Korkeakoulujen työharjoittelujärjestelmän vaatimusmäärittely

Sanna Arvilommi

Haaga-Helia ammattikorkeakoulu

Amk-opinnäytetyö

2021

Tradenomin tutkinto

Tekijä(t)
Sanna Arvilommi

Tutkinto
Tradenomi (AMK), tietojenkäsittelyn koulutusohjelma

Raportin/Opinnäytetyön nimi
Korkeakoulujen työharjoittelujärjestelmän vaatimusmäärittely

Sivu- ja liitesivumäärä
30 + 4

Tämä raportti on kuvaus keväällä 2021 opinnäytetyönä toteutetusta vaatimusmäärittelystä, joka on tehty toimeksiantona Haaga-Helialle. Projekti on tehty osana oppilaitoksen ohjelmistohanketta, jonka tavoitteena on kehittää korkeakouluille ohjelmisto työharjoittelun ohjaus- ja seurantamenettelyn tueksi. Ohjelmiston kehitys oli aloitettu syksyllä 2020 ja kehityksen edetessä alustavaksi laadittu tuotekehitysjono oli todettu ohjelmistonkehityksen ohjaukseen riittämättömäksi.

Vaatimusmäärittely kertoo mitä kehitettävällä ohjelmistolla tulee voida tehdä. Projektin tuotoksena syntyneessä vaatimusmäärittelydokumentissa on esitetty ohjelmiston toiminnalliset käyttäjävaatimukset, jotka ovat perusta ohjelmistoprosessille sekä ohjelmistohankinnan, että kehityksen näkökulmasta.

Projektissa tehtiin laaja käyttäjävaatimusten kartoitus tarkastelemalla Haaga-Heliassa kerättyä taustamateriaalia sekä haastattelemalla harjoitteluprosessin asiantuntijoita Metropolia ammattikorkeakouluissa. Aineistosta esiin nostetut vaatimukset analysointiin ja validoitiin ohjelmistoprojektin ohjausryhmälle järjestetyissä työpajoissa ja vaatimukset dokumentoitiin vaatimustaulukoksi, jonka avulla on mahdollista hallita vaatimuksia ketterässä ohjelmistoprosessissa.

Projektin tuotoksena syntynyt vaatimusmäärittely on ollut kevään 2021 ohjelmistokehityksen onnistumiselle ehdottoman tärkeä ja laadittua vaatimusmäärittelydokumenttia tullaan hyödyntämään ohjelmiston kehityksessä myös tulevaisuudessa. Projektissa tehty harjoitteluprosessin kuvaus tukee myös korkeakoulun työharjoitteluun liittyvän toiminnan kehitystä. Haaga-Heliassa ei ole valmista harjoitteluprosessista laadittua prosessikuvausta, jota kehityksessä voidaan tarkastella.

Projektissa tehty vaatimusten kartoitus osoittaa, että korkeakoulujen työharjoitteluun liittyvä prosessi sisältää paljon toisistaan poikkeavia toimintoja, joista osa johtuu yhtenäisen toimintatavan puutteesta. Eriävät vaatimukset on projektin tavoitteen mukaan nostettu esiin, jotta ratkaisusta voidaan päättää ohjelmistokehityksen ja prosessikehityksen edetessä. Myös projektissa laadittu vaatimustaulukko sisältää ristiriitoja, joita ei mahdollisesti voida ratkaista ohjelmistohankkeessa, vaan toimintaa koskevat tavoitteet tulisi selvittää ensin kokonaisprosessin kehitystyössä.

Asiasanat
Vaatimusmäärittely, ohjelmistosuunnittelu, ohjelmistotuotanto, digitalisaatio

Sisällys

1 Johdanto 1
1.1 Opinnäytetyön tausta ja tavoite 1
1.2 Opinnäytetyön tehtävä 2
1.3 Tehtävän rajaus 3
2 Keskeiset käsitteet 4
3 HHTrainee hankkeen tausta ja tavoite 5
4 Lähestymistapoja ohjelmistoprosessiin 7
4.1 Vesiputousmalli 7
4.2 Ketterät kehitysmallit 8
4.3 Hybridimallit 9
4.4 Digitalisaatio ja EXOD malli 10
5 Vaatimusmäärittelyn käytäntöjä 12
5.1 Vaatimusmäärittelyn vaiheet 13
5.2 Vaatimusten dokumentointi 14
5.2.1 Use case eli käyttötapaus 15
5.2.2 User story eli käyttäjätarina 15
6 Opinnäytetyön tehtävän toteutus 16
6.1 Tehtävä 1: Vaatimusten kartoitus aiemmin kerätystä aineistosta 16
6.2 Tehtävä 2: Metropolian asiantuntijahaastattelut 17
6.3 Tehtävä 3: Vaatimusten analysointi 19
6.4 Tehtävä 4: Vaatimusten dokumentointi 19
7 Tulokset 23
7.1 Vaatimusmäärittelydokumentin ristiriidat 23
7.2 Vaatimusmäärittelydokumentin kehittämiskohteita 25
7.3 Kehittämismallien soveltuvuus HHTrainee hankkeeseen 25
8 Pohdintaa opinnäytetyöprojektista 27
Liite 1. Otos vaatimustaulukosta 31
Liite 2. Käyttötapauskaavio 32
Liite 3. Sidosryhmäkaavio 33
Liite 4. Taulukko harjoitteluprosessin toiminnoista eri vaiheissa 34

1 Johdanto

1.1 Opinnäytetyön tausta ja tavoite

Haaga-Heliassa on laajasti tunnistettu tarve lisätä läpinäkyvyyttä ja yhtenäistää toimintoja opiskelijoiden työharjoitteluprosesseissa. Nykyisessä järjestelmässä työharjoittelun ohjaus- ja seurantamenettelyt saattavat poiketa toisistaan koulutusohjelma- tai harjoitteluohjaajakohtaisesti, eikä työharjoittelukoordinaattoreilla ole riittävää näkyvyyttä opiskelijan edistymiseen tai ohjaus- ja seurantamenettelyihin harjoittelussa. Yhtenäistä sähköistä järjestelmää ei ole käytössä. Harjoitteluprosessin digitalisointi on katsottu tarpeelliseksi, jotta edellä mainitut ongelmat voidaan ratkaista nykyaikaisella tavalla ja taata opiskelijoille yhdenvertainen työharjoittelun ohjaus. Lisäksi Haaga-Heliassa on tunnistettu digitalisoinnin mahdollisuudet tarjota uudenlaista työelämän tarpeista kertovaa tietoa, jota voidaan hyödyntää koulutuksen laadun parantamiseksi.

Työharjoitteluprosessiin liittyviä harjoittelun ohjaus- ja seurantatoimintoja tukemaan on päätetty Haaga-Heliassa kehittää korkeakouluyhteisölle sopiva työharjoittelujärjestelmä, joka kokoaa työharjoitteluihin liittyvät tiedot yhteen paikkaan ja mahdollistaa harjoitteluun liittyvien toimintojen automatisoinnin. Kehitettävän ohjelmiston tulee vastata korkeakoulujen harjoitteluprosessien tarpeisiin. Hankkeen kehittämisen kohteena olevasta ohjelmistosta käytetään toistaiseksi työnimeä HHTrainee, joksi sitä kutsutaan myös tässä raportissa.

Päätös HHTrainee hankkeesta oli tehty jo ennen tätä opinnäytetyöprojektia, ja siihen liittyen oli tehty projektin käynnistämiseen vaaditut selvitykset ja raportit, kuten projektisuunnitelma ja projektiorganisaation nimeäminen. Ohjelmiston kehitystyö oli myös aloitettu syksyllä 2020, jolloin kehityksessä oli edetty toteutuskeskeisesti perustuen alustaviin Haaga-Helian nykyisestä harjoitteluprosessista johdettuihin vaatimuksiin. Kattavaa vaatimusmäärittelyä ei ollut vielä tehty ja alustavaksi suunniteltu vaatimusmäärittely oli jäänyt kehityksen edetessä riittämättömäksi. HHTrainee ohjelmiston kehityksen tueksi tarvittiin vaatimusmäärittely, johon on koottu laajasti eri käyttäjiltä tarkat toiminnalliset vaatimukset. Laajalla ohjelmiston vaatimusten kartoituksella katsottiin mahdolliseksi nostaa esiin myös harjoitteluprosessin kehittämisen kohteita toiminnan kehityksen tueksi. Työn tärkeimpänä tavoitteena oli kartoittaa mahdollisimman tarkasti käyttäjien esittämät toiminnalliset vaatimukset korkeakoulun työharjoittelun ohjaus- ja seurantajärjestelmälle ja dokumentoida ne ketterän ohjelmistokehityksen kannalta järkevällä tavalla.

1.2 Opinnäytetyön tehtävä

Opinnäytetyöprojektin ensimmäisenä tehtävänä oli kartoittaa toiminnalliset vaatimukset HHTrainee hankkeeseen liittyen tehdyistä, jo olemassa olevista aineistoista. Toisena tehtävänä oli haastatella Metropolia ammattikorkeakoulun edustajia. Haastatteluista saaduilla tiedoilla tuli täydentää ensimmäisessä vaiheessa kartoitettuja vaatimuksia. Kolmantena tehtävänä oli vaatimusten analysointi yhdessä projektin ohjausryhmän kanssa ja neljäntenä tehtävänä oli mallintaa ja dokumentoida vaatimukset kehitykseen sopivalla tavalla. Taulukossa 1 on esitetty työn tehtäviin sisältyneitä toimintoja.

Taulukko 1. Opinnäytetyön tehtävät ja niiden osa-alueet

Järj. nro	Tehtävät	Osa-alueet
1	Olemassa olevan aineiston kartoitus	• dokumenttien kokoaminen • dokumenttien läpikäynti
2	Metropolian asiantuntija-haastattelut	• ajankohtien sopiminen osapuolten kanssa • haastatteluiden laatiminen • toteutus • dokumentointi
3	Vaatimusten analysointi	• esitettävän aineiston laatiminen • osallistuminen neuvotteluun projektin ohjausryhmän kanssa
4	Dokumentointi	• vaatimusten kokoaminen aineistoista • käyttäjätarinoiden kirjoittaminen • vaatimustaulukon kehittäminen • vaatimustaulukkoa tukevien dokumenttien laatiminen

Projektin tuotoksena syntyi HHTrainee ohjelmiston vaatimusmäärittelydokumentti, joka tukee ohjelmistokehitystä. Vaatimusmäärittelydokumenttia ei julkaista kokonaisuudessa tämän raportin yhteydessä, mutta tehtävä ja tuotos on kuvattu esimerkkien avulla luvussa 6.

Tässä raportissa esitetään lisäksi HHTrainee hankkeen taustaa. Hankkeen projektisuunnitelmassa kuvattua taustaa on täydennetty tässä raportissa niiltä osin, joiden on katsottu hyödyttävän kehitettävän ohjelmiston vaatimusmäärittelyä, sen jatkokehitystä ja ylläpitoa. Tiedot on saatu keskusteluissa projektin ohjausryhmän kanssa. Lisäksi tässä raportissa analysoidaan toteutetun vaatimusmäärittelyn käytäntöjä sekä työn merkitystä HHTrainee hankkeelle. Tässä projektissa käytettyjä vaatimusmäärittelyn menetelmiä reflektoidaan myös yleisiin teoreettisiin menetelmiin, joita on esitetty raportissa. Tämä raportti pyrkii siis perustelemaan, miksi toteutetussa vaatimusmäärittelyssä on toimittu valitulla tavalla, sekä pohtimaan valittujen käytäntöjen sekä niiden lopputulosten hyötyjä ja kehittämiskohteita.

Raportin lähdeviittauksissa on käytetty Mendeley viitteidenhallintasovelluksen automatiikkaa.

1.3 Tehtävän rajaus

Vaatimusmäärittelyprojektin tärkeimpänä tehtävänä on saattaa käyttäjiltä kerätyt toiminnalliset vaatimukset ohjelmistoprojektin tietoon, ei esittää teknisiä ratkaisuja niiden toteuttamiseksi. Vaatimusten tekniseen toteutettavuuteen kiinnitettiin huomiota vaatimusten analysoinnissa. Ohjelmiston teknologiaa, arkkitehtuuria, integraatioita ja rajapintoja koskeva ohjelmistosuunnittelu on kuitenkin pääasiassa ohjelmiston kehittäjäorganisaation vastuulla, siksi työ keskittyy toiminnallisiin ja käyttäjävaatimuksiin. Lisäksi tietoturvaa koskevia viranomaisvaatimuksia tai muita ei-toiminnallisia vaatimuksia ei kartoiteta laajasti tässä projektissa.

HHtrainee hankkeessa ei suunnitella uutta harjoitteluprosessia, joten myöskään vaatimusmäärittelyssä ei voida tehdä toimintamalleihin liittyviä päätöksiä. Mahdolliset vaatimuksissa esiintyvät toimintamallien ristiriidat esitetään ohjelmistoprojektin ohjausryhmälle, jotta ne voidaan esittää edelleen toimintamalleista päättävälle taholle ratkaistavaksi.

2 Keskeiset käsitteet

Hanke: Suuri projekti, jolla voi olla useampia tavoitteita ja se voi sisältää erillisiä projekteja.

(Tieto)järjestelmä: Tietyllä tavalla toimiva kokonaisuus, jonka avulla voidaan suorittaa jokin (tietojenkäsittely)toiminto. Järjestelmä koostuu osista, kuten ohjelmistot, laitteet tai henkilöt, joilla on keskinäisiä yhteyksiä ja yhteyksiä järjestelmän ulkopuolisiin kohteisiin (JUHTA, 2009).

Katselmointi: Vaatimusten laadun arviointia keskustelemalla, analysoimalla ja tunnistamalla puutteita sekä sopimista toimenpiteistä, joilla havaitut puutteet saadaan poistettua (JUHTA, 2009).

Ohjelmisto: Yksi tai useampia (tietokone)ohjelmia, jotka sisältävät käskyjä, joilla voidaan suorittaa tietokoneen laitteistolla haluttuja toimintoja.

Priorisointi: Asioiden asettaminen tärkeysjärjestykseen.

Projekti: Kertaluonteinen työkokonaisuus, jolla on tietty päämäärä.

Prosessi: Järjestelmällinen toimintatapa tietyn tavoitteen saavuttamiseksi.

Toimija: Toimija liittyy yhteen tai useampaan käyttötapaukseen ja yhteen käyttötapaukseen liittyy aina vähintään yksi toimija (JUHTA, 2009).

