiin pitäisi käyttää yrityksessä käytössä olevaa Atlassian Confluence™ yhteistyöalustaa.

Kuva 10. Kyselyn tulos vaatimusten säilytyspaikasta

Missä projektiin liittyvät vaatimukset tulisi säilyttää? (0 piste)

[Lisätietoja](#)

●	Versiohallinta	0
●	Confluence	7
●	Jira	0
●	Miro	0
●	Jokin muu? Kirjoita ehdotuksesi	2



Johtopäätöksenä edelliseen kysymykseen, jonkin muun vaihtoehdon valinneet kirjoittavat perusteluissa, ettei ehdotettujen ohjelmistojen käyttö ei ole riittävän tuttua tällä hetkellä. Enemmistön näkemyksen perusteella tullaan esittämään Atlassian Confluence™ yhteistyöalustaa projektien dokumentaation käyttöön.

Kysyttäessä nähdäänkö tarpeellisena vaatimusten priorisointi osana vaatimuksen käsittelyä, ensimmäisellä kierroksella vastaajista 83,3 % antavat selvän kyllä vastauksen. Yksi vastaajista kokee, että vaatimuksen priorisointi on näkökulmakysymys. Perusteluna hän vastasi, ettei yleispätevää priorisointia ei voi tehdä, koska prioriteetti muuttuu, kun näkökulma muuttuu.

Tämä kysymys toistettiin toisella kierroksella, jotta nähdään, jos vastaajien mielipide muuttuisi, kun he näkevät ensimmäisen kierroksen tulokset ja muiden vastaajien perustelut kysymykseen. Toisella kierroksella vaatimusten priorisointia näki tarpeellisena osana vaatimusten käsittelyä 89 % vastaajista.

Kuva 11. Kysymys vaatimusten priorisoinnin tarpeellisuudesta osana vaatimuksen käsittelyä.

Näetkö tarpeellisena vaatimusten priorisoinnin osana vaatimuksen käsittelyä? (0 piste)

[Lisätietoja](#)



Johtopäätöksenä edelliseen kysymykseen voidaan todeta, että vaikka yksi vastaajista ei näe tarpeellisena vaatimusten priorisointia osana vaatimusten käsittelyä, silti priorisointia suositellaan käytettäväksi, koska perusteluina aiemmin kyselyssä osa vastaajista vastasi aiemmissa kysymyksissä, että päätettyjä asioita muutellaan ja vaatimukseen liittyvät päätökset ei ole kaikkien saatavilla tai toisinaan sitä ei ole. Priorisointi lisää läpinäkyvyyttä päätöksille ja vaatimuksien käsittelylle on luotava vakioitu menetelmä, tärkeimpänä näkökohtana kuitenkin priorisoinnista jää dokumentti, jonka voi dokumentoida projektille. Dokumentoinnin tärkeys korostuu yrityksessä, jossa tuotekehitysprojektit voivat olla ajallisesti pitkiäkin.

Kysyttäessä ensimmäisellä kysely kierroksella, oletko käyttänyt Value vs. effort-priorisointimenetelmää tai muuta menetelmää vaatimusten priorisointiin, 66.6 % vastaajista vastasi menetelmän olevan tuttu, mutta eivät välttämättä ole käyttäneet Kempillä työskennellessään. Tällä kysymyksellä kartoitettiin vastaajien aikaisempia kokemuksia kyseisestä tai muista priorisointimenetelmistä. Yksi vastaajista näkee siinä mahdollisuuden, kunhan effort-vaatimuksessa on riittävän matala, silloin tuotteisiin otetaan mukaan

mahdollisesti negatiivista arvoa eli ylimääräisiä vaatimuksia, koska niitä on ennenkin ollut. Toisella kysely kierroksella kysymys toistettiin Olisitko valmis käyttämään Value vs. effort-priorisointimenetelmää? ja kaikki kyselyn vastaajat olivat valmiita käyttämään Value vs. effort-priorisointimenetelmää.

Johtopäätöksenä kysymykseen, toisella kierroksella kysyttiin myös mielipidettä kyselyn mukana toimitetulle esimerkki työkalulle Value vs. effort-priorisointimenetelmästä. Avoimissa vastauksissa nousee hyvä ehdotus priorisointimallin yksinkertaistamisesta, jolloin sen käyttöönotto helpottuisi, mutta todennäköisesti lopputulos olisi yhtä hyvä. Vastaaja ehdottaa *"Anna value-pisteet 1–10 asteikolla "Ei hyötyä kenellekään" - "Paljon hyötyä kaikille" ja effort-pisteet 1-10 asteikolla "Helppo ja nopea tehdä" - "Hyvin suuritöinen". Tulokset voisi esittää valmiin suhdeluvun sijaan sellaisena kuin tieto on: kaksi erillistä ulottuvuutta."* Tästä ehdotuksesta kannattaa yrityksessä lähteä liikkeelle, ja kehittää sitä jatkossa palautteen perusteella.

