

Tia Sinisalo

KÄYTTÄJÄVAATIMUSTEN KUVAAMINEN KÄYTTÄJÄTARINOINA

KÄYTTÄJÄVAATIMUSTEN KUVAAMINEN KÄYTTÄJÄTARINOINA

Tia Sinisalo
Opinnäytetyö
Syksy 2021
Tietojenkäsittelyn tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietojenkäsittelyn tutkinto-ohjelma

Tekijä: Tia Sinisalo
Opinnäytetyön nimi: Käyttäjävaatimusten kuvaaminen käyttäjätarinoina
Työn ohjaaja: Jouni Juntunen
Työn valmistumislukukausi ja -vuosi: Syksy 2021
Sivumäärä: 51 + 2 liitettä

Tämän opinnäytetyön tarkoituksena oli tutkia käyttäjätarinoita keinona kuvata ja dokumentoida käyttäjävaatimuksia. Opinnäytetyö sai alkunsa toimeksiantajan ohjelmistoprojektista, joka liittyy laajempaan tietojärjestelmän kehittämishankkeeseen. Toimeksiantajana oli markkinointitoimisto Dagmar Oy. Tavoitteena oli kartoittaa ohjelmiston tulevien käyttäjien tarpeita, ja sen pohjalta kuvata ja dokumentoida käyttäjävaatimukset käyttäjätarinoina.

Tietoperusta jakaantuu kahteen osaan, joista ensimmäisessä osassa aihealuetta tarkastellaan laajemmin avaamalla määrittelyn ja vaatimusten roolia ohjelmistotuotannossa. Konteksti käyttäjätarinoiden käyttöön luodaan tarkastelemalla ohjelmistokehitystä ja vaatimusmäärittelyä ketterän lähestymistavan näkökulmasta. Toisessa osassa syvennytään käyttäjätarinoihin vaatimusmäärittelyn työvälineenä, ja keskitytään erityisesti käyttäjätarinoiden kartoittamiseen, kirjoittamiseen ja dokumentointiin.

Toteutusosassa tutustutaan ohjelmiston tuleviin käyttäjiin, ja kartoitetaan tarpeita uudelle ohjelmistolle. Kartoitusmenetelmäksi valittiin tietoperustan pohjalta haastattelut, jotka toteutettiin yksilöhaastatteluina teemahaastatteluiden muodossa. Tarpeiden pohjalta tunnistetut käyttäjävaatimukset kuvattiin ja dokumentoitiin käyttäjätarinoina. Toteutusosa rajattiin koskemaan Dagmarin Media-yksikön digimainonnan suunnittelijoita.

Tuloksena todettiin, että käyttäjätarinoiden kirjalliset kuvaukset eivät ole tarkkoja käyttäjävaatimuksia, vaan ne ovat ennemminkin keino kuvata toiminnallisuuksia, joista tulee keskustella tarkemmin tulevien käyttäjien kanssa, jotta käyttäjävaatimus voidaan toteuttaa todellisen tarpeen ja yhteisen ymmärryksen pohjalta. Toimeksiantaja voi hyödyntää opinnäytetyötä sekä kirjallisia kuvauksia ohjelmistoprojektin valmisteluvaiheessa.

Asiasanat: käyttäjätarinat, käyttäjävaatimukset, vaatimusmäärittely, ketterä ohjelmistokehitys

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Business Information Systems

Author: Tia Sinisalo

Title of thesis: Representing user requirements as user stories

Supervisor: Jouni Juntunen

Term and year when the thesis was submitted: Autumn 2021

Number of pages: 51 + 2 appendices

The purpose of this thesis was to study user stories as a way of representing and documenting user requirements. The starting point for the thesis was the software project which is a part of wider information system development. The thesis was commissioned by marketing agency Dagmar Oy. The aim of the thesis was to elicit the needs of the software end users, represent and document user requirements as user stories.

The theoretical background of the thesis consists of two parts. The first part introduces the role of defining requirements in software engineering and discusses requirements in an agile framework. The concept of agile and basic principles of agile software development are also described. The second part concentrates on utilizing user stories in requirements definition in the context of eliciting, writing, and documenting.

The empirical part of the thesis focuses on the end users, and the purpose was to elicit the needs for the software. The research method was chosen based on the theoretical part. Individual interviews were conducted as theme-based interviews. After elicitation, the user requirements were represented and documented as user stories. The empirical part was limited to Dagmar's Media unit and digital media planners.

As a result of the study, it can be stated that the descriptions of user stories were not detailed user requirements but rather a way describe functionalities. These functionalities should be conversed more specifically with the end users in order to implement user requirements based on the real needs and shared understanding. The commissioner of the thesis can utilize this study as well as the descriptions to assist in the discovery phase of the upcoming software project.

Keywords: user stories, user requirements, requirements definition, agile software development

SISÄLLYS

1	JOHDANTO	6
1.1	Toimeksiantaja	7
1.2	Toimeksiannon lähtökohdat.....	7
1.3	Tavoitteet ja tarkoitus	8
1.4	Kehitystehtävät ja rajaus	9
2	MÄÄRITTELYN JA VAATIMUSTEN ROOLI OHJELMISTOTUOTANNOSSA	10
2.1	Ohjelmistotuotanto	10
2.2	Ohjelmistoprojekti.....	11
2.3	Ketterä lähestymistapa	14
2.4	Vaatimukset ja vaatimusmäärittely	16
3	KÄYTTÄJÄTARINAT VAATIMUSMÄÄRITTELYN TYÖVÄLINEENÄ.....	20
3.1	Käyttäjätarinat	20
3.2	Kartoittaminen	22
3.3	Kirjoittaminen.....	26
3.4	Dokumentointi	29
4	KÄYTTÄJÄVAATIMUSTEN KARTOITTAMINEN JA DOKUMENTOINTI.....	30
4.1	Käyttjävaatimusten kartoittaminen.....	30
4.1.1	Digimainonta	30
4.1.2	Digisuunnittelijat.....	31
4.1.3	Kartoitusmenetelmät	32
4.2	Käyttjävaatimusten dokumentointi.....	34
4.2.1	Suunnittelu- ja ostoprosessi	34
4.2.2	Käyttäjätarinoiden kuvaukset	37
4.2.3	Tunnistetut kehityskohteet	38
5	POHDINTA.....	41
	LÄHTEET.....	45
	LIITTEET	52

1 JOHDANTO

Teknologinen kehitys on tuonut ohjelmistot merkittäväksi osaksi jokapäiväistä elämäämme, ja yritysten näkökulmasta ne ovat siirtyneet tukiroolista prosessien keskiöön (Vainikainen 2020). Ohjelmistojen avulla pyritään muun muassa tehostamaan prosesseja ja ratkaisemaan erilaisia liiketoimintaan liittyviä haasteita. Kun ohjelmiston avulla haetaan kilpailuetua, päädytään lähes poikkeuksetta räätälöityyn ratkaisuun, jonka yritykselle tuottama arvo voi syntyä yhden tai useamman osa-alueen yhdistelmänä, kuten esimerkiksi automatisoinnista, prosessien tehostamisesta tai paremmasta käyttäjäkokemuksesta. (Gyursanszky 2020; Sofokus 2021.)

Räätälöidyn ohjelmiston kehitykseen sopii parhaiten ketterä lähestymistapa, joka joustavuutensa ansiosta tarjoaa mahdollisuuden toteuttaa muutoksia nopealla tahdilla jatkuvasti muuttuvassa liiketoimintaympäristössä (Tirkkonen 2019; Eficode 2021). Toimiminen teknologisen kehityksen ja muiden muuttuvien tekijöiden muovaamassa liiketoimintaympäristössä edellyttää yritykseltä sopeutumista ja toimintamallien muutosta. Monimutkaisten ohjelmistojen ja kokonaisten tietojärjestelmien kehittäminen voidaankin nähdä osana yrityksen toiminnan kehittämistä, jonka tarkoituksena on saada aikaan toimintatavan muutos. (Pohjonen 2002, 14.)

Muutokseen tulee sitouttaa alusta asti myös ohjelmiston tulevat käyttäjät, näin voidaan edesauttaa muutosten jalkautusta jo ennen uuden ohjelmiston käyttöönottoa. Käyttäjiltä löytyy lisäksi tärkeää substanssiosaamista, jota vaaditaan tunnistamaan tarpeita, joita erinäisten tehtävien suorittamiseen liittyy. (Härkönen 2014.) Tarpeiden pohjalta voidaan selvittää kehityksen kannalta oleelliset käyttäjävaatimukset, jotka kuvaavat tavoitteita, joita käyttäjien on kyettävä ohjelmistolla suorittamaan (Wiegiers & Beatty 2013, luku 1). Käyttäjävaatimuksia voidaan ketterän lähestymistavan mukaisesti kuvata esimerkiksi käyttäjätarinoina (Cohn 2009, luku 1).

Tässä opinnäytetyössä perehdytään ohjelmiston kehitykseen ketterän lähestymistavan näkökulmasta, ja tutkitaan käyttäjätarinoita keinona kuvata ja dokumentoida käyttäjävaatimuksia. Opinnäytetyö sai alkunsa toimeksiantajan ohjelmistoprojektista, joka liittyy laajempaan tietojärjestelmän kehittämishankkeeseen. Toimeksiannon taustalla on tarve lisätä ymmärrystä ohjelmiston tulevien käyttäjien tarpeista ja arjen haasteista sekä siitä, kuinka käyttäjävaatimuksia voidaan kuvata ja dokumentoida osana ketterää ohjelmistoprojektia. Tavoitteena on, että toimeksiantaja voisi hyödyntää opinnäytetyötä ohjelmistoprojektin valmisteluvaiheessa ennen kehitystyön aloittamista.

1.1 Toimeksiantaja

Opinnäytetyön toimeksiantaja on Dagmar Oy, joka on markkinointialalla toimiva suomalainen yritys. Dagmar tarjoaa asiantuntijapalveluita asiakkaidensa liiketoiminnan kehittämiseen markkinoinnin eri osa-alueilla. Palveluvalikoima kattaa keinot muun muassa asiakaskokemuksen kehittämiseen, markkinoinnin vaikuttavuuden ja myynnin tehojen kasvattamiseen sekä markkinoinnin automatisointiin ja uudistamiseen. Suomen suurin markkinointitoimisto Dagmar on perustettu vuonna 1973 ja se on osa Salomaa Yhtiöt -ryhmää, joka on Suomen suurin (Kauppalehti 2021) markkinointiviestintäalan konserni. (Dagmar 2021a.)