Vaatimus: Toiminto, jonka ohjelmistolla pystyy tekemään tai ominaisuus, joka ohjelmistolla tulee olla.

3 HHTrainee hankkeen tausta ja tavoite

Valtioneuvoston asetuksen (1129/2014) 2§:n mukaan ammattikorkeakoulututkintoon johtaviin opintoihin kuuluu työharjoittelu. Haaga-Helia on ammattikorkeakoulu, joka kouluttaa liike-elämän ja palveluelinkeinojen asiantuntijoita. Koulussa opiskelee noin 11 000 opiskelijaa (Haaga-Helia, 2021). Opiskelijoiden suorittamien harjoittelujaksojen määrä ja pituus voivat vaihdella koulutusohjelmakohtaisten tai opiskelijoiden henkilökohtaisten opetussuunnitelmien mukaan, mutta suuri osa työharjoitteluista sisältää kuitenkin saman kaltaisina toistuvia päätoimintoja, kuten harjoittelupaikan ilmoittaminen oppilaitokselle, harjoittelun raportointi sekä harjoittelusuorituksen arviointi. Harjoitteluprosessiin liittyy harjoittelua ohjaavan opettajan osalta paljon rutiininomaista työtä, jota halutaan Haaga-Heliassa helpottaa toimintoja yhtenäistämällä ja automatisoimalla niitä kehitettävän sähköisen työkalun avulla.

Haaga-Heliassa on myös tunnistettu harjoitteluprosessissa tuotetuista raporteista saatavan tiedon potentiaali lisätä opetusta suunnittelevien tahojen ymmärrystä työelämän osaamistarpeista. Toistaiseksi tietoa ei ole voitu täysin hyödyntää, koska sopivaa työkalua tietojen keruuseen ei ole ollut. Lisäksi on todettu, että ilman harjoittelutietoa kokoavaa järjestelmää, työharjoittelun ympärille rakentuvaa kokonaisuutta, alkaen opetuksen suunnittelusta aina yksittäisen työharjoittelun edistymisen seurantaan asti, on vaikea hallita. Näkyvyyden puutteella voi olla vaikutusta myös valmistuvien opiskelijoiden märän ja oppilaitoksen toiminnan ennakointiin.

Oppilaitoksen sähköisenä kurssialustana toimiva Moodle sovellus mahdollistaa esimerkiksi tehtävien tai raporttien palautuksen ja seurannan, mutta ei ratkaise tiedon keräämiseen liittyvää ongelmaa eikä se toisaalta tarjoa riittävästi tukea yhtenäiselle harjoittelun ohjaus- ja seurantaprosessille. Hankesuunnittelussa on vertailtu myös muita jo olemassa olevia ohjelmistoja, mutta niissä on katsottu olevan puutteita tiedon keruussa tai muissa eritysesti korkeakoulun tarpeiden vastaavuudessa. Näin ollen on Haaga-Heliassa päätetty kehittää kokonaan uusi ohjelmisto vastaamaan korkeakoulujen digitalisoidun työharjoitteluprosessin tarpeisiin. HHTrainee hankkeessa kehitettävän ohjelmiston keskeisin tarkoitus on helpottaa harjoittelukoordinaattoreiden, ohjaajien ja opiskelijoiden työtä ja tuottaa ajantasaista tilastotietoa työelämän tarpeista korkeakouluyhteisölle. Hankkeen kehitystyössä hyödynnetään koulun omia resursseja. Kehitystä on tehty kahden lukukauden ajan opiskelijaprojektityönä, jossa ryhmä oppilaitoksen tietojenkäsittelyn opiskelijoita toimivat ohjelmistokehittäjinä.

Hankkeen tavoitteena on tuottaa ohjelmisto, jota voidaan käyttää myös muissa korkeakouluissa, jotta sille saadaan mahdollisimman laajasti käyttäjiä myös Haaga-Helian

ulkopuolelta. Ohjelmistotuotteen yhtä oppilaitosta laajempi käyttö parantaisi mahdollisuuksia sen tehokkaaseen ylläpitoon tulevaisuudessa. Hankkeessa onkin kehityksen alusta asti pyritty huomioimaan korkeakoulujen tarpeita Haaga-Heliaa ja sen koulutusohjelmia laajemmin ja mukaan suunnitteluun on otettu Metropolia ammattikorkeakoulun työharjoitteluasiantuntijoita tiedonantajiksi. Myös Metropoliassa on havaittu saman tyyppisiä harjoitteluprosessin digitalisoinnin puutteesta johtuvia ongelmia kuin Haaga-Heliassa. Toisaalta Metropoliassa on saatu myös konkreettista kokemusta ohjelmiston avulla tuotetun harjoitteluprosessin hyödyistä, sillä osassa sen koulutusohjelmista on käytössä Metropolian tekniikan koulutusohjelmien tarpeisiin kehitetty työharjoittelun ohjaus- ja seurantajärjestelmä. Tämä järjestelmä ei kuitenkaan palvele kaikkia Metropolian koulutusaloja ja lisäksi se ei vastaa riittävästi nykyaikaisen ohjelmiston vaatimuksia, joten Metropoliassa on tarve uudistaa järjestelmä pian.

HHTrainee ohjelmistohankkeessa toteutetaan myös tutkimusta, jonka yhtenä tavoitteena on aiemmassa ohjelmistoprojektissa kehitetyn digitaalisten ratkaisuiden kehittämismallin soveltaminen ja hiominen. Aiemmin kehitetyn toiminnan digitalisoinnin mallin periaatteita on hyödynnetty vaatimusmäärittelyssä ja pohdittu niiden soveltuvuutta HHTrainee hankkeeseen.

HHTrainee ohjelmistohankkeen aikana on Haaga-Heliassa samanaikaisesti käynnissä kehityshanke, jossa kehitetään työharjoittelun ohjauksen ja seurannan toimintamalleja. HHTrainee hankkeen tehtävänä ei lähtökohtaisesti ole prosessin uudelleen suunnittelu, mutta ohjelmistoa koskevilla vaatimuksilla voi olla yhteys kokonaisprosessin toimintoihin ja ohjelmiston ratkaisut voivat vaikuttaa prosessin kulkuun. Prosessin kehitystyössä voidaan hyödyntää ohjelmiston vaatimusmäärittelyssä esiin nostettuja vaatimuksia. Toisaalta myös prosessin kehityksessä tehdyt päätökset vaikuttavat ohjelmistovaatimuksiin, siksi HHTrainee hankkeen tavoitteena on pitää yllä kommunikaatiota prosessinkehityshankkeen kanssa.

4 Lähestymistapoja ohjelmistoprosessiin

Sommervillen (2011, 28) mukaan ohjelmistoprosesseja voidaan luokitella useilla eri tavoilla, mutta ne sisältävät aina neljä perustoimintoa

1. Ohjelmiston toimintojen ja rajoittavien tekijöiden määrittely (*specification*).
2. Määrittelyä vastaava ohjelmiston suunnittelu ja toteutus (*design and implementation*).
3. Ohjelmiston validointi (*validation*) eli toiminta, jolla vahvistetaan, että tuote vastaa asiakkaan vaatimuksia.
4. Ohjelmistoa ylläpitävä kehitys (*evolution*) muuttuvien vaatimusten mukaan. (Sommerville, 2011, 28.)

Tavallisesti ohjelmiston tuottamisen prosessi toteutetaan projektina, jonka sidosryhmiä ovat asiakas eli tilaaja sekä toimittaja. Asiakas esittää vaatimukset ja toimittajan tehtävä on ratkaista asiakkaan ongelmat ohjelmistotuotteen avulla (Haikala & Mikkonen, 2011, 19).

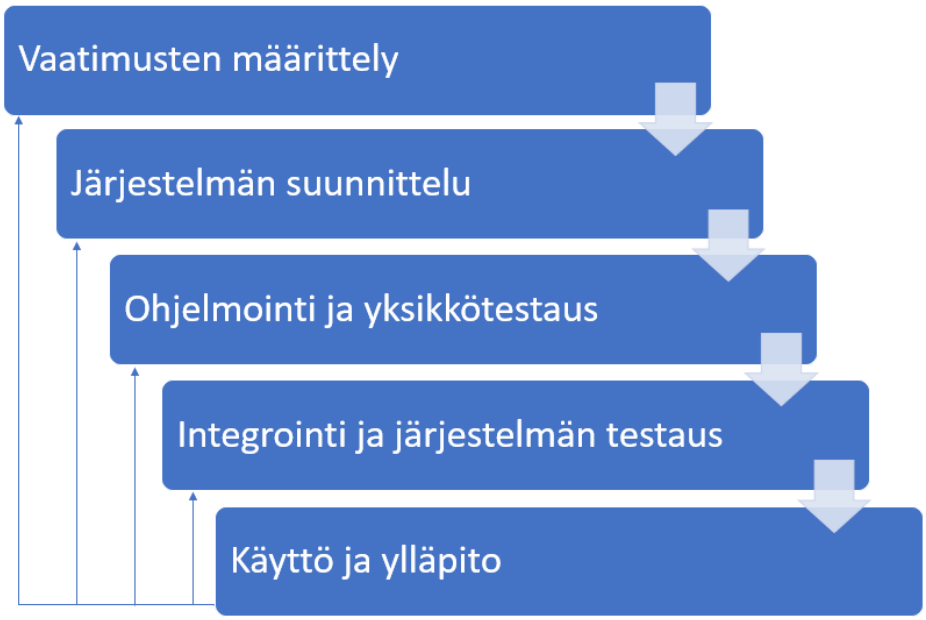
Projektinhallinnalla tarkoitetaan toimintoja, joilla organisoidaan projektin resursseja kuten työvoimaa, rahaa ja aikaa niin että projektin tavoite täyttyy. Projektinhallinta ulottuu kaikkiin projektin toimintoihin, jotka voivat olla toisistaan erillisiä tai niiden välillä voi olla riippuvuuksia, kuten ohjelmistoprosessin vaiheissa. Monisäikeisyys tekee projektinhallinasta vaikeaa ja sitä helpottamaan on kehitetty useita erilaisia projektinhallintamalleja. Tässä luvussa on esitetty kaksi yleistä ohjelmistotuotannon projektityömallia, suunnitelmakeskeinen vesiputousmalli ja ketterä malli. Näitä malleja voidaan pitää ennemminkin kehyksinä tai ohjelmistoprosessin lähestymistapoina, joista on edelleen kehittynyt lukuisia erilaisia standardoituja ohjelmistokehitysmalleja (Sommerville, 2011, 29). Edellä mainittujen lisäksi kuvataan Haaga-Heliassa kehitetyn digitaalisten ratkaisuiden kehittämismallia (Expert Oriented Digitalization Model), joka määrittelee ohjelmistoprosessin yhdeksi osaksi liiketoiminnan prosessin digitalisointia (Lagstedt, Lindstedt & Kauppinen, 2020).

4.1 Vesiputousmalli

Vesiputousmenetelmä on lineaarinen malli, joka koostuu itsenäisistä, peräkkäisistä vaiheista, vaatimusten määrittely, järjestelmän suunnittelu, ohjelmiston toteutus, todentaminen ja ylläpito. Vesiputousmalli on suunnitelmakeskeinen ja sen periaatteisiin kuuluu vaiheiden tarkka suunnittelu ennen varsinaisen työn aloittamista sekä kunkin vaiheen dokumentointi ja hyväksyminen ennen seuraavaan siirtymistä. (Sommerville, 2011, 31-32)

Tieto liukuu vesiputousmaisesti suoraan alaspäin vaiheesta seuraavaan (kuva 1). Työn toteutuksessa edetään tarkasti suunnitelman mukaan.

Mallin tehokkuutta perustellaan sillä, että huolellisella suunnittelulla virheet huomataan ja korjataan vaiheen sisällä, eikä niitä siirretä niitä vesiputouksessa myöhempiin vaiheisiin, jolloin korjaus on mahdollisesti laajempi ja kalliimpi toimenpide. Suunnitelmallisuuden vuoksi ohjelmiston tilaaja voi myös ennustaa resurssitarvetta ja seurata projektin edistymistä peilaamalla toteutusta suunnitelmaan (Sommerville, 2011, 29).



Kuva 1. Vesiputousmalli (mukaillen Sommerville, 2011, 30)

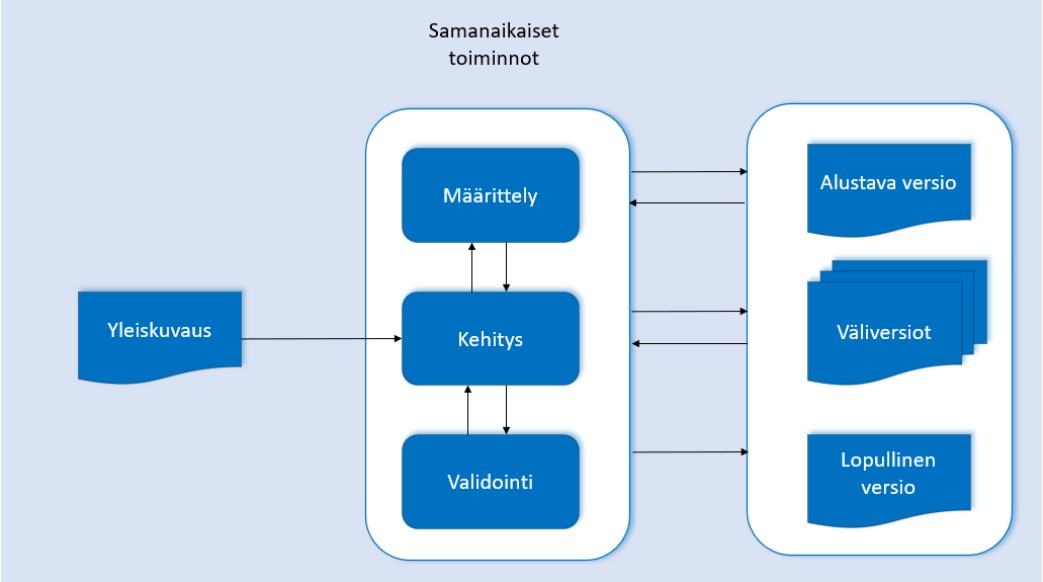
Menetelmän haasteena on, että se ei huomioi mahdollisia virheitä, joita voi suunnitelmista huolimatta sattua, esimerkiksi käyttäjän esittämä vaatimus saatetaan ymmärtää väärin. Vesiputousmallissa ei myöskään oteta huomioon, että vaatimukset saattavat muuttua ohjelmistoprosessin aikana. (Leffingwell, 2011, 6)

4.2 Ketterät kehitysmallit

Iteratiivisessa ja inkrementaalisessa kehityksessä ohjelmistoa kehitetään tuotantosykleissä eli iteraatioissa laajentaen toimintoja vähitellen. Ohjelmistosta julkaistaan useita versioita, joista saadun käyttäjien palautteen avulla kehitetään seuraavia versioita niin kauan, kunnes saavutetaan haluttu lopputulos. (Kuva 2)

Useat nykyisin käytössä olevat ketterät (*agile*) kehitysmallit ovat iteratiivisuuteen ja inkrementaalisuuteen perustuvia malleja. Ketterä malli ei ole valmiita menetelmiä määrittelevä standardi, vaan joukko periaatteita, joihin pohjautuen on kehitetty useita menetelmiä kuten Scrum, Kanban ja SAFe. Periaatteet esimerkiksi kehottavat korostamaan toimivaa

ohjelmistoa enemmän kuin kaiken kattavaa dokumentointia sekä muutokseen vastaamista enemmän kuin suunnitelman noudattamista (agilemanifesto.org s.a.).