Kysyttäessä ensimmäisellä kierroksella avoimella kysymyksellä, mikä olisi mielestäsi paras keino toteuttaa vaatimusten käsittely, johon liittyvät päätökset koskevat useampaa eri teknologiaosastoa? Tällä kysymyksellä haettiin vastaajien ehdotuksia ja näkemyksiä, miten vaatimusten käsittely tulisi hoitaa kokonaisuudessaan. Avoimista vastauksista nousee esille samankaltaisia kommentteja seuraavista asioista. Sidosryhmien edustajilla pitää olla mahdollisuus tutustua, kommentoida ja arvioida vaatimuksia etukäteen. Vaatimusten käsittelyä ei järjestetä pelkästään tuotekehitysosastolla vaan kaikilla tuotteeseen liittyvillä sidosryhmillä pitää olla mahdollisuus arvioida ja kommentoida vaatimuksia. Lopuksi vaatimuksille tulisi olla yhteinen katselmointi, jossa vahvistetaan tai haastetaan vaatimusten arvioinnit ja päätökset. Myös yksi vastaajista ehdottaa eriävää ratkaisua, jossa nimetään jonkun vastuulle tehdä päätökset vaatimuksista Kempin ja asiakkaiden kokonaisedun mukaisesti. Toisella kyselykierroksella tämä kysymys toistettiin strukturoidummin antamalla vain kaksi vastausvaihtoehtoa, jotka oli muodostettu ensimmäisen kierroksen avoimista vastauksista. Kaikki vastaajat toisella kierroksella valitsivat vaihtoehdon *"vaatimuksiin liittyvien sidosryhmien edustajien yhteinen katselmuks ja vaatimusten käsittely"*.

Ensimmäisellä kierroksella kyselyn lopussa oli kyllä ja ei kysymyksiä, joilla haettiin vahvistusta konstruktiossa esitettyihin asioihin. Kysyttiin, koetko tarpeelliseksi aiemmin kirjattujen vaatimusten tai vaatimuspohjien hyödyntämistä yhä uudelleen tulevissa projekteissa? 83 % vastaajista koki aiemmin kirjattujen vaatimusten hyödyntämisen uudelleen tulevissa projekteissa tarpeelliseksi.

Johtopäätöksenä voidaan todeta, että eriävän näkemyksen vastaajat saattavat kokea tämän mahdollisuuden riskinä, että tehdään aina samat ominaisuudet automaattisesti, ilman käsittelyä niiden tarpeellisuudesta uudessa tuotteessa. Jos on mahdollista kopioida aikaisemmin toteutetun tuotteen vaatimukset tulevaan projektiin pohjaksi helposti.

Ensimmäisellä kierroksella kysyttiin, koetko vaatimusmäärittelyn kehittämisen ja hallinnan tärkeäksi? Kaikki vastaajat kokivat tärkeäksi ja tarpeelliseksi. Tällä kysymyksellä kartoitettiin vastaajien motivaatiota tutkimuksen aihetta kohtaan.

Viimeisenä kysymyksenä ensimmäisellä kierroksella oli pitäisikö kaikkien projektiin liittyvien vaatimusten tulla jatkossakin vain tuotehallinnan kautta? 83 % vastaajista oli sitä mieltä, ettei kaikkien vaatimusten tarvitse tulla jatkossa tuotehallinta osaston kautta. Tällä kysymyksellä haettiin vastaajien mielipidettä konstruktiossa esitettyyn ehdotukseen, että kuka tahansa voi kirjata vaatimuksen projekteille.

3.5 Tulosten analysointi ja luotettavuuden pohdinta

Tulosten analysoinnissa pyrittiin tarkastelemaan hedelmällisyyttä, käytännöllisyyttä ja objektiivisuutta. Hedelmällisyydellä tarkoitetaan, että tutkimuksen tulokset herättävät uusia kysymyksiä, nostavat esiin uusia ongelmia ja vaativat uusia näkökulmia edistämään asioita. Hyvä tutkimus antaa mahdollisuuden käytännöllisiin sovelluksiin ja on sellainen, johon liittyviä teorioita voidaan tarkistaa. Objektiivisuudella tarkoitetaan, että tutkijan on noudatettava yleisesti hyväksytyjä tutkimuksen pelisääntöjä eikä voi muuttaa tuloksia tai tulkintaa, omien mieltymystensä mukaisesti. (Anttila, 2014)

Tutkimuksessa pitää pyrkiä siihen, että tutkimus kokonaisuudessaan ja sen tulokset ovat luotettavia eli valideja. On pohdittava millainen tutkimusstrategia olisi validi.

Tutkimustyössä käytetty menetelmä ei yksistään johda tietoon, vaan menetelmä on valittava sen mukaisesti, millaista tietoa tutkimuksella on haluttu. Tutkimustyössä on pyrittävä löytämään mittaus tai mittari, jolla saadaan pysyviä tuloksia aikaiseksi. Mikäli tutkimus toistettaisiin myöhemmin uudelleen, tutkimuksen pysyvyys tarkoittaisi samoja tutkimustuloksia kuin tässäkin tutkimuksessa on saavutettu ja silloin mittaustoimitus olisi reliabeli. (Anttila, 2014)

On syytä todeta aluksi, että tutkimuksessa luotiin kohdeyritykselle ainoastaan yksi toimintamalli vaatimuksien käsittelemiseksi tuotekehityksen ja tuotehallinta osastojen antamien vastauksien perusteella. Luotu toimintamalli on kuitenkin vain yksittäinen näkemys, miten vaatimustenhallintaa kannattaisi kohdeyrityksessä lähteä kehittämään. Keskeistä onkin ymmärtää, että vastauksien perusteella on tehty päätelmiä vaatimustenhallinnan kehitystarpeista, millä päästiin alkuun vaatimustenhallintaprosessin kehittämisessä. Tutkimustyö antaa pohjan ja suunnan, miten kehitystyötä kannattaa vaatimustenhallinnan kehittämiseksi jatkaa. Tutkimustyössä on pyritty siihen, että vaikka jokainen tuotekehitysprojekti on erilainen, mutta vaatimustenhallinta osuus projektista voi toimia yhdellä toimintamallilla.