Yrityksessä työskentelee noin 200 markkinoinnin asiantuntijaa, jotka muodostavat Dagmarin Drive, Stories ja Media -yksiköt. Drive -yksikkö konsultoi markkinoinnin johtamista analytiikan, tutkimusten ja teknologian voimin. Stories -yksikkö on erikoistunut luoviin ja vaikuttaviin sisältöihin. Media -yksikkö suunnittelee tehokkaita kanavastrategioita ja kohtaamisia kaikkiin asiakaspolun vaiheisiin. Dagmar omistaa lisäksi RADLY Oy:n, joka on erikoistunut digitaalisen liiketoiminnan kehittämiseen tekoälyn avulla. (Dagmar 2021a; Dagmar 2021b; Dagmar 2021c; Dagmar 2021d.)

1.2 Toimeksiannon lähtökohdat

Toimeksiannon lähtökohtana on Dagmarin ohjelmistoprojekti, jonka tarkoituksena on uudistaa yrityksen tärkein työkalu, markkinoinnin suunnittelu- ja laskutusohjelmisto. Nykyinen MediaPro -ohjelmisto on Dagmarin oma ohjelmisto, joka on räätälöity yrityksen tarpeisiin. MediaPro on suunniteltu erityisesti mediasuunnittelun ja -ostamisen työkaluksi sekä taloushallinnon käyttöön, mutta se on tärkeä työkalu koko yrityksen näkökulmasta. Kyseessä on toiminnanohjausjärjestelmä, joka on syvällä koko liiketoiminnan ytimessä. (Holopainen, Kallio & Kämäräinen 2020.)

MediaPron avulla tallennetaan ja ylläpidetään muun muassa asiakastietoja, laskutussopimuksia ja asiakkaiden mediasuunnitelmia sekä hoidetaan kaikki tulevien ja lähtevien laskujen käsittelyyn liittyvät asiat. Lisäksi sen avulla ajetaan paljon raportteja ja tilastoja. MediaPro on käyttäjien koneille paikallisesti asennettava ohjelmisto, joka muodostuu käyttäjäryhmittäin kehitetyistä moduuleista. Moduulien välillä on paljon yhteistä, mutta niiden välillä on myös selkeitä eroja ja näkymiä, jotka on luotu vastaamaan kunkin käyttäjäryhmän tarpeita. Käyttäjäryhmiä ovat muun muassa taloushallinto

sekä Media -yksikön mediaryhmien suunnittelijat. (Kuoppa 2021.) Mediaryhmien suunnittelijoilla tarkoitetaan printti-, ulko-, elokuva-, televisio-, radio- ja digimainonnan suunnittelijoita.

Uudistus liittyy laajempaan tietojärjestelmän kehittämishankkeeseen, johon kuuluu myös muita yrityksen sisäisiä kehitysprojekteja. MediaPron osalta kehityskohteet liittyvät muun muassa käytettävyyteen sekä siihen, että ohjelmiston tekniset ja arkkitehtuuriset ratkaisut eivät vastaa enää tämän päivän haasteisiin, jonka takia ohjelmiston ylläpito ja jatkokehitys on koettu haastavaksi. Yrityksessä on tunnistettu myös suunnittelu- ja ostoprosesseihin sekä automaatioon liittyviä kehityskohteita, joihin nykyinen ohjelmisto ei tarjoa ratkaisua. Suunnittelu- ja ostoprosessien osalta kehityskohteet liittyvät muun muassa työn sujuvoittamiseen. (Holopainen ym. 2020.)

Ohjelmistoprojektia on valmisteltu yrityksessä jo pidemmän aikaa ja vaihtoehtoja erilaisten valmiiden ratkaisujen osalta on kartoitettu. Sopivaa valmisohjelmistoa ei ole löytynyt, joten on päädytty ratkaisuun kehittää Dagmarin tarpeisiin uusi, räätälöity markkinoinnin suunnittelu- ja laskutusohjelmisto. Uuden ohjelmiston kehitys tapahtuu yhdessä ulkoisen yhteistyökumppanin kanssa, joka valitaan kilpailutuksen perusteella. Kilpailutukseen on Dagmarin tueksi palkattu ulkoinen asiantuntija, ja Dagmarin sisällä projektille on nimetty projektista vastaavat henkilöt. (Holopainen ym. 2020.)

Ohjelmistoprojekti on kokonaisuudessaan laaja ja sen vaikutukset yltävät kaikkiin Dagmarin eri yksiköihin ja toimintoihin. Uuden ohjelmiston kehityksessä on huomioitava eri yksiköiden väliset erot sekä muun muassa se, että Media -yksikön sisällä eri mediaryhmien suunnittelijoilla on omat tarpeensa, jotka johtuvat suunnittelu- ja ostoprosessien välisistä eroista. Uuden ohjelmiston kehityksessä on otettava huomioon myös kaikkien muiden sisäisten ja ulkoisten sidosryhmien tarpeet. Lisäksi on huomioitava muun muassa riippuvuudet muista ohjelmistoista. (Holopainen ym. 2020.)

1.3 Tavoitteet ja tarkoitus

Ohjelmistoprojektin tavoitteena on modernisoida nykyinen markkinoinnin suunnittelu- ja laskutusohjelmisto. Tarkoituksena on kehittää nykyisen MediaPro -ohjelmiston tilalle korvaava ohjelmisto, joka ratkaisee tunnistetut kehityskohteet ja palvelee yrityksen tarpeita nyt ja tulevaisuudessa. Uuden ohjelmiston tavoitteena on parantaa käytettävyyttä sekä sujuvoittaa päivittäistä työtä poistamalla päällekkäisiä työvaiheita. Tavoitteena on myös kasvattaa työn automaatioastetta ja vapauttaa aikaa tuottavamman työn tekemiseen. (Holopainen ym. 2020.)

Opinnäytetyön tavoitteena on toimia toimeksiantajan apuna ohjelmistoprojektin valmisteluvaiheessa. Tarkoituksena on tehdä pohjatyötä, koska ohjelmiston kehittäminen vaatii ymmärrystä kehitykseen liittyvistä käytännöistä sekä perehtymistä muun muassa ohjelmiston tulevien käyttäjien tarpeisiin, joiden huomioiminen on keskeinen osa ketterää lähestymistapaa ja se on lähtökohta myös käyttäjäkeskeiselle suunnittelulle. Tavoitteena on lisätä ymmärrystä ohjelmiston tulevien käyttäjien tarpeista ja arjen haasteista sekä siitä, kuinka käyttäjävaatimuksia voidaan kuvata ja dokumentoida osana ketterää ohjelmistoprojektia. Tarkoituksena on tutkia käyttäjätarinoita keinona kuvata ja dokumentoida käyttäjävaatimuksia.

1.4 Kehitystehtävät ja rajaus

Opinnäytetyön tietoperustassa tullaan avaamaan aihealueen kannalta keskeisimpiä käsitteitä, joiden pohjalta pyritään muodostamaan yleiskuva ketterästä ohjelmistoprojektista ja luomaan konteksti käyttäjätarinoiden käyttöön. Käyttäjätarinoiden osalta tietoperustassa tullaan keskittymään erityisesti kartoittamiseen, kirjoittamiseen ja dokumentointiin. Tarkoituksena on selvittää mitä käyttäjätarinat ovat, ja kuinka niiden avulla voidaan kuvata ja dokumentoida käyttäjävaatimuksia osana ketterää ohjelmistoprojektia. Tietoperusta perustuu alan kirjallisuuteen ja eri verkkolähteisiin, kuten esimerkiksi avoimen yliopiston oppimateriaaliin sekä artikkeleihin ja blogikirjoituksiin.

Opinnäytetyön toteutusosassa tullaan perehtymään ohjelmiston tuleviin käyttäjiin ja pyrkimyksenä on selvittää tarpeet uudelle markkinoinnin suunnittelu- ja laskutusohjelmistolle. Lisäksi pyritään ymmärtämään nykyisen ohjelmiston käyttöön ja käytettävyyteen liittyviä arjen haasteita. Tarkoituksena on yhdistää tietoperusta käytännön toteutukseen kuvaamalla ja dokumentoimalla kartoitetut käyttäjävaatimukset käyttäjätarinoina. Käyttäjävaatimusten kartoitukseen käytettävät menetelmät tarkentuvat tietoperustan pohjalta. Kaikkien eri käyttäjäryhmien huomioiminen on tämän opinnäytetyön kontekstissa mahdotonta, joten toteutusosa rajataan koskemaan Dagmarin Media -yksikön digimainonnan suunnittelijoita.

2 MÄÄRITTELYN JA VAATIMUSTEN ROOLI OHJELMISTOTUOTANNOSSA

2.1 Ohjelmistotuotanto

Ohjelmistotuotanto (engl. software engineering) on käsitteenä laaja ja lähteestä riippuen se on määritelty eri tavoin. Kansainvälinen tekniikan alan järjestö IEEE (Institute of Electrical and Electronics Engineers) määrittelee sen tarkoittavan systemaattista, kurinalaista, mitattavissa olevaa tapaa ohjelmistojen kehittämiseen, operointiin ja ylläpitoon (IEEE 610.12 1990). Ohjelmistolla (engl. software) tarkoitetaan tietokoneohjelmaa sekä siihen liittyvää dokumentaatiota. Ohjelmistotuotannon tarkoituksena voidaan sanoa olevan tietokoneohjelman tuottaminen tavalla, jolla kehityskustannukset ja aikataulu ovat ennustettavissa riittävällä tarkkuudella. (Haikala & Mikkonen 2011, 11–12.)

Ohjelmistotuotanto pitää sisällään yleisesti käytettyjä tekniikoita, työkaluja, menettelytapoja ja periaatteita, ja siihen voidaan katsoa kuuluvaksi lähes kaikki ohjelmistotyön vaiheet sekä niihin liittyvät tuotantoprosessin osa-alueet. Ohjelmistotyön vaiheita ovat muun muassa määrittely, suunnittelu, toteutus, testaus ja laadunvarmistus. Tuotantoprosessin osa-alueita ovat muun muassa laatujärjestelmä, tuotteenhallinta, projektinhallinta ja dokumentointi. Ohjelmistotuotanto koostuu joukosta hyväksi havaittuja toimintatapoja ja periaatteita, joita sovelletaan insinöörimaisella tavalla ohjelmistokehitykseen. (Haikala & Mikkonen 2011, 11–12; Luukkainen & Ilves 2021b.)