Kuva 2. Inkrementaalisen kehityksen toiminnot (mukaillen Sommerville, 2011, 33)

Ketterässä kehityksessä järjestelmän määrittely-, kehittämis- ja validointitoiminnot kulkevat kehityksessä rinnakkain toisin kuin vesiputousmallissa ja tieto toimintojen välillä on nopeaa ja reaaliaikaista. Ohjelmointi voidaan aloittaa nopeasti, koska tarkasti etukäteen laaditun vaatimusmäärittelyn valmistumista ei tarvitse odottaa. Tuotantosyklin päätyttyä julkaistun ohjelmistoversion avulla voidaan edelleen kartoittaa ja tarkentaa vaatimuksia käyttäjien kokemukseen perustuvan palautteen avulla. Ohjelmisto ja sen tuomat hyödyt liiketoiminnalle saadaan näin ollen nopeasti käyttöön ja ohjelmistoprosessissa voidaan reagoida muuttuviin vaatimuksiin joustavasti ohjelmiston kehityksen aikana.
(Haikala & Mikkonen, 2011, 40-42; Sommerville, 2011, 32-33)

4.3 Hybridimallit

Ketterät menetelmät ovat kehittyneet vastaamaan vanhanaikaisena pidetyn vesiputousmallin haasteisiin. Olennaisten vaatimusten huomiointi ohjelmistosuunnittelussa kuitenkin edellyttää, että ne on tiedettävä etukäteen, siksi vaatimusmäärittelyä ei voida tehdä täysin inkrementaalisesti (Sommerville, 2011, 30). Hybridimallit yhdistävät eri kehitysmallien piirteitä. Erityisesti suurissa järjestelmähankkeissa tämä on järkevää. Myös suurempien järjestelmien yksittäisiä osia voidaan kehittää toisistaan eroavia kehitysmalleja hyödyntäen. Osissa, jotka ovat helposti etukäteen määriteltävissä, voidaan käyttää vesiputousmallia ja taas ne osat, joita ei voida täysin etukäteen määritellä, kuten esimerkiksi käyttöliittymät, tulisi kehittää ketterästi (Sommerville, 2011, 30).

Organisaatiot kehittävät omia projektityömallejaan, jotka voivat hyödyntää osia sekä ketterän, että vesiputousmallin menetelmistä. Malleissa voidaan huomioida organisaation ja projektin yksilökohtaisia tarpeita ja mallien kehittämisessä hyödynnetään aiempien hankkeiden kokemuksia.

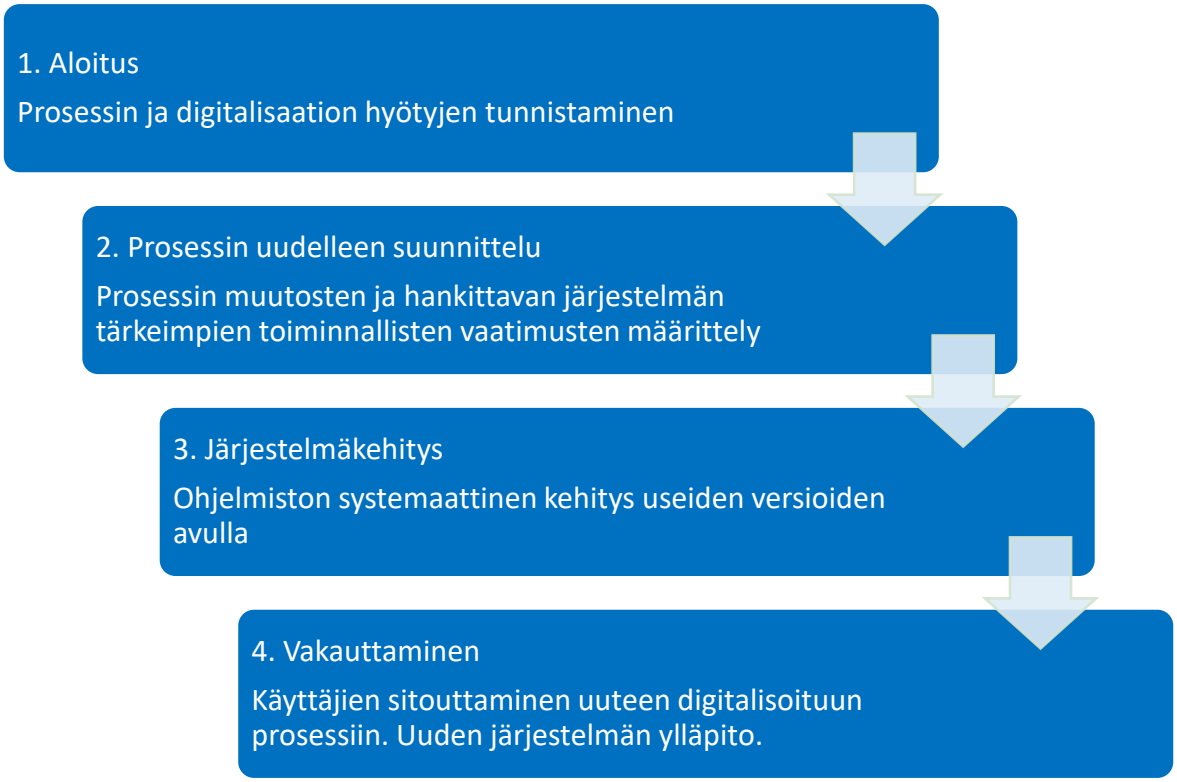
4.4 Digitalisaatio ja EXOD malli

Digitalisaatio ei tarkoita vain ohjelmistojen hankintaa ja käyttöönottoa vaan siihen liittyy myös ihmisten käyttäytymisen ja prosessin toimintojen muutosta. Toimintojen digitalisointi, eli muuttaminen analogisesta digitaaliseksi, ei itsessään ole tärkeää, vaan kehityksen tuomat hyödyt liiketoiminnalle. Aktiivisessa digitalisaatiossa yritys ei vain sopeudu muuttuvaan toimintaympäristöön vaan muuttaa itse toimintamallejaan digitalisaatiota hyödyntäen. (Ilmarinen & Koskela, 2015, 22-25)

Haaga-Heliassa on kehitetty prosessien digitalisoinnin mallia, jossa on huomioitu käyttäjien rooli korkeakoulun prosessien asiantuntijoina, heidän autonomiansa koulutusprosesseissa sekä toimintatapojen varianssi käyttäjien välillä. Expert Oriented Digitalization Model eli EXOD-malli pyrkii tiiviiseen vuorovaikutukseen asiantuntijoiden kanssa prosessin digitalisoinnissa. Tällä pyritään hyödyntämään käyttäjien prosessituntemusta sekä sitouttaa käyttäjiä uuteen digitalisoituun prosessiin tarjoamalla heille mahdollisuus vaikuttaa kehitettävään ohjelmistoon ja välittämällä tietoa sen hyödyistä. Mallin tavoitteena on kehittää sekä toimintaa että ohjelmistoa. Ohjelmistokehityksen lopputuloksessa pyritään ratkaisuun, joka tarjoaa tuen prosessille mahdollisimman yksikertaisella tavalla. (Lagstedt, Lindstedt & Kauppinen, 2020)

EXOD malli hyödyntää toiminnan digitalisointiin periaatteita sekä suunnitelmakeskeisestä että ketterästä mallista. Prosessin tunnistamisen vaiheessa edetään suunnitelmallisemmin ja ohjelmistokehityksen vaiheessa toteutetaan suunnitelmiin perustuva ohjelmisto ketterän ohjelmistoprosessin menetelmiä hyödyntäen. Ohjelmiston julkaisu tehdään useina versioina ja niistä kerättyjen käyttäjien palautteiden avulla kehitystä jatketaan, kunnes haluttu lopputulos on saavutettu. (Lagstedt, Lindstedt & Kauppinen, 2020)

Kuvassa 3 on esitetty EXOD mallin vaiheiden kulku ja sisältö tiivistetysti.

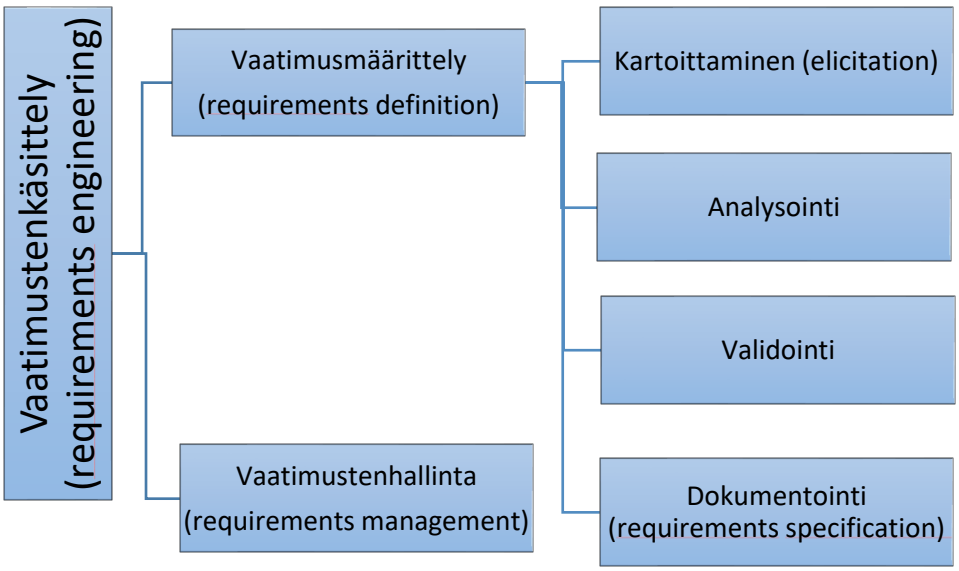


Kuva 3. EXOD mallin vaiheet (mukaillen Lagstedt, Lindstedt & Kauppinen, 2020)

5 Vaatimusmäärittelyn käytäntöjä

Vaatimukset voidaan jakaa karkeasti kahteen eri luokkaan. Toiminnallisiin ja ei-toiminnallisiin vaatimuksiin. Toiminalliset vaatimukset kertovat mitä toimintoja ohjelmistossa tulisi olla, miten sen tulee reagoida käyttäjän syötteisiin ja miten sen tulisi toimia tietyissä tilanteissa. Käyttäjävaatimukset ovat toiminnallisia vaatimuksia, jotka kuvataan usein korkeammalla abstraktiotasolla ja ne kertovat nimensä mukaan käyttäjän odotuksista. Ei-toiminnalliset vaatimukset ovat ohjelmiston toimintoja rajoittavia tekijöitä. Niihin sisältyy esimerkiksi vasteaikoihin, rajapintoihin tai turvallisuussäädöksiin liittyviä vaatimuksia. (Sommerville, 2011, 84-87).

Vaatimuksiin liittyvään prosessiin sisältyy paljon terminologiaa, jota voidaan myös tulkita eri tavoin. Englannin kielinen kattotermi *requirements engineering* käännetään usein suomenkielelle vaatimusmäärittelyksi, joka taas voidaan tulkita vaatimukset esittäväksi dokumentiksi, joka syntyy vaatimusten määrittelyn tuotteena (Vilppola & Terho, 2008, 14) tai vaatimuksiin liittyvän toiminnan osa-alueeksi (Haikala & Mikkonen, 2011, 65). Haikala & Mikkonen (2011, 61) käyttää kattotermistä käännöstä *vaatimustenkäsittely*, joka jakautuu vaatimusmäärittelyyn (*requirements definition*) ja vaatimustenhallintaan. Kuvassa 4 on esitetty vaatimustenkäsittelyn osa-alueet. Tämän raportin lähteenä on käytetty sekä suomen- että englanninkielisiä julkaisuja. Niiden tulkinnassa on pyritty säilyttämään toimintoja kuvaavien termien johdonmukaisuus kuvan luokittelun mukaan.



Kuva 4. Kuvaus vaatimustenkäsittelyyn kuuluvista toiminnoista (mukaillen Haikala & Mikkonen, 2011, 65)

Kuten luvussa 4 on esitetty, vaatimusmäärittely kuuluu yhtenä osana ohjelmistoprosessiin, siksi sen tehtäväksi voidaan katsoa ohjelmistoprosessin muiden toimintojen tukeminen.

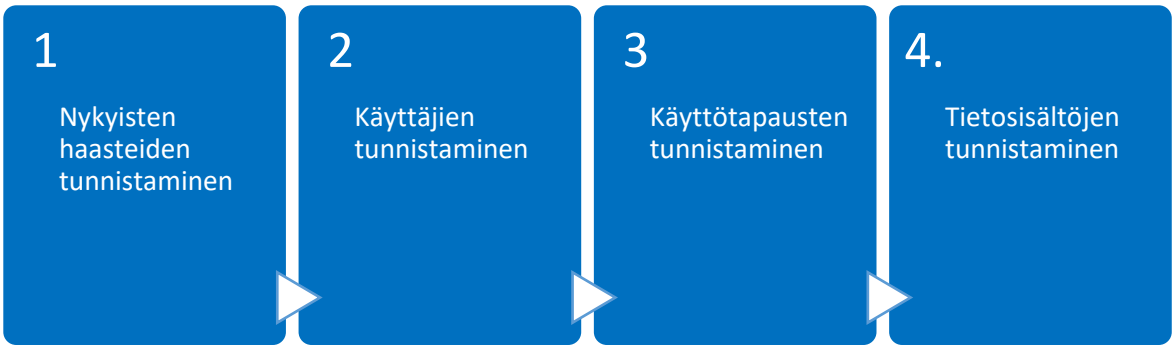
Vaatimusmäärittelyllä on kuitenkin tehtävä myös asiakkaan ohjelmistohankinnan tukemisessa. Vilppola & Terhon (2008, 14) mukaan vaatimusten määrittelyllä tarkoitetaankin myös niitä alustavia toimenpiteitä, joita yritys tekee tietojärjestelmiä hankkiessaan ennen kontaktia järjestelmän toimittajaan. Vaatimusmäärittelyä voidaan pitää ohjelmiston käyttäjä- ja kehittäjäosapuolten kommunikoinnin välineenä.