Kyselyssä saatiin tietoa vaatimustenhallinnan nykytilanteesta eri näkökulmista, koska vastaajien toimenkuvat vaihtelivat ohjelmistosuunnittelijoista liiketoimintayksikön vetäjään. Jokaisella projektiin liittyvällä henkilöllä voi olla erilainen lähestymistapa vaatimustenhallintaan, joten on hyvä tunnistaa erilaiset näkökulmat toimintatapoja kehitettäessä. Tutkimuksessa kerättiin aineistoa vain sellaisilta henkilöiltä, jotka olivat halukkaita vastamaan ja tarjoamaan näkemyksiään vapaaehtoisesti. Delfoi-menetelmän avulla pyrittiin luomaan luottamuksellinen ilmapiiri ja osallistuneille henkilöille kerrottiin, että vastaaminen kyselyyn suoritetaan anonymisti ja mitään henkilöiden nimiä ei tulla käyttämään tutkimuksessa. Näin pyrittiin luomaan ilmapiiri, jossa osallistujat uskaltaisivat vastata kysymyksiin avoimesti. Vastausprosentti kyselytutkimukseen oli 50 %. Tutkimuksen yhdeksi epävarmuustekijäksi voidaan nostaa myös tutkijan osaaminen. Tutkijalla ei ollut aikaisempaa kokemusta vastaavan laadullisen tutkimuksen tekemisestä. Aloitteleva tutkija

on voinut esimerkiksi tulkita tutkimusaineistoa väärin tai luottanut joihinkin havaintoihin totuuksina. Tässä tutkimuksessa Delfoi-menetelmän avulla kerätyn aineiston tulokset ovat nykytilanne kartoitusta ja näin ollen hyvin tilannesidonnaisia sekä niihin liittyy inhimillisiä tekijöitä. Tämän vuoksi oletettavasti tutkimuksessa voisi tulla poikkeavia tuloksia, jos tutkimus mahdollisesti toistettaisiin myöhemmin uudelleen kohdeyrityksessä.

Teoreettinen viitekehys tukee hyvin tutkimuksen lopputulosta. Vaatimustenhallintaa on tutkittu eri yhteyksissä paljon ja siitä on esitetty useita teorioita. Tätä tutkimusta varten löytyi hyvin lähdemateriaalia ja löydetyn teorian pohjalta voidaan aloittaa vaatimustenhallinnan kehittäminen tutkimuksessa esiin tulleiden kehityskohteiden avulla. Tutkimuksessa havaittiin, että kohdeyrityksessä vaatimustenhallinnan kypsyytaso ei ollut vielä kovin korkea, mutta suuntana ja toiveena on selkeästi vaatimustenhallinnan systematisoiminen jatkossa.

3.6 Yhteenveto tutkimustuloksista

Aluksi kun kysyttiin numeroarviointia vaatimukseen liittyvästä muutoskäsittelystä, vaatimusten muutoksien dokumentoinnista ja historiatiedoista, sekä päätöksistä vaatimukseen ja muutokseen vaatimuksissa, kaikki vastaukset jäivät keskiarvoltaan alle 3. Asteikolla 1–5 tulos viittaa siihen, että kehittämistoimenpiteitä tarvitaan. Tulosta voidaan parantaa läpinäkyvyyttä parantamalla, lisäämällä dokumentaatiota ja dokumentaation vaatimukseen liittyen pitää olla kaikkien saatavilla. Yhteistyöalustaksi, mihin vaatimukset dokumentoidaan, enemmistö kyselyssä ehdottaa Atlassian Confluence™ yhteistyöalustaa. Atlassian Confluence™ ja Atlassian JIRA™ ovat joustavia työkaluja, joihin on helppo lisätä ominaisuuksia ja joita voi räätälöidä omiin tarpeisiin sopivaksi. Ohjelmistot ovat verkkopohjaisia ohjelmistoja, joten ne ovat avoimia koko yrityksen organisaatiolle.

Toimenpide mihin löydettiin, konsensus on sidosryhmäanalyysin tekeminen projektien alussa, johon kirjataan sidosryhmät ja heidän tarpeensa ja sovitaan käytännöt kommunikoinnista heidän kanssaan projektin edetessä.

Suosittelaa käytettäväksi jatkossa vaatimusten käsittelemiseen ja priorisointiin analysointityökalua tai -menetelmää asiantuntijoiden päätöksen tukena. Vaatimusten priorisointiin kaikki vastaajat olivat valmiita käyttämään Value vs. effort-priorisointimenetelmää. Aluksi suositellaan käytettäväksi kyselyn vastaajan ehdottamaa yksinkertaistettua versiota Value vs. effort-priorisointimenetelmästä ja kehittää sitä jatkossa saadun palautteen perusteella.

Vaikka täyttä konsensusta ei saatu kysymykseen missä ehdotettiin, että kuka tahansa voi kirjata vaatimuksia projekteille. Vastauksista nousee esille, että nykyistä toimintamallia tulisi kehittää niin, että vaatimukset kerättäisiin yhdenmukaisella tavalla ja niihin liittyvät päätökset dokumentoidaan niin että tieto niistä on kaikille sidosryhmille saatavilla.

4 Pohdinta

Tutkimuksessa etsittiin toimintatapoja, miten projektin vaatimukset saadaan kerättyä yhdenmukaisella tavalla talteen eri sidosryhmiltä ja dokumentoitua ne projektien käytettäväksi sekä miten nämä esitetyt vaatimukset käsitellään ja priorisoidaan projekteissa. Työn tavoitteena oli tehdä kehitysehdotus toimintamallista vaatimusten dokumentoimiseksi ja niiden käsittelemiseksi. Toimintamallilla saataisiin Kempin tuotekehitysprojektien toimintaa yhtenäisemmäksi ja tällä tavoin parannettua valmistuvien tuotteiden laatua. Opinnäytetyön teoreettinen viitekehys käsitteli ohjelmistojen kehitysmallien teoriaa sekä vaatimusmäärittely menettelyjen teoriaa ja niiden kehittämistä. Keskeisiä asioita toiminnan laadun parantamiseen vaatimusmäärittelyssä ovat toimenpiteet, jotka koskevat vaatimuksien käsittelyn vakiointia kehitetyn menetelmän avulla. Toiminnan laatua voidaan tehostaa parantamalla läpinäkyvyyttä. Läpinäkyvyyttä lisää päätöksien dokumentointi.