Ohjelmistokehitys (engl. software development) on käsite, joka pitää sisällään ohjelmiston tuottamisen, mutta myös kysymyksenasettelun muun muassa sen suhteen, että miksi ohjelmistolle on ylipäättään tarve (Luukkainen & Ilves 2021b). Kysymys on oleellinen, koska ohjelmiston ensisijainen tehtävä on ratkaista jokin ongelma ja sen myötä tuottaa arvoa, jonka tuoma hyöty on suhteessa kehityskustannuksiin. Pelkkä ohjelmiston tuottaminen ei siis näin ollen yksinään riitä tuomaan haluttua lopputulosta, joten taustalla olevan ongelman selvittäminen on oleellinen osa ohjelmistokehitystä. (Robertson & Robertson 2013, luku 1.)

Määrittelyn tehtävänä on selvittää ongelma ja määritellä se, mitä ollaan kehittämässä. Tarkoituksena on ymmärtää kehityksen taustalla olevia todellisia tarpeita, jonka pohjalta määritellään ohjelmiston vaatimukset (engl. requirements) eli mitä ohjelmiston tulee tehdä ja kuinka sen tulee toimia.

Yksi alan tunnetuimmista asiantuntijoista, Frederick ”Fred” Brooks (1987) totesi aikoinaan paljon huomiota saaneessa ohjelmistotuotannon monitahoisuutta käsittelevässä No silver bullet -artikkelissaan, että yksi haastavimmista asioista on nimenomaan päättää se, mitä tarkalleen ottaen ollaan kehittämässä. (Haikala & Mikkonen 2011, 61; Robertson & Robertson 2013, luku 1.) Saman ovat todenneet monet muutkin asiantuntijat, ja esimerkiksi Mastering the requirements process -kirjassaan Suzanne ja James Robertson (2013, luku 7) painottavat oikean ongelman tunnistamisen tärkeyttä ja korostavat sitä alla olevalla sitaatilla.

“If I had an hour to save the world, I would spend 59 minutes defining the problem.”

—*Albert Einstein*

Albert Einsteinin lausahdus saattaa kuulostaa äärimmäiseltä vertauskuvalta, mutta sen tarkoituksena on korostaa huolellisen määrittelyn tärkeyttä ja oikean ongelman tunnistamista, joka edesauttaa oikeiden vaatimusten löytymistä (Robertson & Robertson 2013, luku 7). Vaatimusten perusteella suunnitellaan miten ohjelmisto tullaan toteuttamaan eli luodaan pohja ohjelmiston tekniselle toteutukselle. Vaatimusten näkökulmasta ohjelmistokehityksen voidaan yksinkertaistettuna ajatella olevan kuvaus korkealta abstraktiotasolta täsmälliseksi ja yksityiskohtaiseksi kuvaukseksi, ohjelmointikielellä kuvatuksi tietokoneohjelmaksi. (Haikala & Mikkonen 2011, 61–62.) Suunnittelun ja toteutuksen lisäksi vaatimukset luovat pohjan myös muun muassa ohjelmiston testaukselle (Young 2004, 2).

Vaatimusten merkitystä korostaa myös se, että suurin osa kehityksen aikana esiin tulleista virheistä johtuu nimenomaan vaatimuksista ja niihin liittyvistä väärinymmärryksistä (Robertson & Robertson 2013, luku 1). Epäonnistumisen syyt juontavat juurensa yli 60 prosenttisesti huonosti hoidettuun vaatimusten käsittelyyn (engl. requirements engineering), joka toimii yhdistävänä tekijänä muiden ohjelmistotuotantoon liittyvien aktiviteettien välillä (Haikala & Mikkonen 2011, 61). Haikalan ja Mikkosen (2011, 61) mukaan hyvin hoidettu vaatimusten käsittely onkin yksi onnistuneen ohjelmistoprojektin perusedellytyksiä.

2.2 Ohjelmistoprojekti

Ohjelmiston kehittäminen on monimutkainen prosessi ja se organisoidaan tyypillisesti projektiksi tai osaprojekteiksi, jotka liittyvät johonkin laajempaan kokonaisuuteen – hankkeeseen. Projekti

määritellään kertaluonteiseksi ja sen voidaan nähdä koostuvan erilaisista tehtävistä ja vaiheista, joilla on alku- ja loppuajankohdat. Kehitys perustuu tyypillisesti johonkin ohjelmistoalustaan, yleiseen sovelluskehikseen tai olemassa olevaan aiempaan ohjelmistoon, ja projektiin liittyy usein ainakin määrittelyä, suunnittelua, toteutusta ja testausta sekä käyttöönottoon ja ylläpitoon liittyviä asioita. Lisäksi projektiin kuuluu joukko jatkuvia tukitoimia, joita ovat muun muassa projektinhallinta, vaatimustenhallinta, muutostenhallinta, tuotteenhallinta ja dokumentointi. (Haikala & Mikkonen 2011, 19–23, 30–31.)

Kaikkia tehtäviä, vaiheita ja tukitoimia, jotka liittyvät ohjelmiston kehitykseen kutsutaan tässä opinäytetyössä ohjelmistoprojektiksi. Projektin osapuolia ovat asiakas eli ohjelmiston hankkiva taho, ja toimittaja eli ohjelmiston tuottava taho. Toimittajalla tarkoitetaan tässä yhteydessä yritystä, jonka työntekijöitä ovat muun muassa ohjelmistokehittäjät. Asiakkaalla puolestaan tarkoitetaan tässä yhteydessä yritystä, jonka työntekijät tulevat olemaan ohjelmiston loppukäyttäjiä. Ohjelmistoprojektin lähtökohta on asiakkaan liiketoiminnalliset tavoitteet ja projekti itsessään on keino tavoitteiden saattamiseen. (Haikala & Mikkonen 2011, 19–21, 30–31.)

Ohjelmistoprojekti lähtee liikkeelle valmistelusta, jonka voidaan sanoa olevan projektin onnistumisen kannalta ensiarvoisen tärkeä. Suurin osa projektiin liittyvistä ongelmista aiheutuu nimenomaan juuri valmisteluvaiheen puutteista ja laiminlyönneistä. (Juvonen 2018, 22.) Valmisteluvaiheessa tehdään nimensä mukaisesti valmisteluja, ja kuvataan muun muassa alustavasti projektin sisältö ja liiketoiminnalliset perusteet sekä analysoidaan projektin riskit. Liiketoiminnalliset perusteet tulisi laatia huolella, koska ne oikeuttavat projektin olemassaolon – mitä vahvemmat perusteet, sen parempi. (Haikala & Mikkonen 2011, 35; Juvonen 2018, 28–29.)

Liiketoiminnallisten perusteiden avulla pyritään osoittamaan mahdollisimman realistinen arvio projektin tuottamasta taloudellisesta hyödystä. Taloudellisen hyödyn osoittaminen saattaa tosin olla hankalaa, mikäli ohjelmisto tuottaa hyötyä vasta pitkällä aikavälillä. Jos kyseessä on vanhan ohjelmiston korvaaminen uudella, niin hyötynä voi olla ohjelmiston ylläpidon määrän väheneminen tulevien vuosien aikana. Hyötynä voi myös olla uuden ohjelmiston parempi käytettävyys, joka mahdollistaa sen, että ohjelmiston käyttäjiltä alkaa säästyä aikaa tuottavammalle työlle. Uusi ohjelmisto tuottaa siis näin ollen pitkällä aikavälillä hyötyä, mutta lyhyellä aikavälillä kulut kasvavat ohjelmiston kehityksen ja käyttöönoton vuoksi. (Juvonen 2018, 28–29.)

Ohjelmistoprojektin tavoitteena voi olla esimerkiksi nykyisen työn tehostaminen tai uusien kyvyksien mahdollistaminen (Robertson & Robertson 2013, luku 5). Määrittelyn avulla pyritään tässä tapauksessa selvittämään muun muassa se, että mitä uuden ohjelmiston myötä toivotaan tapahtuvan helpommin tai nopeammin, ja kuinka se tulee vaikuttamaan asiakkaan liiketoimintaan (Alfame 2021). Oleellista on tässä yhteydessä kuvata liiketoiminnan kannalta kriittiset prosessit sekä niiden kehittämistarve (ProjectTOP 2021). Pelkkä nykytilan optimointi ei kuitenkaan välttämättä ole järkevää, joten kehittämistarpeita tulisikin tarkastella jatkuvassa muutoksessa olevan liiketoimintaympäristön ja prosessien uudistamisen näkökulmasta (Vainikainen 2020).

Tavoitteiden saavuttaminen edellyttää ohjelmistoprojektin osapuolten välistä toimivaa yhteistyötä, jonka ytimessä on luottamukseen perustuva vuorovaikutus (Ruuska 2020). Tavoitteiden saavuttamiseksi ohjelmistoprojektille on myös annettava käyttöön joukko resursseja, kuten esimerkiksi taloudellisia tai henkilöresursseja, joiden avulla projektin toteuttaminen ja läpivieminen on ylipäättään mahdollista. Läpiviemiseen on olemassa lukuisia erilaisia lähestymistapoja ja projektimalleja, jotka poikkeavat toisistaan ainakin siinä, kuinka ohjelmistotyön vaiheita sovelletaan missäkin projektin vaiheessa. (Haikala & Mikkonen 2011, 21–23, 29.) Malleja ja lähestymistapoja voidaan myös soveltaa keskenään, jolloin voidaan käyttää sanaa hybridi (Garstin 2020).