Vesiputousmallissa vaatimusmäärittely suoritetaan kokonaisuudessaan alusta loppuun ennen ohjelmiston suunnittelu- ja kehittämisvaiheita ja vaatimusten muutostarpeet kartoitetaan vasta kun ohjelmisto on valmis. Vaatimusten määrittely vain tietyn ajanjakson ajan voi johtaa tilanteeseen, jossa kehitysprosessin aikana tapahtuvat muutokset eivät heijastu vaatimuksiin, ja koska vaatimusten määrittely ei dokumentoi realisoituja vaatimuksia, niitä ei voida käyttää esimerkiksi ohjelmiston ylläpidossa tai muissa kehitysprojektissa. (Pohl, 2010, 35-37).

Ketterissä menetelmissä vaatimusmäärittely on koko ohjelmistoprosessin ajan jatkuvaa toimintaa. Sekä asiakkaat että kehittäjät ovat jatkuvasti mukana vaatimusten kartoittamisessa ja tarkentamisessa (Laplante, 2014,172). Jatkuvalla vaatimusmäärittelyllä sekä vaatimusten hallinnalla toteutetaan ketterien menetelmien periaatetta vaalia muuttuvia vaatimuksia koko kehityksen ajan.

5.1 Vaatimusmäärittelyn vaiheet

Vaatimusmäärittely alkaa vaatimusten kartoittamisella, jossa vaatimukset nostetaan esiin esimerkiksi tulevia käyttäjiä haastattelemalla. Kartoituksessa selvitetään nykyinen prosessi ja sen haasteet. Tietojärjestelmähankinnan taustalla on usein jokin ongelma kuten tiedon pirstaleisuus ja siitä aiheutuva prosessin tehottomuus. Yhden toiminnon haaste voi olla seuraus jostain prosessin aiemmassa vaiheessa tapahtuvasta toiminnasta ja siksi ongelmien laajempi tunteminen on tärkeää. Tulevaa järjestelmää voidaan myös myöhemmissä vaiheissa tarkastella ongelmien ratkaisemisen kannalta. Tietojärjestelmän vaatimusten tunnistamisenprosessi voidaan jakaa neljään toisiaan seuraavaan vaiheeseen, kuten kuvassa 5 on esitetty. Ongelmien ja vaatimusten tarkkuus lisääntyy vaihe vaiheelta. (Vilppola & Terho, 2008, 16)



Kuva 5. Vaatimusten tunnistamisprosessi (mukaillen Vilppola & Terho, 2008, 16)

Vaatimusten analysointivaiheessa vaatimuksia voidaan tarkentaa, priorisoida ja tuoda esiin esimerkiksi niiden välisiä suhteita ja riippuvuuksia. Validointivaiheessa vaatimukset katselmoidaan, tavallisesti katselmointi tehdään sekä liiketoiminta- että kehittäjäorganisaatioiden kesken. (Haikala & Mikkonen, 2011, 66-67). Vaatimusten dokumentointi on vaatimusten saattamista kirjalliseen muotoon, dokumentointivaihetta on selitetty tarkemmin luvussa 5.2.

5.2 Vaatimusten dokumentointi

Dokumentointivaiheessa vaatimukset mallinnetaan ja dokumentoidaan sopivalla tavalla esimerkiksi Excel taulukkoon tai vaatimustenhallintajärjestelmään (Haikala & Mikkonen, 2011, 67). Ohjelmiston määrittelyä voidaan tehdä monella eri abstraktiotasoilla, joista korkeimmalla tasolla olevien tarkoitus on kuvata mitä ollaan tekemässä, eikä niissä ole tarkoitus ottaa kantaa toteutustekniikoihin, paitsi niiltä osin kuin tekniikat asettavat reunaehtoja toteutettavalle ohjelmistolle (Haikala & Mikkonen, 2011, 70).

Ohjelmiston määrittelynotaatioita on lukuisia ja niitä on kehitetty vastaamaan erilaisia kehitysprojekteja. UML (Unified Modeling Language) on standardoitu ohjelmistojen määrittelynotaatio, johon sisältyy useita eri nykyaikaisen ohjelmiston kuvaamiseen käytettyjä kaaviotekniikoita (Haikala & Mikkonen, 2011, 72).

Ketterissä menetelmissä dokumentoinnin määrä pyritään pitämään minimaalisena, jotta dokumenttien kirjoittamisesta, lukemisesta ja ylläpidosta säästyy aikaa itse ohjelmiston kehitykseen. “The best documentation is the simplest that gets the job done.” (Ambler s.a.). Useisiin ohjelmistoprojekteihin ei välttämättä tarvitakaan koko UML standardin kattavuuden mukaista määrittelyä, vaan valittu osajoukko standardin kaavioista riittää (Haikala & Mikkonen, 2011, 72). Käyttäjävaatimukset kirjoitetaan tavallisesti luonnollisella kielellä ja niitä täydennetään asianmukaisilla kaavioilla vaatimusmäärittelydokumentissa (Sommerville, 2011, 95).

5.2.1 Use case eli käyttötapaus

Use case eli käyttötapaus tarkoittaa perättäisiä toimintoja, jotka suoritetaan jonkin päämäärän saavuttamiseksi. Käyttötapauksia voidaan kuvata eri tarkkuustasoilla esimerkiksi käyttötapauskaaviona, joka osoittaa ohjelmistoon sisältyvät pääkäyttötapaukset ja niihin liittyvät toimijat, tai tarkkoina käyttötapauskuvauksina, jotka kuvaavat käyttäjän ja järjestelmän, tai kahden järjestelmän, välistä vuorovaikutusta. (JUHTA, 2009).

Käyttötapauskaavio on UML standardiin kuuluva yksinkertainen ja tehokas kaaviotekniikka. Se esittää ohjelmiston käyttötapaukset ja toimijat sekä näiden välisen yhteyden. Käyttötapauskuvausta ei ole UML:ssa standardoitu, mutta yksi tapa on kuvata käyttötapauksen toiminta toimijan ja ohjelmiston välisenä interaktiona. (Haikala & Mikkonen, 2011, 77-80)

5.2.2 User story eli käyttäjätarina

Käyttäjätarinat ovat erityisesti ketteriin ohjelmistoprojekteihin kehitetty tapa kuvata vaatimuksia. Käyttäjätarina kuvaa toiminnallisuutta, joka on arvokasta joko ohjelmiston käyttäjälle tai muulle hankkijaosapuolen sidosryhmälle (Cohn, 2010, 4). Käyttäjätarina on siis tapa esittää ohjelmistovaatimuksia niin, että varsinaisen halutun toiminnon tai ominaisuuden lisäksi ilmaistaan sen tuottama arvo käyttäjälle.

Käyttäjätarinat ovat käyttötapausten tavoin helposti ymmärrettäviä sekä ohjelmiston hankkija- että kehittäjäosapuolille ja siksi ne ovat tehokas vaatimusten ilmaisumuoto. Erityisesti ketterässä ohjelmistokehityksessä hyvin laaditut käyttäjätarinat sopivat pienen kokonsa vuoksi suunnittelu- ja kehitysiteraatioihin. Yksi kehitysiteraatio voidaan esimerkiksi toteuttaa vain muutaman käyttäjätarinan avulla. (Cohn, 2010, 145-149)

6 Opinnäytetyön tehtävän toteutus

Vaatimusmäärittelyn tavoite on tukea ohjelmistokehitystä ja sen keskeisin tehtävä on osoittaa ohjelmistokehittäjille, miten kehitettävän ohjelmiston tulisi toimia. Jotta ohjelmistoprosessissa voidaan edetä tehokkaasti ja järjestelmällisesti on katsottu tarpeelliseksi kartoittaa ohjelmistoa koskevat toiminnalliset vaatimukset laajasti, sillä prosessissa tiedetään olevan selkeitä toistuvia toimintoja, jotka ohjelmistolla tulee voida suorittaa. Vaatimusten tarkka määrittely ennen itse toiminnallisuuden suunnittelua ja toteutusta edesauttaa ohjelmiston arkkitehtuuriratkaisuiden ja teknologioiden suunnittelua, kuten kuvaus suunnitelmakeskeisyyden hyödyistä osoittaa (luku 4.1.).

Vaatimusmäärittelyprojekti on toteutettu yhdessä HHTrainee hankkeen tuoteomistajan ja projektijohtajan kanssa, jotka ovat olleet tiiviisti osallisina vaatimusten kartoituksen suunnittelussa ja vaatimusten analysoinnissa. Tuoteomistajan vastuulla oli siirtää vaatimusmäärittelyprojektin tieto HHTrainee hankkeen ohjelmistokehittäjille sekä ylläpitää vaatimusdokumenttia ohjelmistokehityksen edistymisen osalta.

6.1 Tehtävä 1: Vaatimusten kartoitus aiemmin kerätystä aineistosta

Ennen ohjelmistonkehityksen alkamista ohjelmistolle oli laadittu kehityksen tueksi alustava lista vaatimuksista ja tietotarpeet koskien sähköistä lomaketta, jolla opiskelija ilmoittaa harjoittelupaikan ja harjoittelun tiedot, kuten ajankohdan, oppilaitokselle. Vaatimukset perustuivat Haaga-Helian työharjoittelunohjauskäytäntöihin. Toteutusesimerkiksi ja kehityksen tueksi oli kehittäjille jaettu videohaastattelutallenne, jossa Metropolia ammattikorkeakoulun edustajat esittelevät oppilaitoksessa rajatusti käytössä olevaa sähköistä työharjoittelujärjestelmää. Lisäksi ohjelmiston suunnittelun malleina hyödynnettiin aiemmin toteutetun opinnäytetyön ohjausprosessiin kehitetyn ohjelmiston käyttöohjetta sekä tietomallikaaviota. Kehityksen aikana oli kertynyt lisäksi dokumentaatiota kehitetystä ohjelmistosta ja muistiinpanoja kehityksen aikana käydyistä keskusteluista.

HHTrainee ohjelmiston tavoitetta oli käytetty tehtävänannoissa Haaga-Helian vaatimusmäärittelyä käsittelevän kurssin toteutuksilla syksyllä 2020 ja kurssin opiskelijat olivat tehneet haastatteluja Haaga-Helian asiantuntijoille vaatimusten keräämiseksi sekä hyödyntäneet omia kokemuksiaan tehtävissä. Tämän vaatimusmäärittelyprojektin käyttöön luovutettiin yhteensä 12 kurssityötä, joista asiantuntijahaastatteluja oli tehty kuuteen tehtävään. Haastateltavana oli 12 eri asiantuntijaa, jotka työskentelevät työharjoitteluohjaajina tai muissa asiantuntijatehtävissä Haaga-Heliassa eri koulutusaloilla. Kuudessa kurssityössä ei ollut mainintaa oliko vaatimuksia kerätty haastattelemalla vai opiskelijan omaan näkemykseen perustuen. Lisäksi opiskelijoille oli syksyn 2020 aikana tehty opinnäytetyönä

kyselytutkimus, jossa kysyttiin heidän näkemyksiään ja toiveita harjoitteluprosessiin liittyvästä verkkosovelluksesta.

Opinnäytetyön ensimmäisenä tehtävänä oli koota olemassa olevat aineistot ja kerätä niissä esiin nousseet vaatimukset. Kehitykseen liittyvä aineisto, vaatimusmäärittelykurssien tuotokset ja opinnäytetyönä tehty kyselytutkimus hyödynnettiin vaatimusten kartoituksessa. Kaikki aineistot käytiin tarkoin läpi ja niistä saadut ohjelmistovaatimukset muotoiltiin käyttäjätarinoiksi. Ennakkoaineistosta koottiin reilut 200 vaatimusta.

6.2 Tehtävä 2: Metropolian asiantuntijahaastattelut

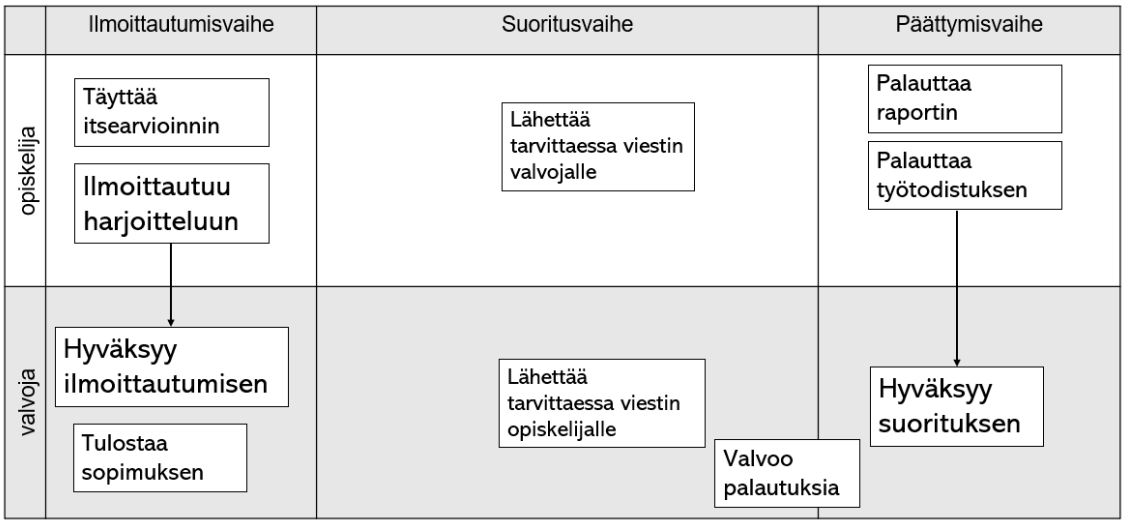
Työn toisena tehtävänä oli haastatella työharjoitteluprosessin asiantuntijoita Metropolia ammattikorkeakoulussa ja täydentää vaatimustaulukkoa haastatteluista saaduilla tiedoilla.