Tärkeää on haasteiden välttämiseksi tiivis ja avoin yhteistyö sidosryhmien kanssa. Vaatimukset ja niihin liittyvät päätökset ovat tärkeää saattaa ymmärrettävään muotoon, että niiden toteuttaminen on selkeää kaikille sidosryhmille.

Kyselytutkimuksessa Kemppiläisille kartoitettiin toimintamallin kehitystarpeita, haasteita ja hyväksi havaittuja käytäntöjä. Tutkimustyön tavoitteena oli kerätä tietoa nykyisistä

toimintatavoista ja löytää konsensus uudesta toimintamallista Delfoi-tutkimuksen avulla hyödyntäen mahdolliset nykyiset hyvät käytännöt osaksi tuotekehityksen ohjelmistoprojektin vaatimuksien käsittelyä sekä yhtenäistää projektien toimintatapoja. Kyselyn vastaukset tukivat teoriasta esille nousseita yleisiä ohjelmistoprojektien vaatimusmäärittelyn toiminnan puutteita. Vaatimusmäärittelyyn toimintatavoissa löytyi Kempin ohjelmistoprojekteissa seuraavia ongelmia.

- Uusia vaatimuksia tulee projektin varrella ja myös jo olemassa olevia sovittuja asioita muutellaan.
- Vaatimukseen liittyvät päätökset sekä päätöshistoria ei ole kaikkien saatavilla tai toisinaan sitä ei ole.
- Yhtenäinen menetelmä vaatimusten päätöksien tekemiseen ja käsittelyyn puuttuu.

Vaatimusmäärittelyyn liittyen kannattaa kehittää dokumentaatiota projekteissa yhdenmukaiseksi, sekä pyrkiä lisäämään sidosryhmissä ymmärrystä vaatimusmäärittelyn tarpeellisuudesta. Kun kaikki projektiin liittyvät sidosryhmät ymmärtävät vaatimustenhallinnan hyödyt, löytyisi aikaa myös vaatimusmäärittelylle. Dokumentoinnin kehittämisen tärkeys korostuu, koska Kempin tuotekehitysprojektit voivat olla ajallisesti pitkiäkin, jolloin asioiden unohtamisen vaara on suuri ilman systemaattista dokumentointia. Dokumentoitu tieto säästää turhalta muistikuormalta, näin estetään myös samojen asioiden ja vaatimusten käsittely uudelleen pitkän projektin aikana. Pelkästään dokumentointia lisäämällä tiedon jakaminen sekä läpinäkyvyys ja jatkuvuus projekteissa parantuvat merkittävästi. Dokumentoinnilla on mahdollista myös ajattelun kirkastaminen projektissa kaikille sidosryhmien jäsenille. Kun vaatimuksen joutuu selkeästi kirjaamaan ylös, se täsmentyy myös vaatimuksen esittäjälle itselleen.

Tutkimustyön yhtenä tavoitteena oli systemaattisen toimintamallin kehittäminen vaatimusten käsittelyä varten, vaikka kyselyn perusteella ei täyttä konsensusa saavutettu, mutta teorian ja enemmistön kannan perusteella suositellaan sitä käytettäväksi. Mallissa kannattaa lähteä liikkeelle yksinkertaisemmasta Value vs. effort-menetelmä versiosta ja kehittää sitä käytännön kokemusten ja saadun palautteen perusteella. Perusteita toimintamallin hyödyistä, jotka tukevat scrum-toimintamallilla tapahtuvaa

ohjelmistokehitystä. Toimintamallin avulla laatu paranee lopputuotteissa. Kun jokainen projekti tai henkilö ei määrittele itse ja ehkä vielä kerta kerralta uudelleen asioita tai kriteerejä vaatimuksille, niin projektien ja koko organisaation työstä tulee tasalaatuisempaa, koska päätöksen pohjana on menetelmä, joka vähentää henkilökohtaisia näkemyksiä päätöksissä. Jos on tarvetta muuttaa vaatimusten käsittelyä menetelmää muuttamalla, otetaan se kaikissa projekteissa käyttöön samalla. Näin saadaan projektien välillä vaihtelut minimiin ja tuotteiden vaatimustaso ja laatu tasaiseksi. Näin saadaan myös selittämättömät viivästykset ja puutteet lopputuotteissa jää pois. Toimintamallin avulla myös päätöksille tulee läpinäkyvät perusteet, että mitä on päätetty ja millä perusteilla.

Teoriaosuudessa tutkittu toinen vaatimusten käsittelymenetelmä Kano-malli ei sovellu sellaisenaan käytettäväksi, koska kun vaatimuksia on paljon, kasvaa kyselyyn vastaamiseen tarvittava aika. Vastaajien on myös ymmärrettävä mallissa selkeästi mitä kyselyn jokainen vaatimus tarkoittaa ja miten ne vaikuttavat tuotteeseen. Kano-malli ei ota vaatimusten toteutukseen riittävästi kantaa, vaan toteutus vaatisi oman menetelmän ja vasta näiden yhteistuloksena saataisiin lopulliset prioriteetit vaatimuksille.