Linearisessa mallissa eli vesiputousmallissa ohjelmistotyön vaiheet toteutetaan lineaarisesti peräkkäin niin, että vaiheet seuraavat toisiaan. Seuraavaan vaiheeseen siirrytään vasta kun edellinen vaihe on valmis. Koko ohjelmisto on yleensä etukäteen tarkkaan määritelty ja suunniteltu ennen kuin se toteutetaan. Testaus puolestaan suoritetaan vasta kun ohjelmisto on valmiina, jolloin virheiden korjaaminen saattaa osoittautua haastavaksi ja voi myös tulla kalliiksi. Vesiputousmallin keskeisin ongelma on kuitenkin siinä, että projektin edetessä asiakkaan tarpeisiin tulee todellisuudessa väistämättä lisäyksiä, tarkennuksia ja muutoksia, jolloin edellisiin työvaiheisiin on pakko palata. (Haikala & Mikkonen 2011, 21–23, 36–37; Luukkainen & Ilves 2021b.)

Iteratiiviset mallit mahdollistavat sen, että aikaisempiin työvaiheisiin voidaan palata uudestaan. Kehitys toteutetaan toistuvina vaiheina eli iteraatioina, joiden aikana ohjelmistotyön vaiheet toteutetaan uudelleen. Iteraatioiden voidaan nähdä muodostavan peräkkäisiä vesiputouksia, joiden aikana ohjelmistotyön vaiheita suoritetaan hieman eri painotuksin. Kattavaa määrittelyä ja suunnittelua ei pyritä tekemään heti projektin alussa, mutta alkuvaiheessa niillä on suurempi painoarvo. Oh-

ohjelmiston toteutus ja testaus alkavat kuitenkin jo ensimmäisten iteraatioiden aikana, ja myöhemässä vaiheessa puolestaan niillä on suurempi painoarvo. (Haikala & Mikkonen 2011, 40–42, Luukkainen & Ilves 2021b.)

Vesiputousmalli sekä monet 90-luvulla kehityt iteratiiviset mallit kuten esimerkiksi RUP (Rational Unified Process) ovat vahvasti suunnitelmavetoisia ja niiden perusoletuksena on, että ohjelmistoprojekti on jossain määrin kontrolloitavissa oleva prosessi. Mallit ovat vaiheistettuja ja ne tukevat erityisesti laajojen, pitkäikäisten ohjelmistojen kehitystyötä, jotka vaativat tarkasti kontrolloidun prosessin. Pienten ja keskisuurten ohjelmistojen kehitykseen edellä mainitut mallit ovat kuitenkin osoittautuneet turhan jäykiksi, joten näitä varten on kehitetty menetelmiä, jotka pohjautuvat iteratiiviseen kehitykseen, mutta ovat lähestymistavaltaan kevyempiä ja joustavampia – lähestymistapaa kutsutaan ketteräksi lähestymistavaksi. (Luukkainen & Ilves 2021b.)

2.3 Ketterä lähestymistapa

Ketterän (engl. agile) lähestymistavan myötä ohjelmiston kehitys on siirtynyt suunnitelmavetoisesta arvopohjaisempaan suuntaan ja siinä korostuvat inhimilliset arvot. Ohjelmiston kehitykseen ketterä lähestymistapa tarjoaa joustavan mallin, jossa kehitys tapahtuu asteittain ja empiirisesti. (Schön, Thomaschewski & Escalona 2017a.) Ketterän lähestymistavan perusoletuksena on, että ohjelmistoprojektin ominainen piirre on jatkuva muutos ja näin ollen kyseessä ei siis ole kontrolloitavissa oleva prosessi. Muuttuvia tekijöitä on projektin aikana useita ja niitä on usein mahdotonta ennakoita etukäteen. Myös asiakkaan tarpeet, projektiin osallistuvat henkilöt sekä jatkuvasti kehittyvät toteutusteknologiat muuttuvat ja vaikuttavat jokaiseen projektiin eri tavalla. Näin ollen tarkkaan etukäteissuunnitelmaan perustuvan lähestymistavan sijaan pienten ja keskisuurten ohjelmistoprojektien läpivientiin ja hallinnointiin sopii paremmin empiirinen prosessi, jonka taustalla olevina periaatteina ovat läpinäkyvyys, tarkkaileminen ja mukauttaminen. (Luukkainen & Ilves 2021b.)

Ohjelmiston kehityksen yhteydessä ketterästä lähestymistavasta käytetään nimitystä ketterä ohjelmistokehitys (engl. agile software development). Ketterä ohjelmistokehitys on yleisnimitys tavalle, jolla ohjelmistoja kehitetään. Se on kattotason termi, jonka alle kuuluu joukko ketterän lähestymistavan mukaisia menetelmiä ja niihin liittyviä käytäntöjä. Menetelmiä ovat esimerkiksi Extreme Programming (XP) ja Scrum, jota toisinaan saatetaan käyttää jopa synonyymina ketteryydelle. Eri menetelmiä sovelletaan usein keskenään ja esimerkiksi Lean -ajattelusta peräisin olevaa Kanban -

työhallintatapaa käytetään osana Scrumia. Ketteriä menetelmiä yhdistävät samat arvot ja periaatteet, jotka ohjaavat kehitystyötä ja sen kulkua. (Haikala & Mikkonen 2011, 43–56; Agile Alliance 2021a.) Taustalla on ketterän ohjelmistokehityksen julistus eli Agile Manifesto, jonka arvoissa korostuvat yksilöt ja vuorovaikutus, toimiva ohjelmisto, asiakasyhteistyö sekä kyky vastata muutokseen (Beck ym. 2001a). Arvot konkretisoituvat periaatteisiin, joiden ensisijaisena tavoitteena on pitää asiakas tyytyväisenä toimittamalla toimivia ohjelmistoversioita projektin alusta asti tasaisella tahdilla (Beck ym. 2001b).

Ketterille menetelmille, kuten esimerkiksi Scrumille, onkin ominaista, että ohjelmiston kehitys jaetaan iteraatioihin, joiden aikana ohjelmisto valmistuu pala palalta eli inkrementaalisesti. Iteraatiot ovat kiinteän pituisia jaksoja, joiden kesto vaihtelee projektista riippuen. Tyypillisesti kesto on muutamasta viikosta neljään viikkoa. Scrumin yhteydessä iteraatiosta käytetään nimitystä sprintti. Jokaisen iteraation aikana ohjelmistoon kehitetään lisää toiminnallisuuksia ja ominaisuuksia, ja jokainen iteraatio tuottaa toimivan ohjelmistoversion, joka voidaan demonstroida asiakkaalle. (Agile Alliance 2021c; Agile Alliance 2021e; Agile Alliance 2021f.)

Ketterä ohjelmistokehitys perustuu jatkuvaan palautteeseen ja sen pohjalta tehtävään ohjelmiston kehitykseen, jonka pyrkimyksenä on tuottaa maksimaalista arvoa asiakkaan liiketoiminnalle. Liiketoiminnallisen arvon näkökulmasta esiin nousee DevOps -toimintamalli, jonka ideana on automatisoida testausta ja julkaisua siten, että ohjelmisto siirtyy kehittäjiltä varsinaiseen käyttöön mahdollisimman suoraviivaisesti (Juvonen 2018, 21). Käytettävät menetelmät ja toimintamallit eivät kuitenkaan vielä tee ohjelmistoprojektista ketterää, vaan ketteryys syntyy projektin laajuuteen, budjettiin, laatuun ja aikatauluun liittyvästä joustosta. Molemmilta ohjelmistoprojektin osapuolilta vaaditaan ymmärrystä ketteryyden peruseriaatteista ja sitoutumista niihin. (Ruuska 2020.)

Parhaimmillaan ketterän lähestymistavan avulla voidaan saavuttaa monia etuja ja toteuttaa muutoksia nopealla tahdilla. Tämä edellyttää kuitenkin asiakkaalta priorisointi- ja päätöksentekokykyä. Asiakkaan päässä tärkein henkilö on Scrumin tapauksessa tuoteomistaja (engl. product owner), jonka tulee olla aktiivisesti mukana projektissa. (Tirkkonen 2019; Ruuska 2020.) Ketterän lähestymistavan ohjelmistoprojekteissa ohjelmistoa ei yritetä määritellä ja suunnitella kokonaisuudessaan ennen toteutustyön aloittamista, joten parhaiten ketterä lähestymistapa soveltuu sellaisiin projekteihin, joissa ei ole mahdollista tai tarkoituksenmukaista määritellä ohjelmiston vaatimuksia tarkasti etukäteen (Kulmala 2016).

2.4 Vaatimukset ja vaatimusmäärittely

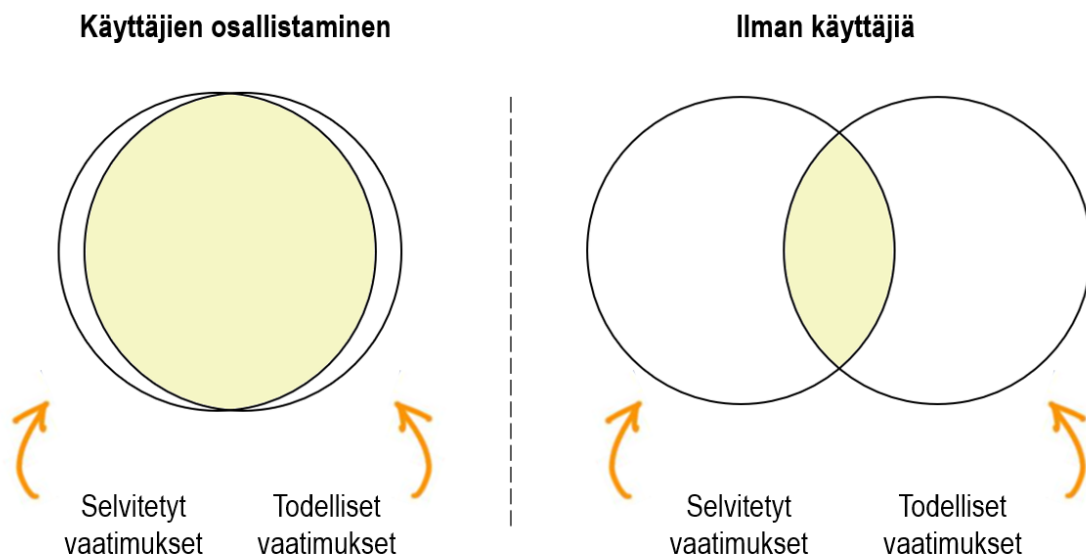
Vaatimuksen määritelmänä voidaan pitää sitä, että se on jotain mitä ohjelmiston tulee tehdä, tai se on ominaisuus, joka ohjelmistolla tulee olla. Vaatimukset pohjautuvat asiakkaan tarpeisiin ja niiden määrittely lähtee korkealta abstraktiotasolta asiakkaan liiketoiminnallisista tavoitteista, jolloin voidaan myös puhua liiketoiminnallisista vaatimuksista (engl. business requirements). Asiakkaan tarpeista lähteviä vaatimuksia voidaan kuitenkin yleisesti kutsua asiakasvaatimuksiksi (engl. customer requirements). Asiakasvaatimukset toteutetaan ohjelmistovaatimuksilla (engl. software requirements), jotka lopulta kuvautuvat joukoksi täsmällisiä ja yksityiskohtaisia teknisiä vaatimuksia. (Haikala & Mikkonen 2011, 61–63; Wiegers & Beatty 2013, luku 1.)