Käyttäjien kuuntelua kehitysvaiheessa pidetään merkittävänä tekijänä korkeakoulujen prosessien kehityksessä (Lagstedt, Lindstedt & Kauppinen, 2020), siksi vaatimusten kartoituksen tueksi tehtiin laajasti haastatteluja tuleville käyttäjille. Lisäksi ohjelmiston kehityksen tärkeänä tavoitteena on kehittää myös muita korkeakouluja palveleva järjestelmä ja sen vuoksi haastatteluihin otettiin mukaan asiantuntijoita myös Haaga-Helian ulkopuolelta. Metropolian edustajilta oli projektin tuoteomistajan aiemmin tekemässä haastattelussa saatu tietoa koulussa käytössä olevasta järjestelmästä, ja Metropoliassa oli osoitettu kiinnostusta HHTrainee ohjelmistoa kohtaan, joten oli luontevaa ehdottaa Metropolian asiantuntijoita haastateltaviksi vaatimusten kartoitukseen.

Yhteydenotto Metropoliaan tapahtui koulun harjoitteluinsinöörien kautta, jotka olivat olleet mukana Metropolian oman järjestelmän kehityksessä ja asiantuntijoina osallistuneet HHTrainee hankkeeseen kertomalla nykyisen järjestelmän toiminnasta ja kehittämiskohteista. Heidän lisäkseen haastatteluihin osallistui neljä asiantuntijaa Metropolian eri koulutusaloilta. Haastateltavia oli siis yhteensä kuusi, joista kaksi oli tekniikan, yksi liiketalouden, yksi kulttuurin, yksi sosiaali- ja yksi terveydenhoitoalan koulutuksen asiantuntijoita. Kukin haastatteluun osallistunut Metropolian asiantuntija toimii koulun opiskelijoiden työharjoitteluohjaajana tai on muutoin työssään tekemisissä harjoitteluprosessin kanssa.

Haastattelut toteutettiin helmikuussa 2021 parihaastatteluina. Parihaastattelusta katsottiin ajankäytön lisäksi olevan etua myös keskustelussa, jolloin toinen haastateltavista voi täydentää ja kyseenalaistaa toisen sanomia. Ohjelmistokehitykselle tärkeiden vaatimusten esiin tuomisen lisäksi vuoropuhelu vastaa myös EXOD – mallin pyrkimyksiin viestiä muutoksesta ja osallistamisen avulla sitouttaa käyttäjiä digitalisaatioon, kuten luvussa 4.4 on esitetty.

Haastatteluiden tarkoitus oli vaatimusten tunnistaminen nykyisiin prosesseihin perehtymällä. Tärkeää oli saada selkeä käsitys Metropolian harjoittelun ohjausprosessin nykyisistä käytännöistä ja työkaluista harjoittelun eri vaiheissa, sekä eri koulutusalojen erityispiirteistä työharjoitteluprosessissa. Haastattelutekniikkana sovellettiin yhdistettyä strukturoitua haastattelua ja avointa keskustelua. Keskustelun rukona käytettiin ennakkotietojen perusteella tehtyjen arvioiden mukaan laadittua prosessin vaihetaulukkoa, jonka avulla keskustelussa edettiin harjoitteluprosessin vaiheiden mukaan. Lisäksi keskustelun tukena käytettiin etukäteen laadittua kysymyslistaa, josta oli mahdollista poimia kuhunkin keskusteluun sopivia täydentäviä kysymyksiä prosessin havainnoimiseksi. Kuvassa 6 on esimerkkinä liiketalouden ja kulttuurin alojen asiantuntijahaastattelussa käytetyn luonnoksen mukainen prosessin vaihekaavio, jossa esimerkiksi opiskelijan toimintoja ovat ilmoittautumisvaiheessa harjoitteluun ilmoittautuminen, suoritusvaiheessa viestin lähetys valvojalle sekä päättymisvaiheessa harjoitteluraportin palauttaminen ja valvojan toimintoja ilmoittautumisen hyväksyminen ilmoittautumisvaiheessa ja päättymisvaiheessa suorituksen hyväksyminen.



Kuva 6. Esimerkki keskustelunrukona käytetystä prosessivaihekaaviosta

Haastattelut käytiin videokeskustelusovelluksen avulla, keskustelun jälkeen puhelun äänitteistä tehtiin muistiot, joissa on pyritty kuvaamaan Metropolian eri alojen harjoitteluprosesseja ja haasteita nykytilassa, sekä kirjaamaan kaikki uuteen järjestelmään liittyvät toiveet. Haastatteluissa esiin nousseet vaatimukset muotoiltiin käyttäjätarinoiksi. Esiin nousseita, aiemmassa kartoituksessa mainitsemattomia vaatimuksia kertyi reilut 30. Haastatteluista kirjoitetut muistiot on jaettu HHtrainee hankkeen ohjausryhmälle tausta- ja lisätietona vaatimustaulukkoon kirjatuille käyttäjätarinoille.

6.3 Tehtävä 3: Vaatimusten analysointi

Projektin kolmas tehtävä oli esitettyjen vaatimusten analysointi mahdollisimman kattavasti. Aineistoista kerätyt vaatimukset esitettiin ohjausryhmälle työpajoissa, jotka pidettiin tehtävän 1 kuvauksessa esitetyn ennakkoaineiston kartoituksen ja tehtävän 2 kuvauksessa eitettyjen haastattelujen jälkeen. Työpajojen keskusteluissa käytiin läpi jokaisen esiin nousseen vaatimuksen tarpeellisuutta ja tärkeyttä käyttäjälle ja prosessille, vaatimuksen toteutettavuutta sekä kuhunkin vaatimukseen liittyviä järjestelmän tai esimerkiksi tietoturvamääräysten rajoittavia tekijöitä. Metropolian asiantuntijahaastatteluista koottuja vaatimuksia verrattiin myös aiemmin esitettyihin ja jo kertaalleen analysoituihin vaatimuksiin, jolloin saatiin aiempaa tarkempi käsitys vaatimuksista. Työpajoissa käydyn analyysin perusteella taulukkoon lisättiin uusia käyttäjätarinoita, jo kirjattuja vaatimuksia tarkennettiin ja epäolennaisiksi katsotut vaatimukset karsittiin pois. Vaatimustaulukkoa taulukkoa kehitettiin keskustelusta saatujen tietojen perusteella ja siitä julkaistiin useampia versioita vaatimusmäärittelyprosessin aikana.

Lopullinen vaatimusten priorisointi sovittiin jätettäväksi tuoteomistajan vastuulle, jotta vaatimustenhallinta projektissa säilyisi tuoteomistajalla myös tämän opinnäytetyöprojektin päätyttyä. Vaatimusten dokumentoinnissa ja analysointineuvotteluissa on pyritty mahdollisimman tarkasti esittämään vaatimusten priorisointiin vaikuttavia tekijöitä.

6.4 Tehtävä 4: Vaatimusten dokumentointi

Projektin neljäntenä tehtävänä oli mallintaa ja dokumentoida vaatimukset HHTrainee ohjelmistokehitykseen sopivalla tavalla. Vaatimusten esittämisessä on käytetty käyttäjätarinoita, joiden tarkoitus ja hyödyt on esitetty kappaleessa 5.2.2. Vaatimusten hallintaan käytetään Excel-taulukko-ohjelmaa. Käyttäjätarinat ja niiden hallinta Excelin avulla oli valittu ohjelmistoprojektin alustavien vaatimusten käsittelyn työkaluiksi jo ohjelmistokehityksen alkaessa, ja uudet vaatimukset tuli saattaa kehittäjätiimin käyttöön mahdollisimman nopeasti, siksi vaatimusten esittäminen jo käytössä olevien työkalujen avulla katsottiin parhaaksi. Työkaluna Excel tiedosto soveltuu vaatimustenhallintaan myös sen helppokäyttöisten taulukko-ominaisuuksien takia. Kehitystiimi hyödyntää omaan sisäiseen vaatimusten hallintaan Jira tehtävänhallintaohjelmistoa. Kehitystiimi voi Jiraa hyödyntäen pilkkoa yhden vaatimustaulukossa esitetyn käyttäjätarinan useampaan pienempään tehtävään kehitykseen sopivalla tavalla, hallita tehtävien edistymistä ja ylläpitää jäljitettävyyttä tehtävästä aina ohjelmakoodiin asti.

Työn edetessä vaatimustaulukosta laadittiin useita versioita. Lopullisessa versiossa vaatimustaulukossa oli tämän projektin päätyttyä 104 riviä ja sarakkeet A-M. Liitteessä 1 on esimerkkinä kuvakaappaus vaatimustaulukon osasta. Jokainen rivi taulukossa vastaa yhtä vaatimusta. Käyttäjätarinan lisäksi taulukon eri sarakkeissa on esitetty esimerkiksi vaatimuksen yksilöivä tunniste, vaatimuksen esittäjän käyttäjärooli, vaatimuksen lähde ja käyttötapaus. Taulukossa 2 on esitetty kaikki vaatimustaulukon sisältävät sarakkeet sekä esimerkit tiedoista. Vaatimustaulukon voi järjestää ja suodattaa Excelin taulukko-ominaisuuksien avulla kunkin sarakkeen mukaan tarvittaessa, joka helpottaa ylläpitoa.

Taulukko 2. Vaatimustaulukon tiedot

sarake	otsikko	kuvaus	Esimerkki
A	ID	Käyttäjätarinan yksilöivä numero	1003
B	Prio	Prioriteetti, prioriteettiluokitukset tuoteomistajan vastuulla	1
C	Käyttäjärooli	Vaatimuksen esittäjän käyttäjärooli	Opiskelija
D	Käyttäjätarina	Vaatimus esitettynä käyttäjätarinana	Haluan tarvittavien henkilötietojen siirtyvän harjoittelupaikkailmoituslomakkeelle opiskelijarekisteristä automaattisesti, jotta minun ei tarvitse etsiä ja täyttää henkilötietojani erikseen.
E	Kuvaus	Tarkempi kuvaus käyttäjätarinasta	Kaikkien opiskelijarekisteristä löytyvien lomaketietojen haku. Kaikkia tietoja ei ole välttämätöntä näyttää, mutta pitää olla siirrettävissä lomakkeen mukaan tulostettaville todistuspohjille.
F	Kommentteja ja huomioita	Käyttäjätarinaan liittyvä huomio	Mitä opiskelijan henkilö- ja opiskelutietoja tallennetaan HHtrainee järjestelmään. Mitä tietoja näytetään lomakekenttinä. Pepistä tulevia tietoja ei voi muokata.
G	Liittyvä tarina	Käyttäjätarinaan liittyvän toisen käyttäjätarinan id, esimerkiksi riippuvuus	1001,1002
H	Hyväksymiskriteerit	Hyväksymiskriteerit, joilla tarina voidaan siirtää tilaan valmis. Esitetään esim. sprintin aloituskokouksessa.	1. Tiedot haetaan opiskelijarekisteristä kirjautumistunnisteen perusteella 2. Tarvittavat tiedot näkyvät opiskelijalle 3.Opiskelijan tunniste tallentuu hakemuksen yhteydessä kantaan
I	Käyttötapaus, ensisijainen	Käyttötapaus, johon käyttäjätarina ensisijaisesti liittyy	Harjoittelupaikkailmoituksen tekeminen
J	Lähde	Vaatimuksen lähde, esim. haastattelu	alustava tuotekehitysjono, ID 1
K	Käyttötapaus, muu	Muut käyttötapaukset, joihin käyttäjätarina voidaan liittää	Harjoittelutietojen katselu
L	Tila	Kehitysvaihe/ tila	kesken
M	Tilaan liitt. kommentti	Tilaan liittyvä kommentti	opiskelijanumero

Vaatimusten välisten suhteiden esittämisen tueksi ohjelmistolle on laadittu käyttötapauskaavio (liite 2). Siinä esitetyt pääkäyttötapaukset, on laadittu esiin nostettujen ja validoitujen vaatimusten perusteella. Kaavio osoittaa ohjelmiston pääkäyttötapaukset ja toimijat, joilla on näkyvyys kyseiseen käyttötapaukseen. Myös vaatimustaulukossa jokaiselle vaatimukselle on esitetty sarakkeessa I käyttötapaus, johon vaatimus voidaan liittää. Käyttötapauksen esittäminen omassa sarakkeessa mahdollistaa taulukon suodattamisen tai järjestäminen käyttötapauksen mukaan. Vaatimuksen käyttötapaus on ilmaistu taulukossa myös värikoodilla. Kaikki vaatimustaulukossa esiintyvät käyttötapaukset löytyvät erillisestä käyttötapauskaaviosta ja taulukkoon merkityn vaatimuksen esittäjän rooli, joka on merkitty sarakkeeseen C, on yhtenevä käyttötapauskaaviossa esiintyvän toimijan kanssa. Osa vaatimuksista voidaan suoraan tai välillisesti liittää useampaan käyttötapaukseen, nämä on huomioitu taulukon K-sarakkeessa.

Käyttötapauskaaviossa katkoviiva tarkoittaa, että käyttäjän näkyvyyteen liittyy rajoituksen tai lisäselvityksen tarve. Esimerkiksi työnantaja osapuolen pääsyä harjoittelutietojen katseluun tulee mahdollisesti rajoittaa, koska harjoittelusta saatetaan kerätä myös sellaisia tietoja, joiden tulee pysyä opiskelijan ja oppilaitoksen välisinä. Kaikki lisäselvitystä vaativat toimijoiden näkyvyydet käyttötapauksiin on esitetty projektin ohjausryhmälle myös erikseen. Käyttötapauksista on lisäksi lyhyt kuvaus vaatimustaulukon sisältävässä Excel-tiedostossa erillisessä taulussa.