Tutkimusmenetelmässä oli rajoitteita, koska tutkimuksessa esitettyä toimintamallia vaatimusten käsittelyyn ei toteutettu käytännössä. Koska tämä opinnäytetyö rajoittui vain ohjelmistokehitysprojektin vaatimusmäärittelyvaiheeseen, jäi vielä käytännön toteutusvaiheen mahdollisesti tuomat haasteet toimintamallin käyttöönotossa kokonaan käsittelemättä. Tutkimuksessa esitetyt tulokset eivät ole täysin yleistettäviä, eikä yleistettävyys ollut edes tutkimuksen tavoitteena. Tutkimustyössä esiin nousseita kehitystoimenpiteitä voidaan kuitenkin hyödyntää muidenkin alojen ohjelmistoprojekteissa sekä muissakin projekteissa, niin sarjatuotantoon liittyvässä valmistuksessa kuin asiakaskohtaisissa projekteissa.

Koin mielekkääksi projektin vaatimusmäärittely- ja priorisointivaiheen kehittämisessä uusien työmenetelmien opettelun ja sen, että jouduin kehittämään toimintamallia itsenäisesti, kun mitään täysin valmista menetelmää ei ollut aikaisemmin käytössä yrityksessä. Lopuksi voisi tiivistää, että tutkimuksen päätavoitteena oli kiinnittää huomiota vaatimustenhallinnan

kehittämiseen systemaattisemmaksi kohde yrityksessä ja tämän kehittämiseen työssä löydettiin menetelmiä.

4.1 Jatkotutkimusehdotukset

Tutkimuksessa jäi myös paljon selvittämättä. Tuoteprojektien vaatimustenhallinnan kehittäminen on yrityksen menestystekijänä niin merkittävä, että sitä kannattaa tutkia jatkuvasti lisää pyrkien löytämään vaatimustenhallinnalle uusia näkökulmia ja tarkastella myös vanhoja näkökulmia uusin silmin. Digitalisaation nopean kehittymisen vuoksi ohjelmistoprojektien vaatimukset muuttuvat jatkuvasti. Tässä mielessä vaatimustenhallintaan kohdistuu jatkuvasti sitä vähitellen muokkaavia muutostekijöitä, joiden vaikutusta kehittymiseen tulisi seurata ja tutkia jatkossakin.

Jatkotutkimusmahdollisuuksia on vielä paljon kohdeyrityksenkin vaatimustenhallinnassa.

- Value vs. effort-priorisointimenetelmän soveltuminen scrum-ohjelmistokehitysmenetelmän kanssa käytännössä kohdeyrityksessä.
- Muiden kuin työssä esitettyjen vaatimusten priorisointimenetelmien soveltuvuus kohdeorganisaation käyttöön ja soveltuminen scrum-ohjelmistokehitysmenetelmän kanssa.
- Muiden kuin työssä esitettyjen ohjelmistokehitysmenetelmien soveltuvuus kohdeorganisaation käyttöön ja hyvien käytäntöjen soveltaminen käyttöön.
- Tuotteiden ja vaatimusten elinkaaren merkitys vaatimustenhallinnassa ja vaatimuksien priorisoinnissa.

Lähteet

Airila, M. (2022). *Mitä on lean? Leanisti kohti yhä sujuvampaa työtä*. Talentree.

<https://talentree.fi/konsultointi/mita-on-lean/>

Andreev, M. (2.11.2020). Prioritization Methods and Techniques - Part 4: Value vs. Effort Matrix as a Lean prioritization approach. *BlinkLane Consulting*.

<https://www.blinklane.com/insights/prioritization-methods-and-techniques-part-4-value-vs-effort-matrix-lean-prioritization-approach/>

Anttila, P. (17.5.2014). *Tutkimisen taito ja tiedon hankinta*. Metodix.

<https://metodix.fi/2014/05/17/anttila-pirkko-tutkimisen-taito-ja-tiedon-hankinta/>

Atlassian. (n.d.). *The power of teamwork*. <https://www.atlassian.com/company>

ESAB. (n.d.). *Hitsauslisäaineet*. <https://assets.esab.com/assetbank-esab/assetfile/12268.pdf>

Firesmith, D. (2004). *Prioritizing Requirements*. Software Engineering Institute. ETH Zurich.

Hietikko, E. (15.3.2021). KANO-ONGELMAT. BLOGI SAVONIA.

<https://blogi.savonia.fi/savoniaonkone/2021/03/15/kano-ongelmat/>

Hietikko, E. (9.3.2021). KANO-PERIAATE. BLOGI SAVONIA.

<https://blogi.savonia.fi/savoniaonkone/2021/03/09/kano-periaate/>

Johnson, J. (2020). *CHAOS 2020: Beyond Infinity*. Standish Group.

Juvonen, R. (2018). Ohjelmistoprojektin sudenkuopat ja miten ne vältetään. Books on Demand. <https://www.ellibslibrary.com/fi/hamk/9789528001454>

Kemppi. (2020). Kemppi globaalisti lähellä. <https://www.kemppi.com/fi-FI/yritys/kemppi/globalisti-lahella/>

Kemppi. (2020). Kemppi yrityksenä. <https://www.kemppi.com/fi-FI/yritys/kemppi/kemppi-yrityksena/>

Kuusi, O. (1999). Delfoi-menetelmä. Metodix. <https://metodix.fi/2014/05/19/kuusi-delfoi-metodi/>

Linturi, H. (2019). Uudistuva Delfoi-metodi ja eDelphi 2020. Metodix. <https://metodix.fi/2020/01/06/uudistuva-delfoi-metodi/>

Lukka, K. (2001). *Konstruktiivinen tutkimusote*. Metodix. <https://metodix.fi/2014/05/19/lukka-konstruktiivinen-tutkimusote/>

Luukkainen, M. (2021). *Ohjelmistotuotanto ja sen osa-alueet*. <https://ohjelmistotuotanto-hy.github.io/osa1/>