Vaatimukset voidaan luokitella toiminnallisiin vaatimuksiin, ei-toiminnallisiin vaatimuksiin ja reunaehtoihin. Toiminnalliset vaatimukset (engl. functional requirements) kuvaavat ohjelmiston toimintaa eli niitä asioita, joita ohjelmistolla voidaan tehdä. Ei-toiminnalliset vaatimukset (engl. non-functional requirements) kuvaavat ohjelmiston toiminnan kannalta tärkeitä ominaisuuksia, kuten käytettävyyttä ja suorituskykyä sekä riippuvuuksia muista ohjelmistoista. Reunaehdot (engl. constraints) ovat rajoitteita, joilla on vaikutusta ohjelmiston kehitystyöhön. Tällaisia rajoitteita voivat olla esimerkiksi ohjelmointikieli tai palvelinympäristö. (Haikala & Mikkonen 2011, 61–63; Wiegers & Beatty 2013, luku 1.)

Toteutuessaan vaatimukset tuottavat arvoa asiakkaan liiketoiminnalle tai ohjelmiston tuleville käyttäjille. Tulevien käyttäjien tarpeista lähteviä vaatimuksia voidaan kutsua yleisesti käyttäjävaatimuksiksi (engl. user requirements). Käyttäjävaatimukset ovat tulevien käyttäjien näkökulmasta kuvattuja ohjelmiston toiminnallisuuksia ja ne kuvaavat abstraktiotasolla jonkin tietyn tavoitteen, joka käyttäjän on kyettävä ohjelmistolla suorittamaan. Useimmissa ohjelmistoprojekteissa käyttäjiä on tarkasteltava käyttäjäryhmittäin, joten käyttäjävaatimukset kuvataan tyypillisesti jonkun tietyn käyttäjäryhmän näkökulmasta. (Wiegers & Beatty 2013, luku 1.)

Ohjelmistoprojektissa on usein myös muita sidosryhmiä (engl. stakeholders), joiden tarpeet on otettava huomioon. Sidosryhmiä ovat kaikki ne tahot, jotka ovat suoraan tai epäsuorasti tekemisissä kehitettävän ohjelmiston kanssa. Kaikkia tahoja ei välttämättä kyetä tunnistamaan heti alkuvaiheessa, mutta tärkeimmät sidosryhmät pitäisi kuitenkin saada tunnistettua mahdollisimman varhaisessa vaiheessa. Oleellista on tiedostaa, että sidosryhmien tarpeet saattavat olla keskenään ristiriitaisia. (Haikala & Mikkonen 2011, 155; Wiegers & Beatty 2013, luku 1.) Jos kyseessä on vanhan

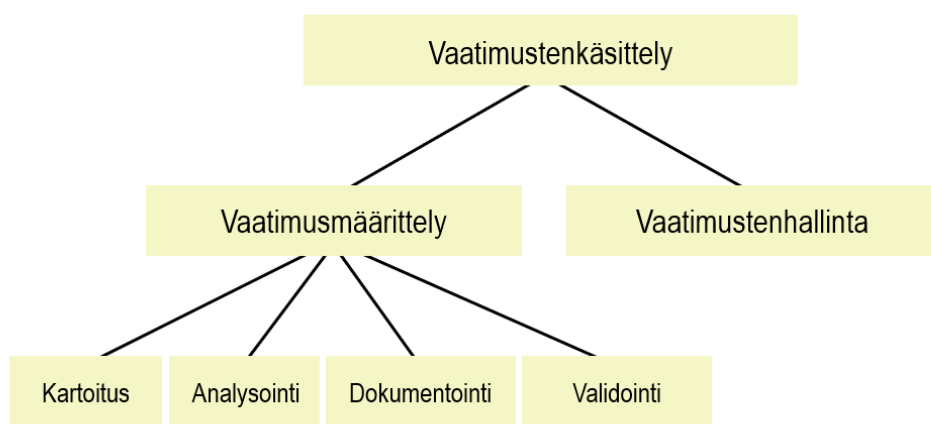
ohjelmiston korvaaminen uudella, ja samalla nykyisen työn tehostaminen ja automatisointi, niin työskentelyprosessia tulisi tarkastella eri sidosryhmien näkökulmasta. Keskeiset sidosryhmät tulisi osallistaa mukaan potentiaalisten automatisointikohteiden kartoittamiseen, ja lähtökohdaksi tulisi ottaa laajemmin koko työskentelyprosessi ja sen läpiviennin parantaminen. (Capgemini 2019.) Sidosryhmät ovat siis vaatimusten lähde ja yhtenä tärkeimmistä voidaan pitää ohjelmiston tulevia käyttäjiä. Käyttäjien osallistaminen mukaan vaatimusmäärittelyyn auttaa ehkäisemään ristiriidat toteutuksen ja todellisten tarpeiden välillä (KUVIO 1) (Kähönen 2016).



KUVIO 1. Käyttäjien osallistaminen vaatimusmäärittelyyn auttaa ehkäisemään ristiriidat toteutuksen ja todellisten tarpeiden välillä (Kähönen 2016).

Vaatimusmäärittely (engl. requirements definition) on prosessi, jonka tarkoituksena on selvittää vaatimukset kehitettävälle ohjelmistolle. Prosessi pitää sisällään vaatimusten kartoitusta, dokumentointia, analysointia ja validointia. Vaatimuksia voidaan kartoittaa eri menetelmin, jonka jälkeen kartoitetut vaatimukset dokumentoidaan. Analysoinnin perusteella vaatimuksia tarkennetaan, samalla kun niiden keskinäisiä suhteita ja prioriteetteja selvitetään. Tässä yhteydessä myös varmistetaan, että vaatimusten toteutuminen on ylipäättään mahdollista. Validoinnin myötä on tarkoitus saavuttaa yhteinen ymmärrys vaatimuksesta. (Haikala & Mikkonen 2011, 65–67; Wiegers & Beatty 2013, luku 1.)

Haikalan ja Mikkosen (2011, 65) mukaan vaatimusmäärittely voidaan nähdä osana vaatimusten-
 käsittelyä (engl. requirements engineering), joka pitää sisällään myös vaatimustenhallinnan (engl.
 requirements management). Ohjelmistotuotannon käytännöt -kirjassa esitetty kuva (KUVIO 2) pe-
 rustuu alun perin Dean Leffingwellin Scaling software agility -kirjaan, ja täysin saman tyylinen kuva
 löytyy myös Wiegersin ja Beattyn (2013, luku 1) Software requirements -kirjasta. Luukkaisen ja
 Ilveksen (2021a) mukaan sanaa vaatimusmäärittely voidaan kuitenkin käyttää silloin kun puhutaan
 kokonaisuudesta, joka pitää sisällään kaikki vaatimuksiin liittyvät työvaiheet eli kartoituksen, ana-
 lysoinnin, dokumentoinnin, validoinnin sekä hallinnoinnin.



KUVIO 2. Vaatimusmäärittely voidaan nähdä osana vaatimusten käsittelyä, joka pitää sisällään myös vaatimustenhallinnan (Haikala & Mikkonen 2011, 65; Wiegers & Beatty 2013, luku 1).

Vaatimuksiin liittyvä terminologia vaihtelee hieman lähteestä riippuen, mutta Wiegersin ja Beattyn (2013, luku 1) mukaan kaikki vaatimusten käsittelyn alle lukeutuvat työvaiheet ovat kuitenkin sellaisia, että ne kuuluvat jokaiseen ohjelmistoprojektiin. Dokumentaation määrä puolestaan riippuu täysin kehitettävästä ohjelmistosta ja vaatimusmäärittelyn tavoitteena saattaa joissain tapauksissa olla kirjallisen vaatimusdokumentin laatiminen, joka on usein oma projektinsa (Haikala & Mikkonen 2011, 64–65). Ketterässä ohjelmistoprojektissa vaatimusmäärittelyä tapahtuu kuitenkin jokaisen iteraation yhteydessä ja prosessin eri työvaiheet voidaan implementoida osaksi käytettävää menetelmää, kuten esimerkiksi Scrumia (Lucia & Qusef 2010; Alfame 2021).

Kattavaa kirjallista vaatimusdokumenttia ei siis ketterässä ohjelmistoprojektissa ole tapana tehdä, vaan vaatimukset listataan tuotteen kehitysjonoon (engl. product backlog), joka elää koko projektin

ajan. Scrumin tapauksessa tuotteen kehitysjonon ylläpidosta vastaa tuoteomistaja, joka toimii samalla myös rajapintana kaikkiin eri sidosryhmiin. (Haikala & Mikkonen 2011, 48, 65.) Ketterässä ohjelmistoprojektissa vaatimuksia tulee lisää sitä mukaa kun ohjelmiston kehitys etenee ja uusia tarpeita tunnistetaan. Tuotteen kehitysjonossa vaatimuksia priorisoidaan, jonka perusteella ne jaetaan iteraatioihin. Kuhunkin iteraatioon valitaan toteutettavaksi ne vaatimukset, jotka tuottavat eniten arvoa eli ovat asiakkaan näkökulmasta tärkeimpiä. Vaatimuksia ja niihin liittyviä työmääräarvioita myös tarkennetaan, mikä tarkoittaa sitä, että mitä lähempänä vaatimukset ovat priorisoidun listan kärkeä, sitä tarkempia ja yksityiskohtaisempia ne ovat. (Agile Alliance 2021g; Luukkainen & Ilves 2021a.)