Käyttötapausten tunnistamisen tueksi laadittiin myös sidosryhmäkaavio (liite 3). Se osoittaa ohjelmistoon liittyvät sidosryhmät laajemmin ja käyttötapausta yksityiskohtaisemmat toiminnot, joiden kautta näillä on pääsy järjestelmään. Sidosryhmäkaavion nuolet kertovat sidosryhmän luku- ja kirjoitusoikeuksista. Esimerkiksi ohjaajalla on oikeus kirjoittaa eli muuttaa järjestelmän tietoja, kuten lisätä harjoittelupaikkailmoituksen kommentti, mutta opolla eli opinto-ohjaajalla on mahdollisuus ainoastaan opiskelijan tietojen lukuun eli tarkasteluun. Sidosryhmäkaavion katkoviivalla rajatulla alueella on esitetty sidosryhmät, jotka ovat mahdollisesti kehityksen alkuvaiheissa toissijaisia, eli voidaan huomioida priorisoitaessa vaatimuksia. Rajattuihin sidosryhmiin liittyy lisäselvitystarpeita, kuten käyttötapauskaaviossa, ja/tai keskusteluissa on todettu mahdolliseksi, että näihin sidosryhmiin liittyvät toiminnat voidaan hoitaa järjestelmän ulkopuolella.

Sidosryhmäkaaviossa esiintyviä toimintoja on luokiteltu myös harjoitteluprosessin eri vaiheiden mukaan omaan taulukkoon toimintojen tunnistamisen tueksi (liite 4). Taulukossa harjoitteluprosessi on jaettu viiteen vaiheeseen ja toimintoja on luokiteltu vaiheiden mukaan. Taulukkoon on otettu vaikutteita UML standardin aktiviteettikaaviosta, jonka avulla voidaan mallintaa prosesseja. Taulukon valkoiset solut viittaavat sote-koulutusalojen harjoitteluprosessiin, joka poikkeaa muiden koulutusalojen prosessista, toimintoihin liittyy

esimerkiksi työnantajaosapuoli ja toiminnot suoritetaan hieman eri järjestyksessä, johtuen järjestelmän eroista.

Sidosryhmäkaavio ja toimintojen vaiheiden mukainen taulukko on ensisijaisesti laadittu käyttötapausten ja prosessin tunnistamisen tueksi, mutta niitä voidaan hyödyntää myös vaatimusten analysoinnin ja priorisoinnin tukena. Ne on laadittu esittämään vaihtoehtoisia luokittelutapoja ohjelmiston vaadituille toiminnoille ja niitä on mahdollista hyödyntää jatkossa esimerkiksi käyttötapauskuvausten tai aktiviteettikaavioiden esitietoina.

Käyttötapaus- ja sidosryhmäkaavioista sekä vaiheittaisten toimintojen taulukosta tulee huomioida, että ne on laadittu perustuen niiden laatimishetkeen mennessä esitettyihin ja analysoituihin vaatimuksiin ja esimerkiksi niissä esitetyt toimijoiden ja toimintojen väliset suhteet voivat muuttua, esimerkiksi ohjelmistossa tai prosessikehityksessä tehtävien päätösten takia. Tarkkoja käyttäjän ja järjestelmän vuorovaikutusta määritteleviä käyttötapauskuvauksia ei ollut mahdollista laatia tämän projektin aikana, koska harjoitteluprosessin toimintamalleissa on vielä aukkoja ja niiden ratkaisuja kehitetään ohjelmistohankkeen ulkopuolella.

7 Tulokset

Opinnäytetyöprojektin toiminnallisen osan tuloksista ja hyödyistä on keskusteltu projektin tuoteomistajan kanssa, jonka mukaan työ on hyödyttänyt sekä projektin ohjausryhmää, että kehittäjiä. Erityisesti vaatimusten kartoitus on ollut kriittisen tärkeää ohjelmistohankkeen edistymisen kannalta. Työn tuotoksena syntynyttä vaatimusmäärittelydokumenttia on käytetty ohjelmistokehityksessä erityisesti kevään 2021 aikana ja sitä tullaan hyödyntämään ohjelmistohankkeessa myös jatkossa.

Opinnäytetyöprojektin aikana valmistuneessa vaatimusmäärittelyssä on puutteita sekä lisäselvitystarpeita, jotka tulee ratkaista seuraavissa kehitysvaiheissa. Lisäselvitystä tarvitaan esimerkiksi esitettyjen vaatimusten ristiriitojen ratkaisemiseksi sekä viranomaisvaatimusten tarkempaa määrittelyä varten.

7.1 Vaatimusmäärittelydokumentin ristiriidat

Projektissa toteutetussa vaatimusten kartoituksessa huomattiin, että harjoittelun ohjausprosessissa on jonkin verran ohjaajakohtaisia eroja, kuten erilaisten työkalujen käyttö harjoitteluiden seurannassa. Ennakkoaineistosta ja Metropolian asiantuntijoiden haastatteluissa oli kuitenkin havaittavissa, että ohjaajakohtaisista käytännöistä ollaan valmiita luopumaan, koska yhtenäisen toimintamallin etuja pidetään merkittävämpinä prosessin kehitykselle. Koulutusalojen välisiä eroja harjoitteluprosessissa tulee voida olla jatkossakin, sillä opintosuunnitelmissa on suuria eroja harjoitteluiden suorittamisen suhteen. Myös opiskelijan mahdollisuus toteuttaa omaa henkilökohtaista opetussuunnitelmaa tulee säilyttää. Osa opiskelijoista voi esimerkiksi ansaita harjoittelusuorituksen aiemman työkokemuksen perusteella, osa suorittaa harjoittelua osa-aika- tai keikka- tai vapaaehtoistyössä ja osa täysaikaisessa työsuhteessa.

Selkeänä poikkeuksena muista koulutusaloista erottuu sosiaali- ja terveysalojen koulutusten työharjoittelumenettelyt. Niihin sisältyy muita koulutusaloja enemmän vaiheita ja riippuvuuksia sekä opiskelijan ohjaukseen ja arviointiin sitoutuneita osapuolia. Merkittävimpänä erona on se, että harjoittelua ohjaa työpaikalla nimetty harjoitteluohjaaja, joka osallistuu opiskelijan työnohjaukseen ja harjoittelusuorituksen arviointiin. Työharjoittelu on olennainen osa koulutusta ja ko. koulutusaloilla koulutukset johtavat esimerkiksi valvontaviranomaisen myöntämään terveydenhuollon ammattihenkilön laillistamiseen ja ammattinimikkeen käyttöoikeuteen (Valvira, 2021). Työharjoittelu on osa ammattipätevyyttä ja harjoittelutietojen oikeellisuuden varmistamiseen, tiedonsiirtoon ja -tallennukseen liittyy siksi enemmän ja tarkoin viranomaistasolla säänneltyjä vaatimuksia sosiaali- ja terveydenhoitoaloilla kuin muilla Metropolian tai Haaga-Helian tarjoamilla koulutusaloilla.

Oppilaitoskohtaisia prosessin eroja havaittiin ohjaus- ja seurantamenettelyiden lisäksi harjoittelun ohjauksen organisoinnissa.

Edellä mainituista prosessien eroista johtuen vaatimustaulukkoon on jätetty tiedostettuja ristiriitoja ja epäjohdonmukaisuuksia. Taulukkoon haluttiin jättää näkyville vaihtoehtoiset mallit, jotka edellyttävät laajempaa selvitystä ja ratkaisua esimerkiksi prosessin toimintojen tai ohjelmiston arkkitehtuurin tasolla. Myös sosiaali- ja terveydenhoidon koulutusalojen vaatimukset on esitetty taulukossa, huolimatta ristiriidoista muihin vaatimuksiin nähden. Metropolian tarjoamista koulutusohjelmista sosiaali- ja terveydenhoidonalojen koulutus on opiskelijamäärältään suurin koulutusala ja niiden harjoitteluprosessin huomioiminen ohjelmistokehityksessä on tärkeää, jotta ohjelmiston käytön laajentaminen koko Metropoliaan olisi mahdollista.

Esiin nousseet vaatimusten mahdolliset ristiriidat ja riippuvuudet on pyritty huomioimaan vaatimustaulukossa F-sarakkeeseen kirjatussa kommentissa (Taulukko 2). Havaitut ristiriitoja aiheuttavat ongelmat on esitetty projektin tuoteomistajalle myös erikseen. Taulukossa 3 on esimerkkejä havaituista ristiriidoista.

Taulukko 3. Esimerkkejä esiin nousseista ristiriidoista

Vaihtoehto 1	Vaihtoehto 2
Yksi harjoitteluprojekti voi sisältää vain yhden harjoittelupaikkailmoituksen - selkeys raportoinnissa ja harjoittelun arvioinnissa	**Harjoitteluprojektiin tulee voida lisätä useita harjoittelupaikkoja** - harjoitteluprojekti on selkeästi yksi harjoittelujakso, joka on mahdollista suorittaa keikka- tai osa-aikatyössä eri työnantajilla
Työnantajan pääsy järjestelmään ei ole tarpeellista - ei tarvita erillisiä luku- ja kirjoitusoikeuksia oikeuksia työnantaja käyttäjäryhmälle - työantajan osallistuminen arviointiin ei ole tarpeellista	**Työnantajalla tulee olla pääsy harjoittelujärjestelmään** - sote-aloilla työnantajan osallisuudesta harjoittelun ohjaukseen ja arviointiin ei voida valvontaviranomaisten määräysten vuoksi luopua
Harjoittelun päättymispäivä ei ole pakollinen tieto - harjoittelun suoritus epäsäännöllisessä työssä mahdollista - ei vaadi jatkuvaa päivitystä	**Harjoittelun päättymistieto oltava pakollinen, jotta siihen perustuen voidaan seurata harjoitteluiden edistymistä** - automaattiset muistutukset mahdollisia
Harjoittelupaikkailmoitus tehdään aina järjestelmään lomakkeella, jossa on harjoittelijan ja harjoittelupaikan tiedot - palvelee kaikille yhtenäistä prosessia	**Harjoittelupaikkailmoitus tehdään järjestelmän ulkopuolella. Esimerkiksi sote aloilla käytössä Jobiili järjestelmä harjoittelupaikan varaamiseen ja ilmoittamiseen oppilaitokselle.**
Raportointipohjan kysymykset kaikille opiskelijoille samat. - vertailukelpoinen tilastotieto mahdollista	**Raporttipohjan kysymykset tulee olla ala- ja/tai harjoittelujaksokohtaiset** - alakohtainen tilastotieto mahdollista

7.2 Vaatimusmäärittelydokumentin kehittämiskohteita

Opinnäytetyöprojektin puitteissa ei ole tarkasti selvitetty ei-toiminnallisia vaatimuksia, joita ovat esimerkiksi käytettävyyttä, suorituskykyä tai turvallisuutta koskevat vaatimukset (Sommerville, 2011, 87). Niiden toteutuminen ohjelmistossa on jätetty hiljaisen tiedon varaan, koska niitä ei ole tarkasti dokumentoitu. Esimerkiksi kehitettävän ohjelmiston integraatio opiskelija- ja opintorekisterijärjestelmään luo ohjelmistolle yhteyden opiskelijoiden henkilötietoihin. Lisäksi on todennäköistä, että järjestelmä sisältää esimerkiksi yrityskohtaista tietoa harjoittelupaikoista tai opiskelijoiden henkilötietoihin yhdistettyjä opiskelijoiden arvioita työnantajasta. Projektin ohjausryhmän kanssa käydyissä keskusteluissa on havaittu, että vaadittuihin toimintoihin liittyy tarve tietojen keräämiseen, yhdistämiseen, siirtoon ja tallennukseen, joihin liittyvät määräykset on selvitettävä tarkemmin tulevassa kehityksessä. Riskienhallinnan parantamiseksi erityisesti viranomaisvaatimukset tulisi myös dokumentoida tarkemmin tulevaisuudessa. Niillä, kuten muillakin ei-toiminnallisilla vaatimuksilla voi olla suuri vaikutus useisiin toiminnallisiin vaatimuksiin, esimerkiksi yksi tietoturvaan liittyvä rajoite voi rajoittaa laajasti muiden myös toiminnallisten vaatimusten toteutusta ja siksi ne tulisi voida huomioida ohjelmistosuunnittelussa.

Ohjelmistohankkeen edetessä saattaa tulla tarvetta kehittää vaatimusmäärittelyn dokumentaatiota osoittamaan selkeämmin myös vaatimusten välisiä yhteyksiä. Käyttäjätarinat on tarkoitettu itsenäisiksi ja vaatimustaulukon tueksi laaditut kaaviot esittävät yhteyden vain käyttötapauksen, sidosryhmän tai prosessin vaiheen kautta. Vaatimukset voivat sisältää yhteyksiä myös muilla tavoilla, yksi käyttäjätarina voi rajoittaa useita muita tarinoita ja yhteen tarinaan voi myös liittyä useita yhteyksiä, siksi vaatimusten tarkentuessa ja lisääntyessä jatkossa voi olla tarpeellista hallita näitä yhteyksiä.

7.3 Kehittämismallien soveltuvuus HHTrainee hankkeeseen

Harjoitteluprosessi voidaan katsoa kuuluvan korkeakoulujen rutiinitoimintoihin, koska prosessi toistuu lähes jokaisen korkeakoulututkintoon tähtäävän opiskelijan kohdalla vähintään kerran. Harjoittelun ohjaus- ja seurantatoimintoihin liittyy suoraviivaisuutta ja vakiintuneisuutta. Kehityksen aikana on myös todettu vaatimusten ennalta kartoittamisen hyödyttävän ohjelmistoprosessia erityisesti ohjelmiston ennakoivaa suunnittelua. Näistä syistä puhtaasti ketterän mallin mukainen ohjelmistoprosessi, jossa vaatimuksia ei tarkasti määritellä etukäteen, ei mahdollisesti olisi tuottanut arvoa HHTrainee ohjelmiston kehitykselle.

Toisaalta vaatimusten määrittely on osoittanut, että täysin lukkoon lyötyjä yhtenäisiä toimintamalleja ei harjoitteluprosessissa ole, näin ollen ohjelmistoprosessin aikaiset muutokset vaatimuksiin ovat todennäköisiä ja ohjelmiston kehityksessä ei siksi ole järkevää edetä myöskään täysin vesiputousmallin mukaisesti.

EXOD malli yhdistää ketterän ja vesiputousmallin hyödyt korkeakoulujen prosessien digitalisointiin sopivalla tavalla, mutta ohjailee samalla prosessikehitystä. HHTrainee hanke kuitenkin ulottuu vain ohjelmistoprosessiin ja rajaa harjoitteluprosessin kehittämisen sen ulkopuolelle, siksi HHTrainee hankkeen näkökulmasta EXOD mallin soveltuvuutta on vaikea arvioida ennen kuin sekä prosessin kehitys, että HHTrainee ohjelmistohanke ovat edenneet pidemmälle. Jotta tulevissa Haaga-Helian kehityshankkeissa olisi mahdollista hyödyntää EXOD mallia ja erityisesti sen vaiheittaista etenemistä olisi prosessin digitalisointi järkevää toteuttaa sekä toiminnan että ohjelmiston kehityksen osalta yhtenäisenä projektina.