Luukkainen, M. (2021). Vaatimusmäärittely. <https://ohjelmistotuotanto-hy.github.io/osa2/>

Paakki, J. (2011). *Ohjelmistojen vaatimusmäärittely*. Helsingin yliopisto. <https://www.cs.helsinki.fi/u/paakki/Vaatimus-11-Luentokalvot-4.pdf>

Productplan. (n.d.). *Value vs. Complexity*. ProductPlan. Haettu 19.4.2022 <https://www.productplan.com/glossary/value-vs-complexity/>

Royce, W. (1970). Managing the development of large software systems. IEEE WESCON. <http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf>

Schwaber, K. & Sutherland, J. (2020). The 2020 Scrum guide™ <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf#zoom=100>

Tieteen termipankki. (6.11.2022). *Kasvatustieteet sidosryhmä*. <https://tieteentermipankki.fi/wiki/Kasvatustieteet:sidosryhmä>

Zacarias, D. (n.d.). *The complete guide to the Kano model*. Career.pm.

<https://www.career.pm/briefings/kano-model/>

Liite 1: Esimerkkityökalu Value vs. Effort-menetelmästä

Kirjaa vaatimus tähän: <input type="text"/>		
<i>Valitse vaatimusta kuvaavimmat vastaukset pudotus valikoista Value & Effort laskee prioriteetin</i>		
VALUE	Valitse sopivin	Pisteet
Toiminnon uutuusarvo?	Täysin uusi toiminto joka mahdollisesti patentoitavissa	20
Onko tämä ominaisuus "perusvaatimus" Odottavatko asiakkaat tätä ominaisuutta?	Asiakkaat odottavat tätä ominaisuutta ja ilman tätä tuotetta ei voi käyttää	20
Hyödyttääkö tämä ominaisuus useimpia käyttäjiä vai pienempää osaa käyttäjistä?	Tämä ominaisuus hyödyttää tietyn maan käyttäjiä pelkästään.	5
Saadaanko tällä ominaisuudella Lisenssi myyntiä kasvatettua?	Tämän ominaisuuden lisensointi on päivän selvä asia.	20
EFFORT	Valitse sopivin	Pisteet
Voiko tämän ominaisuuden tuoda myöhemmin	Tämän ominaisuuden voi lisätä päivityksellä mutta vaatii komponentin lisäyksen jonka asiakas voi tarvittaessa itse lisätä	8
Onko tälle ominaisuudelle teknologia Kempillä	Teknologia ei ole täysin hallussa vain muutama henkilö hallitsee jollain tasolla teknologian. Tarvittavista komponenteista ei varmuutta	8
Työmäärä arvio	Työ on helppo toteuttaa ja siihen löytyy useita tekijä vaihtoehtoja	1
Miten tämä vaatimus vaikuttaa tuotteen omakustannukseen?	Pieni vaikutus (huomaamaton)	5
Priority	Fill ins Big value / Lot of work	3,0

Vaatimustenhallinta kyselykierros 1

Kemppi Oy

Tämä kysely liittyy Jarno Kolkan ylemmän ammattikorkeakoulun opinnäytetyöhön, joka tehdään osana teknologiaosaamisen johtamisen tutkintoa.

Tutkimustyö tehdään KWM liiketoimintayksikön tuotekehitysosastolle ja se liittyy vaatimustenhallintaan tuotekehitysprojekteissa.

Tämä kyselytutkimus tehdään Delfoi-menetelmää käyttäen.

Delfoi-lyhyesti

Delfoi-menetelmä on kyselytutkimusta lähestyvä laadullinen menetelmä, jolla pyritään ennustamaan tulevaisuuden näkymiä ja muutoksia tai saamaan selkoa asioista, jotka ovat epäselviä ja arvaamattomia. Delfoi-prosessissa tuotetaan erilaisia näkökulmia, hypoteeseja ja väitteitä, jotka saatetaan avoimen asiantuntijatestin ja argumentoinnin kohteeksi. Prosessissa pyritään seulomaan näkemykset jaetuiksi tai erimielisiksi yhteisönäkemyksiksi. Molemmat lopputulemat ovat arvokkaita. Erimielisiä voidaan olla paitsi argumenteista myös tavoitteista, vaihtoehtojen todennäköisyydestä ja haluttavuudesta.

Delfoi menetelmä käytännössä.

Delfoi-menetelmässä kuullaan nykyisin tyypillisesti kahdella tai kolmella kierroksella 15–40 asiantuntijaa. Keskeinen piirre menetelmässä on vastauksien antaminen ilman nimiä eli anonymisti. Päinvastoin kuin gallupeissa mielipiteitä ei kerätä vain analysoitavaksi, vaan vastaustieto kierrätetään takaisin asiantuntijoille. Palautetiedon avulla vastaajia ohjataan perustelemaan valintojaan. Tiedon muodostus etenee kierroksittain niin, että edellinen kyselykierros muodostaa pohjan seuraavalle. Delfoi-tutkimus käynnistetään usein jostain ajankohtaisesta riita- tai kiistakysymyksestä.

Tässä tutkimuksessa olennaisin selvittävä asia on: Miten Kempillä tulisi tuotteiden vaatimukset käsitellä ja priorisoida?

Sähköpostin liitteenä Powerpoint, jossa on lisätietoa ja konstruktio uudesta toimintatavasta Kempillä.

* Pakollinen

1.Miten hyvin vaatimusten hallinta tällä hetkellä toimii tuotekehitysprojekteissa?