Tuotteen kehitysjonoon vaatimukset voidaan dokumentoida esimerkiksi käyttäjätarinoina (engl. user stories) (Agile Alliance 2021g). Käyttäjätarinoiden kirjalliset kuvaukset ovat luonnollisella kielellä kirjoitettuja ja ne kuvaavat kehitettävän ohjelmiston toimintaa tulevien käyttäjien näkökulmasta. Alun perin käyttäjätarinat ovat lähtöisin Extreme Programmingista. (Cohn 2009, luku 1; Agile Alliance 2021j.) Nykyään niitä käytetään kuitenkin yleisesti ketterässä vaatimusmäärittelyssä, jolla tarkoitetaan tässä opinnäytetyössä vaatimusmäärittelyä, jota tehdään osana ketterää ohjelmistoprojektia. Ketterässä vaatimusmäärittelyssä noudatetaan ketterän ohjelmistokehityksen arvoja ja periaatteita, ja siinä on myös vahvasti läsnä ihmiskeskeinen suunnittelu (engl. human-centered design) (Schön, Thomaschewski & Escalona 2017b). Ketterä vaatimusmäärittely on ennen kaikkea ihmisten välistä yhteistyötä, jossa kommunikaatio on avainasemassa (Schön ym. 2017a).

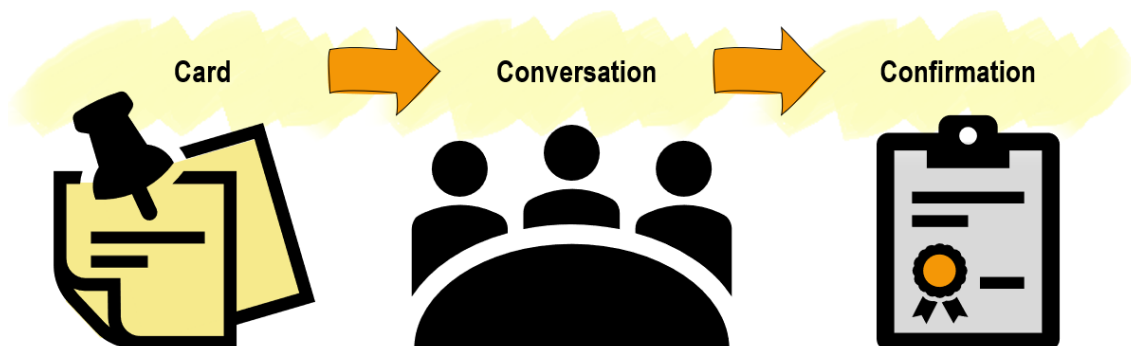
Käyttäjätarinat ovat avain kommunikaatioon, ja ne tarjoavat yhteisen kielen kehittäjien ja tulevien käyttäjien väliseen yhteistyöhön (Wake 2003; Leffingwell 2011, luku 6). Käyttäjätarinoiden avulla käyttäjät saadaan osallistettua mukaan ohjelmiston kehitykseen ja niiden avulla käyttäjien tarpeet saadaan myös pidettyä kehityksen aikana päällimmäisenä mielessä (Cohn 2009, luku 13; Garreta-Domingo 2021). Käyttäjätarinoiden ensisijainen tarkoitus on auttaa siirtämään painopiste suulliseen kommunikaatioon pelkän kirjallisen dokumentaation sijaan (Mountain Goat Software 2021). Vaatimusmäärittelyyn käyttäjätarinat tarjoavat työvälineen, jonka avulla toiminnallisuudet voidaan määrittää ketterästi ja näin ollen ohjelmisto voidaan rakentaa osissa (Cohn 2009, luku 1; Agile Alliance 2021j). Käyttäjätarinat auttavat myös varmistamaan, että ohjelmisto täyttää sille asetetut vaatimukset (Visual Paradigm 2021b). Luukkaisen ja Ilveksen (2021a) mukaan käyttäjätarinoita voidaan pitää ketterän vaatimusmäärittelyn tärkeimpänä työvälineenä.

3 KÄYTTÄJÄTARINAT VAATIMUSMÄÄRITTELYN TYÖVÄLINEENÄ

3.1 Käyttäjätarinat

Käyttäjätarinoiden näkyvin osa on kirjallinen kuvaus, mutta se ei kuitenkaan ole käyttäjätarinoiden tärkein osa. Nimensä käyttäjätarinat ovatkin saaneet siitä, kuinka niitä on tarkoitus käyttää. Käyttäjätarinoiden idean isän, XP-menetelmän kehittäjän Kent Beckin, mukaan käyttäjätarina on ikään kuin muistilappu. Ideana on kirjata asioita muistiin, jotta yksityiskohdista muistetaan keskustella siinä vaiheessa, kun ne ollaan kehityksen aikana toteuttamassa. Käyttäjätarinat eivät siis ole yksityiskohtaisia vaatimuksia, vaan niiden tarkoituksena on herättää keskustelua, ja keskusteluita voidaankin pitää käyttäjätarinoiden tärkeimpänä osana. (Cohn 2009, luku 1; Patton & Economy 2014, luku 6.)

Yksinkertaisen idean taustalla on ajatus siitä, että keskustelemalla asioista on mahdollisuus rakentaa yhteistä ymmärrystä ja löytää paras mahdollinen ratkaisu toiminnallisuuden toteuttamiseen. Beck nimesi ideansa alkuun pelkiksi tarinoiksi, mutta siihen lisättiin hyvin pian käyttäjä -etuliite muistuttamaan siitä, että keskustelut tulee käydä nimenomaan käyttäjien näkökulmasta. (Patton & Economy 2014, luku 7.) Kirjallisen kuvauksen ja keskusteluiden lisäksi käyttäjätarinoissa on kolmaskin vaihe, jonka tarkoituksena on vahvistaa, että yhteinen ymmärrys on saavutettu. Ron Jeffries (2001) on nimennyt nämä kolme vaihetta (KUVIO 3) seuraavasti: Card, Conversation, Confirmation. (Cohn 2009, luku 1; Patton & Economy 2014, luku 6; Agile Alliance 2021i.) Näitä kolmea vaihetta on avattu seuraavalla sivulla hieman tarkemmin.



KUVIO 3. Käyttäjätarinoiden kolme vaihetta, jotka Ron Jeffries (2001) on nimennyt seuraavasti: Card, Conversation, Confirmation.

Kirjallinen kuvaus tai kortti (engl. card) on käyttäjätarinan näkyvin osa. Kirjallinen kuvaus kirjoitetaan ohjelmiston tulevien käyttäjien näkökulmasta ja se sisältää käyttäjäroolin, sen mitä käyttäjäryhmä ohjelmistolla haluaa tehdä ja miksi. Kuvaus kirjoitetaan luonnollisella kielellä ja se voidaan kirjoittaa esimerkiksi muistilapulle, pahvikorttiin tai hyödyntäen jotakin digitaalista työkalua. Kirjallisen kuvauksen tehtävänä on muistuttaa mihin toiminnallisuuteen se liittyy, jotta toiminnallisuudesta ja siihen liittyvistä yksityiskohdista muistetaan keskustella tarkemmin myöhemmässä vaiheessa. (Jeffries 2001; Cohn 2009, luku 1; Patton & Economy 2014, luku 6.) Kirjallisista kuvauksista voidaan muodostaa myös käyttäjätarinakartta (engl. user story map), joka tarjoaa tuotteen kehitysjonoa visuaalisemman näkymän auttaen samalla hahmottamaan suuremman kokonaisuuden sekä huomaamaan puutteet tuotteen kehitysjonossa (Patton & Economy 2014, luku 2; Agile Alliance 2021h).

Keskustelun (engl. conversation) tarkoituksena on keskustella kirjallisesta kuvauksesta ja täsmentää siihen liittyviä yksityiskohtia. Keskustelut ovat käyttäjätarinoiden tärkein osa, koska niiden avulla on mahdollista saavuttaa yhteinen ymmärrys kehitettävästä toiminnallisuudesta ja toiminnallisuuden liittyvistä yksityiskohdista. Keskusteluihin voidaan osallistaa useita henkilöitä, ja ne käydään niiden käyttäjien kanssa, joita toiminnallisuus koskee. Keskustelut käydään viimeistään siinä vaiheessa, kun kyseinen toiminnallisuus on priorisoitu kehityksen aikana tuotteen kehitysjonossa toteutettavaksi. Keskusteluita on suositeltavaa täydentää erilaisilla muistiinpanoilla, kuten esimerkiksi sanoilla ja kuvilla, jotka auttavat rakentamaan yhteistä ymmärrystä ja myöhemmin muistamaan keskusteluiden yksityiskohtia. (Jeffries 2001; Cohn 2009, luku 1; Patton & Economy 2014, luku 6.)

Vahvistuksen (engl. confirmation) tarkoituksena on vahvistaa hyväksymistestien avulla, että toiminnallisuus on toteutettu niin, että se täyttää sille asetut kriteerit. Hyväksymistestit koostuvat kriteereistä eli listasta tarkistettavia asioita, jotka on kirjattu muistiin keskusteluiden aikana. Kriteerit tarkentavat kirjallista kuvausta ja niiden kirjoittaminen helpottaa myös toteutukseen kuluvan työmäärän arviointia. Kriteerit määrittävät sen, kuinka ohjelmiston tulee toimia ja niiden kirjaaminen on tärkeää, jotta molemmilla osapuolilla on yhteinen ymmärrys toteutettavasta toiminnallisuudesta. Toiminnallisuuden voidaan vahvistaa olevan toteutettu vasta sitten, kun se läpäisee siihen liittyvän hyväksyntätestin eli testatessa täyttää sille määritetyt kriteerit. Hyväksymistestit voivat olla joko automaattisesti tai manuaalisesti suoritettavia. (Jeffries 2001; Cohn 2009, luku 1; Patton & Economy 2014, luku 6.) Suurin osa testeistä voidaan kuitenkin automatisoida, joten aina kun mahdollista, niin tähän tulisi pyrkiä (Cohn 2009, luku 2).