Toisaalta vaatimusmäärittely kulkee kuitenkin luontevasti prosessin ja ohjelmistokehityksen välillä ja tuottaa tietoa molempien hyödyksi. EXOD malli vaikuttaa palvelevan ohjelmistoprosessin määrittelyvaihetta hyvin. Mallissa määrittelytoiminnot on esitetty karkeasti tulkittuna niin, että prosessi määritellään vesiputousmallin tapaisesti ennen seuraavaa vaihetta eli ohjelmistoprosessia, ja kokonaisuudessaan ohjelmistoprosessi, jossa myös ohjelmistovaatimukset kootaan vaatimusmäärittelyksi, tapahtuu jatkuvana toimintana läpi ohjelmiston elinkaaren. Toteutetun vaatimusmäärittelyn myötä HHrainee hankkeeseen on sovellettu aiempaa kehitystä laajemmin EXOD-mallin vaiheita ja erityisesti sen periaatetta pitää kehityksessä yllä kommunikaatiota prosessin asiantuntijoihin ja tuleviin käyttäjiin. Laajasti käyttäjiä kuunnellen tehty vaatimusmäärittely tuottaa tietoa myös prosessinkehityksen tueksi. Sen avulla on mahdollista huomata ne kohdat prosesseissa, joissa toimintoja on järkevä järjestellä uudelleen sekä ne, jotka tulee säilyttää sellaisinaan myös ohjelmiston tukemassa prosessissa.

8 Pohdintaa opinnäytetyöprojektista

Ennakkoaineiston, kuten vaatimusmäärittelykurssien tehtävien tulokset valmistuivat vasta syksyn 2020 loppupuolella, jolloin ohjelmistokehitys oli jo meneillään. Tämän vuoksi aineistoja ei ehditty hyödyntämään kehityksessä syksyn 2020 aikana. Uudet vaatimukset tulikin saattaa kehittäjäorganisaation käyttöön, mahdollisimman nopeasti. Vaatimusmäärittelyn toteutuksen suunnitteluun ei juuri jäänyt aikaa ja siksi esimerkiksi vaatimuslistaa jouduttiin käymään läpi useaan otteeseen aina uusien vaatimusten esiin tullessa ja vaatimukset esitettiin jo käytössä olevien mallien mukaan sen sijaan, että olisi tutkittu muita vaatimusten esittämiskeinoja ja työkaluja.

Projektin edetessä selkeni ymmärrys, että täysin ristiriidattomia vaatimuksia on mahdotonta määrittää aikaisessa ohjelmiston ja prosessin kehitysvaiheessa, koska tarkkaa prosessiakaan ei ole vielä olemassa. Lisäksi havaittiin, että ohjelmistolle esitetyt vaatimukset saattavat toteutuessaan määritellä myös harjoitteluprosessin toiminnot tiettyyn muottiin. Siksi kaikkia työjonolle päätyneitä vaatimuksia ei todennäköisesti voida ottaa mukaan kehitykseen sellaisenaan, vaan harjoitteluprosessi on määriteltävä ensin ja suorittaa ohjelmistoprosessin seuraavat kehitysiteraaatiot sen mukaan. Edellä mainituista syistä myös vaatimusten analysointiin kului suunniteltua enemmän aikaa ja tuoteomistajan vastuu vaatimusten analysoinnista jäi arvioitua suuremmaksi, esimerkiksi vaatimusten priorisoinnin osalta. Lopputuloksessa on kuitenkin onnistuttu huomioimaan ohjelmistoprosessin ja siihen kuuluvan vaatimusmäärittelyn teoreettiset hyvät käytännöt sekä HHTrainee ohjelmistohankkeen erityispiirteet.

Ammatillisen kehittymisen näkökulmasta projektin edetessä oli hyödyllistä huomata, että vaatimusmäärittelyssä todennäköisesti tarkimman käsityksen toiminnoista ja siihen liittyvistä haasteista saa määrittelijä, joka käy konkreettisia keskusteluja eri käyttäjäryhmien kanssa. Toisin sanoen yleisesti ohjelmistoprojektille on todennäköisesti hyödyllisempää, että vaatimusmäärittelyn tekijä on projektissa mukana ohjelmistoprosessin alusta loppuun. Tämän projektin yhdeksi henkilökohtaiseksi tavoitteeksi muodostuikin tiedon tarkka siirto projektin käyttöön, jotta tulevissa kehitysvaiheissa voidaan mahdollisimman hyvin hyödyntää kaikki kerätty tieto, jolloin myös toimeksianto voidaan todeta onnistuneesti toteutetuksi.

Projektin aikana on käyty paljon keskustelua hankkeen ohjausryhmän kanssa, jolla on ollut positiivinen vaikutus vaatimusmäärittelyn tarkkuuteen ja oikea-aikaisuuteen. Myös ketterien menetelmien periaatteet korostavat keskustelun tehoa ylitse muiden kommunikointimenetelmien, siksi työn menetelmiä voidaan pitää onnistuneina; Projektin dokumentaatio on kompaktia, vaatimustaulukko on laadituista dokumenteista ainoa, joka suunnitelman

mukaan vaatii ylläpitoa, muut projektin aikana tuotetut dokumentit on laadittu kertakäyttöisiksi ja juuri keskustelua tukeviksi. Ohjelmistohankkeen ohjausryhmän kanssa käytyihin keskusteluihin osallistuvat kaikki mukana olleet ja niitä voidaan pitää tämän projektin onnistumisen kannalta ensisijaisen tärkeinä.

Haaga-Helian ohjelmistotuotannon opiskelijana olin itse mukana HHTrainee-ohjelmistoprojektissa syyslukukaudella 2020 kehittäjän roolissa. Tuolloin selkeiden vaatimusten ja tavoitteiden puute tuntui turhauttavalta. Ohjelmiston kokonaiskuvan ja vaatimusten kirkastaminen kehittäjille tuntui ohjelmistonkehityksen onnistumisen kannalta tärkeältä, siksi myös päätös tämän projektin valinnasta omaksi opinnäytetyöksi oli helppo. Työn edetessä kuitenkin vahvistui käsitys siitä, että lopputulosta ja tarkkoja vaatimuksia ei voida vielä täysin määritellä. Samalla lisääntyi ymmärrys ohjelmistokehitystyöstä ja ketteryyden merkityksestä ohjelmistotuotannolle. Silloin kun tarkkoja vaatimuksia ei ole on edettävä saatavilla olevan tiedon mukaan ja tarvittaessa muutettava kehityksen suuntaa, ketterästi.

Lähteet

Agilemanifesto.org (s.a.). *Agile manifesto*. Luettavissa: https://agilemanifesto.org/ Luettu: 26.4.2021.

Ambler, Scott W. (s.a.). *Agile/Lean Documentation: Strategies for Agile Software Development*. Luettavissa: http://agilemodeling.com/essays/agileDocumentationBestPractices.htm#JustSimpleEnough. Luettu: 26.4.2021.

Haaga-Helia (2021). *Tietoa Haaga-Heliasta*. Luettavissa: https://www.haaga-helia.fi/fi/haaga-heliasta. Luettu: 23.4.2021.

Cohn, M. (2010) *User Stories Applied, For Agile Software Development*. Pearson Education Inc.

Haikala, I. and Mikkonen, T. (2011) *Ohjelmistotuotannon käytännöt*. Talentum Media Oy.

Ilmarinen, V. and Koskela, K. (2015) *Digitalisaatio Yritysjohdon käsikirja*. 2nd edn. Talentum Media Oy.

JUHTA (2009) 'JHS 173. ICT-palveluiden kehittäminen: vaatimusmäärittely', pp. 1–29. Luettavissa: http://docs.jhs-suositukset.fi/jhs-suositukset/JHS173/JHS173.pdf. Luettu: 22.4.2021

Lagstedt, A., Lindstedt, J. P. and Kauppinen, R. (2020) 'An outcome of expert-oriented digitalization of university processes', *Education and Information Technologies*, 25(6), pp. 5853–5871. doi: 10.1007/s10639-020-10252-x.

Laplante, P. A. (2014) *Requirements Engineering for Software and Systems*. 2nd edn. CRC Press.

Leffingwell, D. (2011) *Agile Software Requirements Lean Requirements Practices for Teams, Programs, and the Enterprise*. Boston: Pearson Education Inc.

Pohl, K. (2010) *Requirements Engineering, Fundamentals, principles and techniques*. Springer.

Sommerville, I. (2011) *Software Engineering*. Pearson Education Inc.

Sosiaali- ja terveysalan lupa- ja valvontavirasto Valvira (2015). *Ammattioikeudet*. Luettavissa: https://www.valvira.fi/terveydenhuolto/ammattioikeudet. Luettu: 26.4.2021

Vilppola, I. and Terho, K. (2008) *Tehokkuutta tuotannon tietojärjestelmiin -loppukäyttäjät mukaan määrittelyyn*. Helsinki: Teknologiateollisuus.

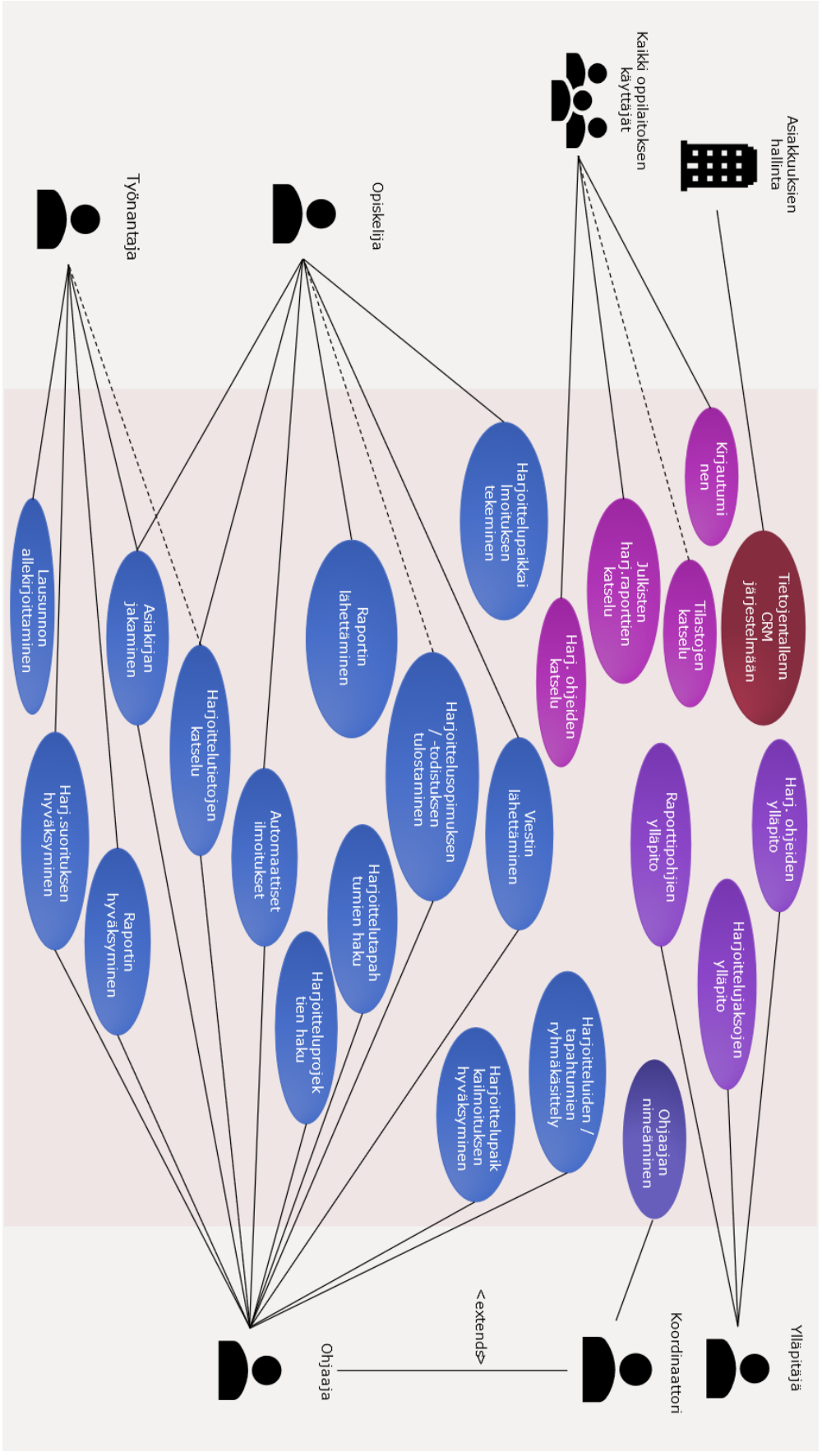
Valtioneuvoston asetus ammattikorkeakouluista 1129/2014