(Arvioi asteikolla 1–5 ja perustele tarvittaessa) *

1 2 3 4 5
Välttävästi Erinomaisesti

2.Perustelut ja kommentit edelliseen kysymykseen

3.Kuinka hyvin uuden muutoksen käsittely toimii jo aiemmin hyväksytyyn vaatimukseen? (Arvio asteikoilla 1–5) Perustele ja kommentoi tarvittaessa *

1 2 3 4 5

Välttävästi

Erinomaisesti

4.Perustelut ja kommentit edelliseen kysymykseen

5.Kuinka hyvin vaatimukseen liittyvä muutoshistoria on dokumentoitu ja saatavilla?

(Arvioi asteikolla 1–5) Perustele tarvittaessa. *

1 2 3 4 5

Välttävästi

Erinomaisesti

6.Perustelut ja kommentit edelliseen kysymykseen

7. Kuinka hyvin päätökset ja päätöshistoria on dokumentoitu ja saatavilla vaatimukseen liittyen? (Arvioi asteikolla 1–5) Perustele tarvittaessa. *

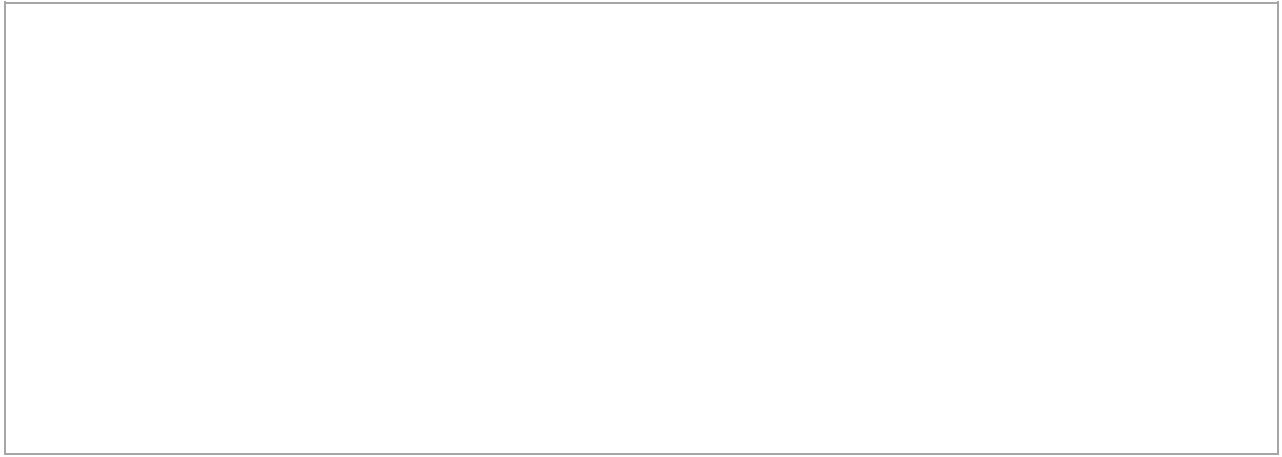
1 2 3 4 5

Välttävästi Erinomaisesti

8. Perustelut ja kommentit edelliseen kysymykseen

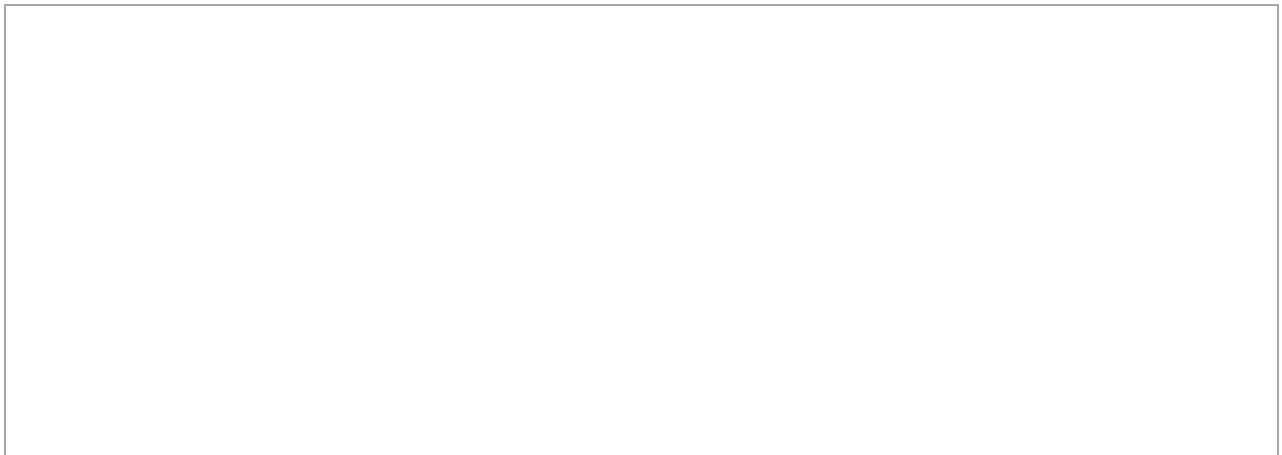
9. Miten vaatimuksessa pitäisi huomioida mielestäsi taloudellinen näkökulma, käyttäjän näkökulma ja teknologian näkökulma? *