3.2 Kartoittaminen

Käyttäjätarinoita voidaan kartoittaa eri tavoin, kuten esimerkiksi haastatteluiden ja kyselylomakkeiden avulla sekä havainnoimalla ohjelmiston tulevien käyttäjien toimintaa ja järjestämällä näille erilaisia työpajoja. Tarkoituksena on tunnistaa ohjelmiston tulevien käyttäjien tarpeita ja selvittää käyttäjävaatimukset, joiden pohjalta voidaan muodostaa käyttäjätarinoita. Käyttäjätarinoita ei ole tarkoitus kartoittaa yhdellä kertaa, joten menetelmien on oltava riittävän kevyitä ja häiritsemättömiä, jotta niitä voidaan käyttää koko ohjelmistoprojektin ajan. (Cohn 2009, luku 4.) Kartoitukseen käytettävät menetelmät ovat ennen kaikkea apuvälineitä, joiden avulla on tarkoitus saada aikaan vuoropuhelu ohjelmiston tulevien käyttäjien kanssa (Robertson & Robertson 2013, luku 5). Tässä opinäytetyössä esitetyt kartoitusmenetelmät perustuvat User stories applied -kirjassa esitettyihin menetelmiin, joita kirjan kirjoittaja, ja alan suurimmaksi auktoriteetiksi tituleerattu Mike Cohn, pitää käyttäjätarinoiden näkökulmasta tärkeimpinä. Cohn ei kuitenkaan ole avannut kirjassaan eri menetelmien käyttöä kovin tarkalla tasolla, joten lisätietoa kyseisistä menetelmistä on hankittu myös muista lähteistä.

Kartoitusmenetelmät voidaan yleisesti jakaa perinteisiin menetelmiin, sovellettuihin menetelmiin ja innovatiivisiin menetelmiin. Perinteisiä menetelmiä ovat muun muassa haastattelut ja kyselyt. Sovellettuja menetelmiä ovat muun muassa havainnointi ja käyttökokeututkimus. Innovatiivisia menetelmiä ovat muun muassa erilaiset työpajat. (Hanington 2003.) Käytettävät menetelmät tulee suunnitella projektikohtaisesti ja kehitettävästä ohjelmistosta riippuen, koska samat tekniikat eivät sovellu kaikkiin projekteihin. Olennaista on myös huomioida, että ohjelmiston tulevat käyttäjät eivät välttämättä osaa kertoa suoraan todellisia tarpeitaan. (Cohn 2009, luku 4.) Kirjassaan Cohn (2009, luku 4) lainaa Suzanne ja James Robertsonin (2013, luku 5) käyttämää troolauksen termiä, joka viittaa kalastusmenetelmään. Metaforalla pyritään kuvaamaan muun muassa sitä, kuinka käyttäjävaatimuksia voidaan "troolata" eri kokoisilla verkoilla. Alkuun voidaan käyttää suuremman silmäkoon verkkoja, jolloin haaviin jää suuremmat vaatimukset ja myöhemmin voidaan siirtyä pienemmän silmäkoon verkkoihin. Parhaat tulokset saavutetaan siis usein silloin kun ohjelmistoprojektin aikana käytetään useampia eri kartoitusmenetelmiä. (Cohn 2009, luku 4.)

Kartoitusmenetelmistä haastattelut ovat yksi yleisimmin käytetyistä ja käyttäjätarinoiden kartoituksessa niitä voidaan pitää oletusmenetelmänä. Vuoropuhelun kannalta haastattelut ovat hyvä vaihtoehto, koska haastatteluiden tarkoituksena on nimenomaan syventyä ymmärtämään haastatelta-

via. Oleellinen asia on valita haastatteluihin oikeat henkilöt, joiden tulisi edustaa ohjelmiston todellisia käyttäjiä ja eri käyttäjärooleja. (Cohn 2009, luku 4.) Haastattelut voidaan toteuttaa fyysisesti samassa tilassa haastateltavan kanssa tai videopuhelun välityksellä. Videopuheluun sopivia työkaluja ovat esimerkiksi Google Meet, Zoom, Whereby tai Teams. Videopuheluissa haastattelut kannattaa tallentaa, koska se mahdollistaa haastatteluun palaamisen myöhemmin uudestaan. Tallennukseen on kuitenkin muistettava kysyä lupa. (Vehviläinen 2021.)

Haastattelut voidaan toteuttaa yksilö- tai ryhmähaastatteluina. Ryhmähaastatteluissa on tarkoituksena haastatella useampia henkilöitä samalla kertaa, jolloin optimaalisena ryhmän kokona voidaan pitää neljää tai viittä haastateltavaa. Ryhmähaastattelut ovat kustannustehokas menetelmä, joka tarjoaa haastateltavalle tilaisuuden vertailla omia kokemuksiaan ja näkemyksiään muiden haastateltavien kanssa. Haasteena on kuitenkin se, että joku haastateltavista saattaa ottaa johtajan roolin ohjaamalla keskustelua ja mielipiteitä. Palvelumuotoiluun keskittyneen suunnittelutoimisto Palon kokemusten mukaan yksilöhaastattelut toimivat yleensä aina, olipa aiheena mikä tahansa. Yksilöhaastatteluissa tarkoituksena on haastatella vain yhtä henkilöä kerrallaan. Tämä antaa mahdollisuuden syventyä ymmärtämään paremmin haastateltavaa. Yksilöhaastatteluun voidaan yhdistää myös käyttötilanteiden havainnointia, jolloin menetelmästä voidaan käyttää nimeä käyttökokemustutkimus. (Palvelumuotoilu Palo 2021.)

Haastattelut voivat olla luonteeltaan esimerkiksi teemahaastatteluita, joissa esitetään pääasiassa avoimia kysymyksiä, joihin ei ole valmiita vastausvaihtoehtoja. Cohnin (2009, luku 4) mukaan haastatteluissa kannattaa juuri nimenomaan esittää avoimia kysymyksiä, jotta haastateltavat voivat ilmaista syvällisempiä ajatuksiaan ja mielipiteitään. Teemahaastattelut etenevät hyvin vapaamuotoisesti ennalta määriteltujen teemojen pohjalta, eikä kysymysmuotoja tai -järjestystä tarvitse etukäteen tarkasti määritellä. Haastattelijalla voi olla kysymyslista, joka toimii enemminkin muistilistana. Teemahaastattelun etuna on se, että haastattelijalla on mahdollisuus kysyä tarkentavia kysymyksiä, jotka auttavat vastausten ymmärtämistä ja näin ollen päästään myös keskustelemaan asioista syvemmällä tasolla. (Palvelumuotoilu Palo 2021.)

Vapaamuotoiseen teemahaastatteluun kannattaa kuitenkin laatia etukäteen kirjallinen runko eli haastattelun käsikirjoitus ja mikäli haastattelussa on tarkoitus hyödyntää luovia tai toiminnallisia tehtäviä, niin niiden käyttäminen kannattaa suunnitella etukäteen huolella. Tehtävät voivat olla esimerkiksi erilaisia kuvatehtäviä, joiden avulla keskustelu saadaan vietyä uudelle tasolle. Haastattelijan tehtävänä on pyrkiä luomaan haastatteluiden ilmapiiristä rento ja luottamuksellinen, joten

alussa on tärkeää muistaa kertoa ainakin haastattelun taustasta sekä mihin haastattelun tietoja tullaan tarkalleen ottaen käyttämään. Haastattelun aikana haastattelijan tulisi pysyä mahdollisimman neutraalina eli haastattelija ei saisi paljastaa haastateltavalle omaa mielipidettään. (Palvelumuotoilu Palo 2021.) Haastatteluista ei kuitenkaan ole mitään hyötyä, jos yksi oleellinen asia unohtuu. Kuuntelu on yksi tärkeimmistä asioista, jonka myötä voidaan saavuttaa syvälinen ymmärrys käyttäjän tarpeesta. *Mastering the requirements process* -kirjassaan Suzanne ja James Robertson (2013, luku 5) muistuttavat G. K. Chestertonin sanoin siitä, että ihmisellä on kaksi korvaa ja yksi suu, ja näitä tulisi käyttää siinä suhteessa. (Robertson & Robertson 2013, luku 5.) Haastattelijan tärkein tehtävä on siis kuunnella mitä haastateltavat rehellisesti ajattelevat ja miten he käsiteltävät asiat kokevat. Lopetusvaiheessa haastattelijan on hyvä kertoa kuinka asiat haastatteluiden jälkeen etenevät, ja lopettaa haastattelu kiitoksiin. (Palvelumuotoilu Palo 2021.)

Kartoitusmenetelmänä kyselyt ovat kustannustehokas vaihtoehto, koska niiden avulla saadaan kerättyä tietoa samanaikaisesti suurelta joukolta käyttäjiä. Tarvittaessa kyselyt voidaan kohdentaa pienemmälle joukolle, kuten esimerkiksi jollekin tietylle käyttäjäryhmälle, jolta voidaan kerätä kyselyn avulla täsmätietoa oikealla hetkellä. Ideana on pyytää käyttäjiltä vastauksia ennalta määriteltyihin kysymyksiin. Kyselyiden avulla voidaan kysyä avoimia kysymyksiä, mutta vastausvaihtoehdot voidaan toteuttaa myös jollakin etukäteen rajatulla vaihtoehdolla, kuten esimerkiksi numeroskaalalla tai kuvallisella vaihtoehdolla. Kysymysten on hyvä olla erityyppisiä ja vastausvaihtoehtoja tulee olla useita. Lisäksi kysymykset tulee esittää niin, että ne ovat yksiselitteisiä. Kyselyn käytettävyyttä ja ymmärrettävyyttä on myös suositeltavaa testata ennen kyselyn lähettämistä. Kyselyt toteutetaan sähköisesti ja niitä varten on olemassa lukuisia erilaisia työkaluja, kuten esimerkiksi Google Forms, Limesurvey, Survey Monkey, SurveyPal, Typeform ja Webropol. Työkalusta riippuen saattaa tarjolla olla erilaisia suodatusmahdollisuuksia sekä mahdollisuus tulosten analysointiin. Vastausten käsittelyyn on kuitenkin hyvä olla selkeä suunnitelma, ja sitä varten tulee varata riittävästi resursseja erityisesti silloin kun kysely sisältää avoimia kysymyksiä ja vastauksia on odotettavissa runsaasti. (Innanen 2021.)