Liitteet

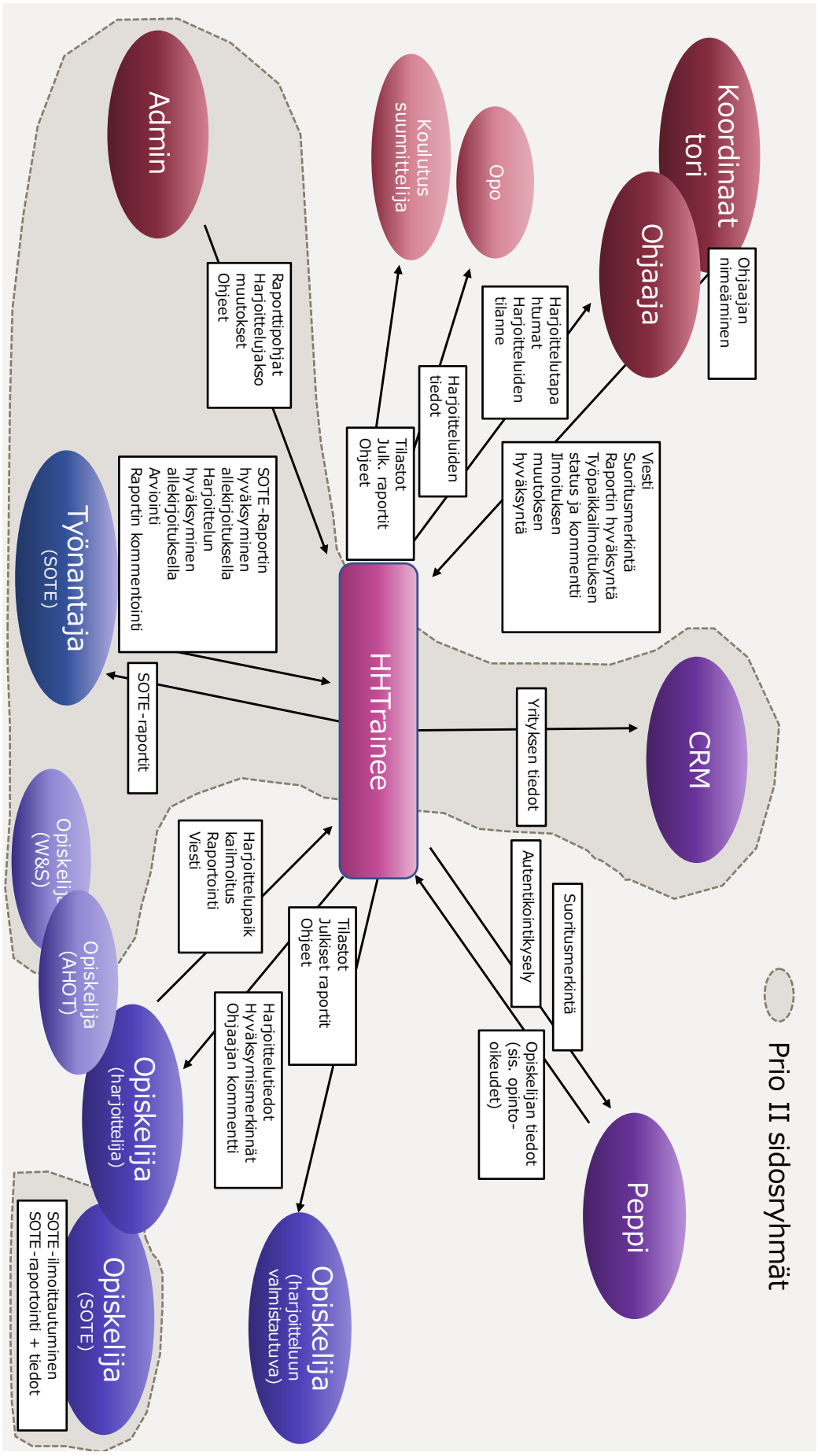
Liite 1. Otos vaatimustaulukosta

ID	Prio	Käyttäjäroo	Käyttäjätarina	Kuvaus	Kommentteja/huomioita	Liittyvät t	Hyväksymiskriteerit	Käyttötapaus ensisijainen	Lähde	Käyttötapaus muu
1038	2	Opiskelija	Haluan ladata asiakirjan järjestelmään		opiskelija voi ladata tiedostoja kuten tekstivan palautus	1041		Asiakirjan lähettäminen	Tuotelehtiysjo no ID 11	
1039	1	Opiskelija	Haluan lähettää raportin	Valmiit kaikille yhteiset raporttipohjat itsearviointi ym. jotka täytetään suoraan järjestelmään. Raporttipohjat (kysymykset) tuotenmistajilta. Raporttien lähettämisessä mahdollisesti riippuvuuksia jotka tarkentuvat. Esim. alohtusraportti pitäisi olla palautettu, jotta pääsee täyttämään loppuraportin tms.	Milloin raportti "avautuvat" opiskelijalle? Harkitaan onko tarvetta erilliselle deadline pvm (kuka määrittelee päivän), dl oltava, jos halutaan päivämäärään liittyviä automaattisia ilmoituksia. Metropolian keskustelussa ei opiskelijan vastuulla ja seurataan vain palautuksia. Opiskelijalle kannustimena toimii, suoritusmerkintä, jota ei voi saada ennen kuin vaadittavat palautukset on tehty.		Raportin lähettäminen	Tuotelehtiysjo no ID 11	Harjoittelutulejen kaseelu, Automaattisten ilmoitusten vastaanottaminen
1040	10	Ohjaaja	Haluan, että opiskelijan antaman raportin voi tarvittaessa piilottaa, jotta yrityksen salassa pidettävät tiedot ovat vain opiskelijan ja oman ohjaajan nähtävillä.	Ohjaaja voi ladata tiedostoja opiskelijan projektinäkymään.	Myös ohjaajalla mahdollisuus jakaa tiedostoja. Näin ohjaaja voisi jakaa esim. oman kyselyohjan arviteessa.	1038		Raportin lähettäminen	Metropolia tallenne, osa_1, 11.45	Asiakirjan lähettäminen, Harjoittelutulejen kaseelu
1041	100	Ohjaaja	Haluan ladata tiedostoja, jotka voin tarvittaessa antaa lisäohjeita ja tehtäviä.					Asiakirjan lähettäminen	Tuotelehtiysjo no ID 11	
		Ohjaaja	Haluan voida muokata harjoitteluraportin kysymyksiä, jotta eri koulutusohjelmien harjoitteluiden erityspiirteet voidaan huomioida tehtävissä.		Näkyvinen systeemi liian ajykä. Kaikille alolle / harjoittelulle ei sovi samat raporttipohjat. Toistaiseksi ohjaaja voi ladata tiedostoja kaikille pakollisten palauteraporttien lisäksi. Tulevaisuudessa pohdittaan tarvitsarko esim. valmiita vaihtoehtoisia	1065		Raportintiedostojen ylläpito	Metropolia tallenne, osa_3, 0.40	
1042	50	Ohjaaja	Haluan nykyistä kalevämpää rakatsua päiväkirjamaiseen raportin kirjoittamiseen, joka vaatisi vähemmän töitä työharjoittelulle ja ohjaajalle.		lisäraportiochia, joista ohjaaja voi tarvittaessa valita.					
1043	50	Ohjaaja	Haluan saada työnohjstuksen helposti työnantajalta, jotta saan tietää, onko oppilas suorittanut työharjoittelunsa.		Sähköpostella ja KYVYT järjestelmillä, jossa e ole integrointeia HHn järjestelmiin. KYVYT ei lähetä mitään ilmoituksia ohjaajalle, kun työharjoittelija on päivittänyt työkokemuksiaan alustalle. Taman takia joudutaan käymään usella kertoja tarkistamassa, onko tullut lisäksiä. (vaatimuksen esitäisän kommentti). Vaatii	1042		Raportin hyväksymmen	Kurssityö3	Harjoittelusuunnituksen hyväksymmen, Raportin lähettäminen
1044	50	Ohjaaja	Haluan saada työnohjstuksen helposti työnantajalta, jotta saan tietää, onko oppilas suorittanut työharjoittelunsa.		jos työnantajalle mahdollistetaan pääsy järjestelmään tiedoston voisi jakaa myös työnantaja. Vaihoehtona opiskelijan vastuulla lähettää työlodstus tiedostona.			Asiakirjan jakaminen	Kurssityö7	
1045	50	Ohjaaja	Haluan että voin sopivalla tavalla seurata osa-aikatyössä suoritettavaa harjoittelua, jotta pysyn ajantasalla myös vaihtoehtoisella tavalla suoritettavista harjoitteluista		Työuntten kirjaaminen, kun harjoittelu koostuu päkistä. Matkailualan harjoittelulle on tyypillistä, että miita tehdään päkissä, kuten lita- ja viikonloppuruorina. Toivotaan työkalua, jolla näiden kirjaaminen ja seuraaminen olisi helppoa.(vaatimuksen esitäisän kommentti). Tanilaanko erillinen	1038, 1042		Asiakirjan jakaminen	Kurssityö3	Raportintiedostojen ylläpito, Raportin lähettäminen
1046	3									
		Ohjaaja	Haluan voida muokata ohjausprosessiin liittyviä tehtäviä/seurantapisteitä, jotta		Harkitava liitetäänkö harjoittelupröjektiin seurantapisteitä palautusten määrärpäviä tms.	1042,1047, 1097		Raportintiedostojen ylläpito	Kurssityö3	Harjoittelutulejen kaseelu, Asiakirjan jakaminen

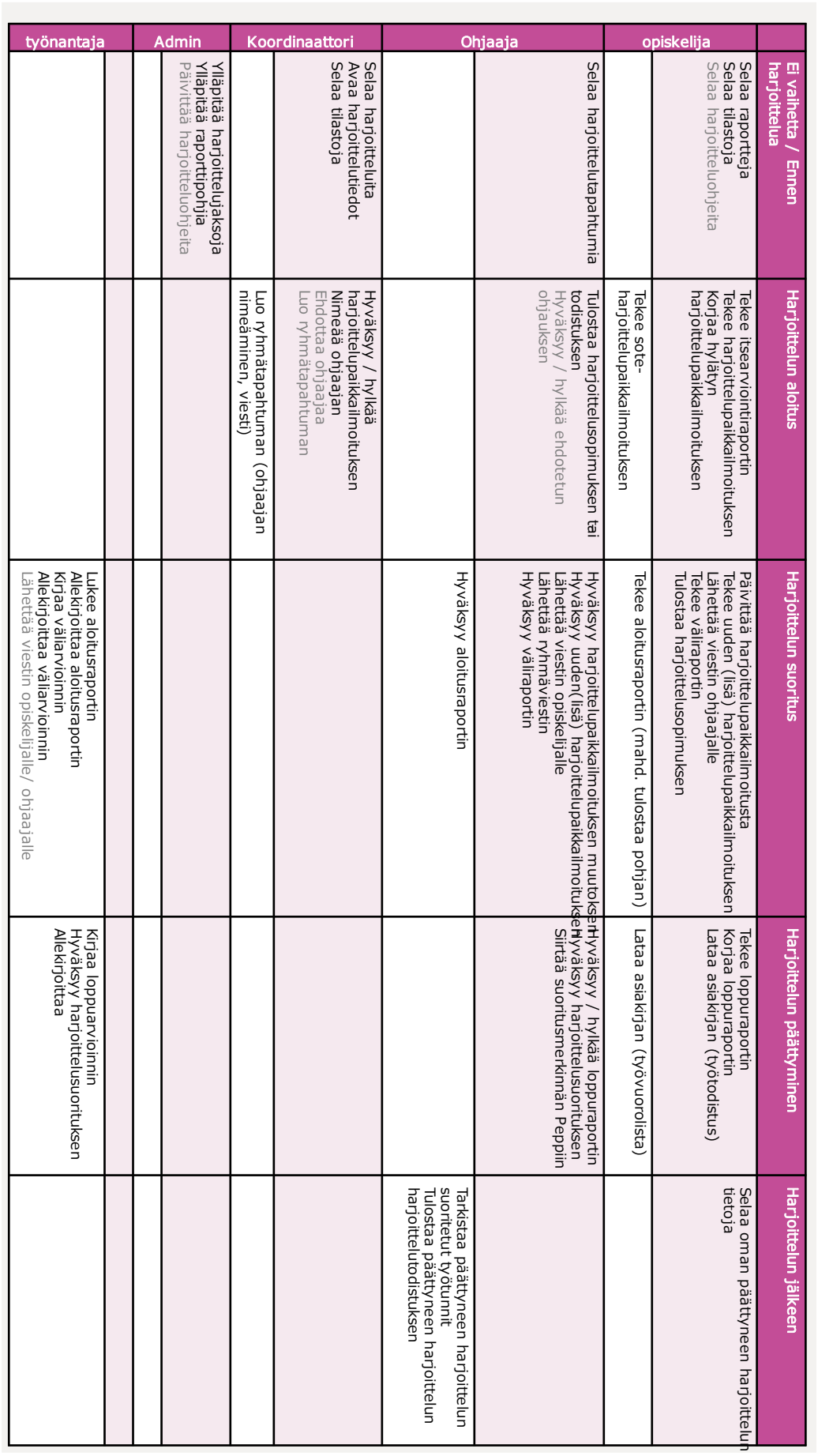
Liite 2. Käyttötapauskaavio



Liite 3. Sidosryhmäkaavio



Liite 4. Taulukko harjoitteluprosessin toiminnoista eri vaiheissa



	Ei vaihetta / Ennen harjoittelua	Harjoittelun aloitus	Harjoittelun suoritus	Harjoittelun päättyminen	Harjoittelun jälkeen
opiskelija	Selaa raportteja Selaa tilastoja Selaa harjoitteluohjeita	Tekee itsearviointiraportin Tekee harjoittelupaikkailmoituksen Korjaa hylätyn harjoittelupaikkailmoituksen	Päivittää harjoittelupaikkailmoitusta Tekee uuden (lisä) harjoittelupaikkailmoituksen Lähettää viestin ohjaajalle Tekee väliraportin Tulostaa harjoittelusopimuksen	Tekee loppuraportin Korjaa loppuraportin Lataa asiakirjan (työtodistus)	Selaa oman päättyneen harjoittelun tietoja
	Selaa harjoittelutapahtumia	Tekee sote-harjoittelupaikkailmoituksen	Tekee aloitusraportin (mahd. tulostaa pohjan)	Lataa asiakirjan (työvuorolista)	
Ohjaaja	Selaa harjoittelutapahtumia	Tulostaa harjoittelusopimuksen tai todistuksen Hyväksyy / hylkää ehdotetun ohjauksen	Hyväksyy harjoittelupaikkailmoituksen muutoksen Hyväksyy uuden(lisä) harjoittelupaikkailmoituksen Lähettää viestin opiskelijalle Lähettää ryhmäviestin Hyväksyy väliraportin	Hyväksyy / hylkää loppuraportin Hyväksyy harjoittelusuorituksen Siirtää suoritusmerkinnän Peppiin	
			Hyväksyy aloitusraportin		Tarkistaa päättyneen harjoittelun suoritetut työtunnit Tulostaa päättyneen harjoittelun harjoittelutodistuksen
Koordinaattori	Selaa harjoitteluita Avaa harjoittelutiedot Selaa tilastoja	Hyväksyy / hylkää harjoittelupaikkailmoituksen Nimeää ohjaajan Ehdottaa ohjaajaa Luo ryhmätapahtuman			
		Luo ryhmätapahtuman (ohjaajan nimeäminen, viesti)			
Admin	Ylläpitää harjoittelujaksoja Ylläpitää raporttipohjia Päivittää harjoitteluohjeita				
työnantaja			Lukee aloitusraportin Allekirjoittaa aloitusraportin Kirjaa väliarvioinnin Allekirjoittaa väliarvioinnin Lähettää viestin opiskelijalle/ ohjaajalle	Kirjaa loppuarvioinnin Hyväksyy harjoittelusuorituksen Allekirjoittaa	