10. Missä mielestäsi projekteissa esille nousseet vaatimukset tulisi säilyttää? *

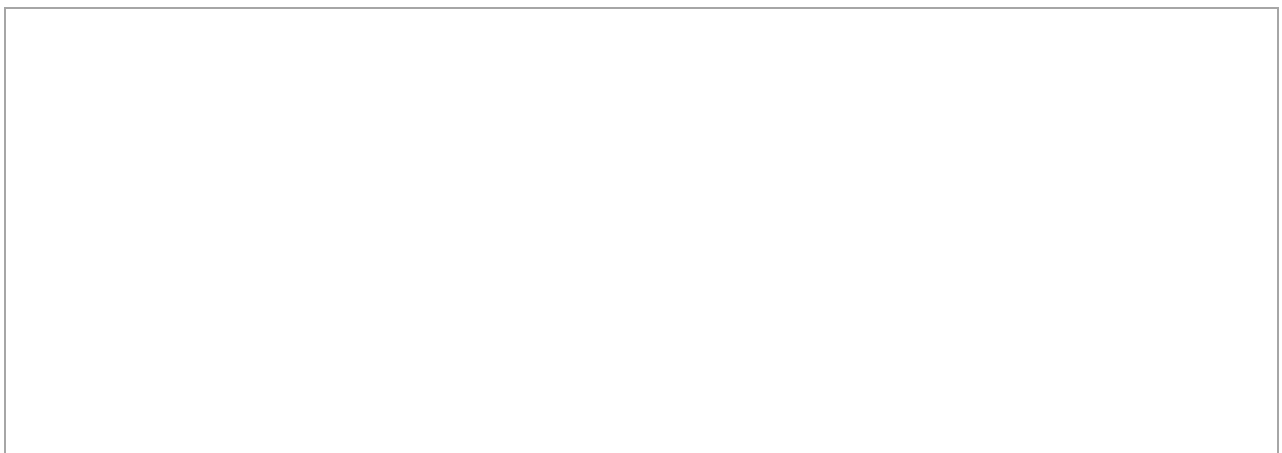


11.Näetkö tarpeellisena vaatimusten priorisoinnin osana vaatimuksen käsittelyä?

Perustele vastauksesi *



12.Oletko käyttänyt "Value vs. Effort" priorisointimallia tai muuta menetelmää vaatimusten priorisointiin? *



13. Mikä olisi mielestäsi paras keino toteuttaa vaatimusten käsittely, johon liittyvät päätökset koskevat useampaa eri teknologiaosastoa? *

14. Koetko tarpeelliseksi aiemmin kirjattujen vaatimusten tai vaatimuspohjien hyödyntämistä yhä uudelleen tulevilla projekteilla? *

- Koen tarpeelliseksi
- En koe tarpeelliseksi

15. Koetko vaatimusmäärittelyn kehittämisen ja hallinnan tärkeäksi? *

- Koen tärkeäksi
- En koe, vaatimustenmäärittely ja hallinta on jo kunnossa

16. Pitäisikö kaikkien projektiin liittyvien vaatimusten tulla jatkossakin vain tuotehallinnan kautta? *

- Kyllä
- Ei

17.Jäikö jokin tärkeä asia puuttumaan kyselyssä? Voit antaa muuta palautetta kyselystä.

Liite 3: Kyselykierros 2

Vaatimustenhallinta kyselykierros 2

Kemppi Oy

Tämä kysely liittyy Jarno Kolkan ylemmän ammattikorkeakoulun opinnäytetyöhön, joka tehdään osana teknologiaosaamisen johtamisen tutkintoa.

Tutkimustyö tehdään KWM liiketoimintayksikön tuotekehitysosastolle ja se liittyy vaatimustenhallintaan tuotekehitysprojekteissa.

Sähköpostin liitteenä PowerPoint kooste edellisestä kyselykierroksesta. Liitteenä myös esimerkkityökalu Value vs. effort-menetelmästä.

* Pakollinen

1.Miten hyvin vaatimusten hallinta tällä hetkellä toimii tuotekehitysprojekteissa?

(Arvioi asteikolla 1–5) *

1 2 3 4 5

Välttävästi Erinomaisesti

2.Olisiko mielestäsi vaatimustenhallinnan kannalta projektin alussa hyvä tehdä sidosryhmäanalyysi? *

- Kyllä
- Ei

3.Voisiko mielestäsi vaatimuksien esittäminen tuoteprojekteihin olla kaikkien Kemppläisten kirjattavissa vastaavalla lomakkeella kuin KempIdea? *

- Kyllä
- Ei

4.Miten esitetyn vaatimuksen toteutuksen ja priorisoinnin arvioinnissa pitäisi mielestäsi huomioida taloudellinen näkökulma, käyttäjänäkökulma ja teknologianäkökulma? *

- Asiantuntijat päättävät palaverissa ilman sovittua menetelmää, nämä asiat huomioiden vaatimuksen toteutuksesta
- Käytetään jotain analysointi työkalua/metelmää asiantuntijoiden päätöksen tukena?
- Jokin muu toimintapa? Perustele vastauksesi

5.Jos valitsit edellisessä 4. kysymyksessä "Jokin muu toimintapa?" niin perustele vastauksesi

4. Kysymykseen Perustelut tai kommentit.

6. Missä projektiin liittyvät vaatimukset tulisi säilyttää? *

Ehdotukset kirjattu edellisen kyselykierroksen perusteella.

Versiohallinta

Confluence

Jira

Miro

Jokin muu? Kirjoita ehdotuksesi

7. Jos valitsit edellisessä 6. kysymyksessä vaihtoehdon "Jokin muu?" Kirjoita tähän ehdotuksesi

8. Näetkö tarpeellisena vaatimusten priorisoinnin osana vaatimuksen

käsittelyä? *

Kyllä

Ei

9.Olisitko valmis käyttämään Value vs. effort-mallia? *

- Kyllä
- Ei

10.Mielipiteesi miltä vaikuttaa esimerkkinä toimitettu Value vs. effort-priorisointimalli? *

11.Mikä olisi mielestäsi paras keino toteuttaa vaatimusten käsittely? *

- Vaatimukseen liittyvien sidosryhmien edustajien yhteinen katselmus ja vaatimusten käsittely
- Nimetään jonkun vastuulle tehdä päätökset vaatimuksista Kempin ja asiakkaiden kokonaisedun mukaisesti

12. Voit antaa palautetta kyselystä.