Vuoropuhelun näkökulmasta kyselyt ovat kuitenkin yksipuolisuutensa takia huono vaihtoehto. Kyselyt eivät mahdollista samalaista tarkentavien kysymysten esittämistä kuin haastattelut, joten syvälinen ymmärryksen saavuttaminen kyselyiden avulla on haastavaa. Kyselyt eivät myöskään yleensä sovellu ensisijaiseksi menetelmäksi, mikäli tarkoituksena on kartoittaa täysin uusia käyttäjätarinoita. Kyselyt voivat kuitenkin olla tehokas keino hankkia lisätietoa jo olemassa oleviin käyttäjätarinoihin, jotka on kirjoitettu korkealla tasolla esimerkiksi haastatteluiden pohjalta. Kyselyt voivat

siis olla perusteltu menetelmä, kun tarvitaan useilta käyttäjiltä tarkentavia vastauksia tai mielipiteitä. Mikäli kehitettävällä ohjelmistolla on paljon käyttäjiä, niin kyselyiden avulla voidaan saada apua tarvittaessa muun muassa käyttäjätarinoiden priorisointiin. (Cohn 2009, luku 4.)

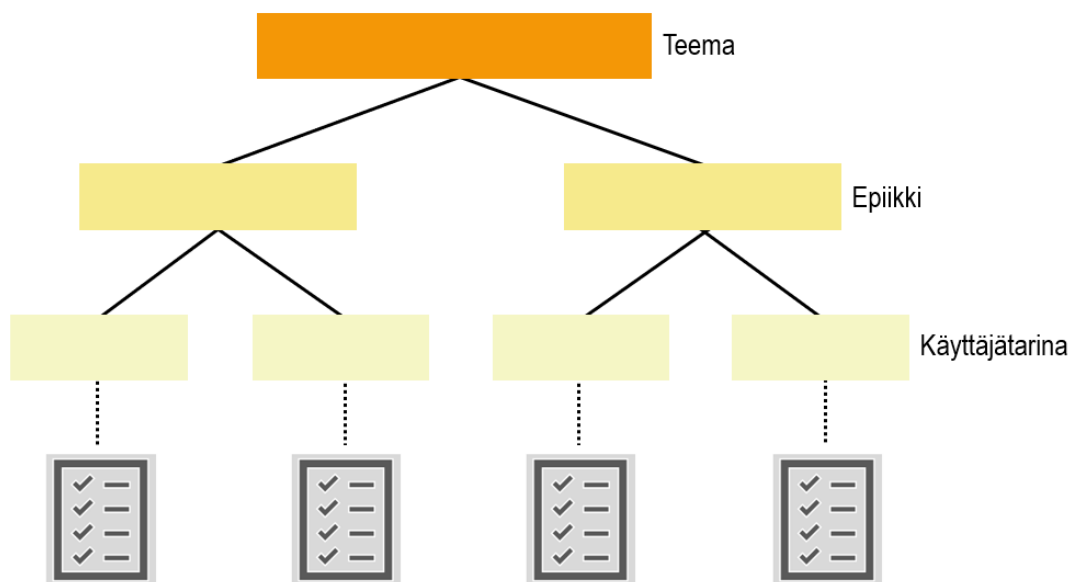
Kartoitusmenetelmistä havainnointi on ehdottomasti paras tapa saada oivalluksia siitä, mitä oikeasti tapahtuu ja kuinka käyttäjät toimivat vuorovaikutuksessa ohjelmiston kanssa. Havainnointi tapahtuu oikeassa kontekstissa ja ympäristössä, ja usein fyysisesti samassa tilassa käyttäjien kanssa. Havainnointi mahdollistaa nopean ja suoran palautteen sekä sellaisten asioiden ja tarpeiden huomioimisen mitä käyttäjät eivät välttämättä itse osaa tuoda esille. Mikäli havainnointiin on mahdollisuus, niin sitä kannattaa ehdottomasti tehdä. (Cohn 2009, luku 4.) Jos kyseessä on vanhan ohjelmiston korvaaminen uudella ja samalla koko työskentelyprosessin muutos, niin käyttäjien työskentelyä voidaan havainnoida nykyisen ohjelmiston parissa, mutta tärkeää on muistaa tarkastella laajemmin myös itse työskentelyprosessia (Luukkainen & Ilves 2021a).

Käyttäjiä havainnoimalla voidaan tunnistaa tarpeita, jotka jäisivät muuten huomioimatta. Menetelmän avulla voidaan selvittää kuinka käyttäjät oikeasti toimivat, eikä vain sitä, kuinka he sanovat toimivansa. Käyttötilanteiden havainnointiin voidaan yhdistää myös haastattelua, ja kuten aikaisemmin yksilöhaastatteluiden yhteydessä todettiin, niin tästä menetelmästä voidaan käyttää nimeä käyttökokemustutkimus. Käyttökokemustutkimus on siis menetelmien yhdistelmä, jossa käyttäjä pyydetään näyttämään kuinka hän ohjelmistoa tai esimerkiksi prototyyppiä käyttää, ja ajattelemaan samalla ääneen sekä vastaamaan haastattelijan tarkentaviin kysymyksiin. (Komulainen 2021.)

Kartoitusmenetelmänä työpajat voivat olla esimerkiksi käyttäjätarinoiden kirjoitustyöpajoja, joihin on tarkoitus osallistaa kerralla useita henkilöitä. Ideana on kirjoittaa mahdollisimman paljon uusia käyttäjätarinoita, mahdollisimman lyhyessä ajassa. Oikein toteutettuna käyttäjätarinoiden kirjoitustyöpajat ovat Cohnin (2009, luku 4) mukaan tehokkain tapa tarpeiden tunnistamiseen ja käyttäjätarinoiden kirjoittamiseen. Kirjoitustyöpajoissa käyttäjätarinat kirjoitetaan korkealla tasolla eli tarkoituksena ei ole silloin keskittyä yksityiskohtiin. (Cohn 2009, luku 4.) Tyypillisesti käyttäjätarinoiden kirjoitustyöpaja järjestetään lähellä projektin aloitusta (Mountain Goat Software 2021). Yleisesti työpajat voidaan toteuttaa joko fyysisesti samassa tilassa käyttäjien kanssa tai etätyöpajoina. Etätyöpaja voidaan toteuttaa esimerkiksi Teamsin avulla yhdistämällä siihen jokin yhteistyöalusta, kuten esimerkiksi Miro tai Mural, jotka ovat kuin virtuaalisia post-it-tauluja. (Vehviläinen 2021.)

3.3 Kirjoittaminen

Käyttäjätarinoita kirjoitetaan koko ohjelmistoprojektin ajan ja kirjalliset kuvaukset voidaan kirjoittaa eri tarkkuustasoilla. Alkuun kuvaukset on usein kirjoitettu korkealla tasolla, mikä tarkoittaa sitä, että käyttäjätarinat ovat vielä liian suuria toteutettavaksi. Tällaisesta suuresta kuvauksesta voidaan käyttää sanaa epiikki tai eepos (engl. epic). (Mountain Goat Software 2021.) Epiikit voidaan hahmottaa kuuluvaksi johonkin tiettyyn teeman (engl. theme), jolla tarkoitetaan kokonaisuutta, johon epiikit liittyvät. Kirjoitettu kuvaus on siis projektin aikana todennäköisesti ensin epiikki, joka tarkentuessaan jakaantuu useammaksi tarkemmaksi kuvaukseksi eli käyttäjätarinaksi. (Agile Alliance 2021b; Visual Paradigm 2021a.) Eri tasojen muodostama hierarkia on kuvattu alla olevassa kuvassa (KUVIO 4).

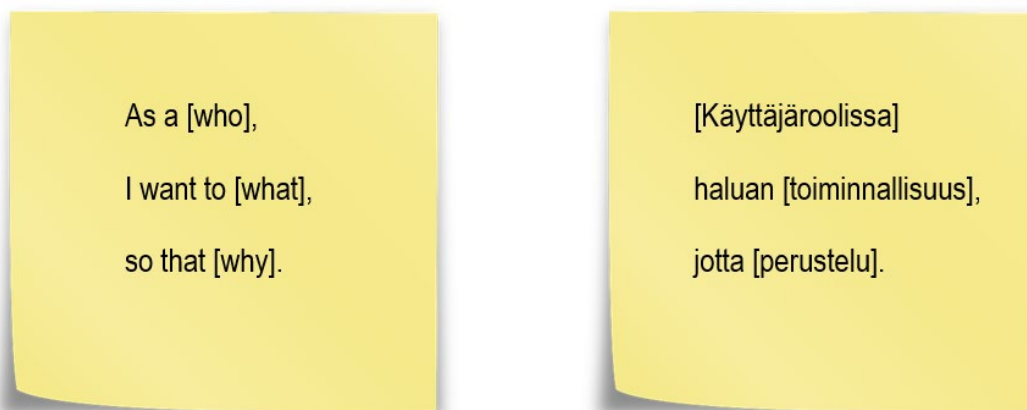


KUVIO 4. Epiikit voidaan hahmottaa kuuluvaksi johonkin tiettyyn teeman ja myöhemmin kun epiikit tarkentuvat, niistä muodostuu käyttäjätarinoita (Visual Paradigm 2021a).

Ajallisesti kokonaisen teeman toteutukseen kuluva aika voi olla jopa useita kuukausia. Epiikin toteutukseen kuluva aika on puolestaan viikkoja, mikä tarkoittaa sitä, että epiikit voivat olla esimerkiksi niin suuria tai monimutkaisia kuvauksia, että niitä ei ole ajallisesti mahdollista toteuttaa yhden iteraation aikana. Kun puhutaan käyttäjätarinan toteutukseen kuluva ajasta, niin voidaan puhua päivistä. Käyttäjätarinan voidaankin nähdä olevan pienin yksikkö, joka voidaan toteuttaa yhden iteraation aikana. Jokaiseen käyttäjätarinaan liittyy lisäksi useampia tehtäviä (engl. tasks), jotka kuvaavat kuinka käyttäjätarina tullaan kehittäjien toimesta teknisesti toteuttamaan. Tehtävien toteutukseen kuluva aika voidaan ilmaista tunneissa. (Visual Paradigm 2021a.)

Käyttäjätarinoita voi kirjoittaa projektin aikana oikeastaan kuka tahansa, mutta päävastuu niiden kirjoittamisesta on asiakkaalla. Scrumia käytettäessä tuoteomistaja on asiakkaan päässä henkilö, joka vastaa kehitysjonon ylläpidosta ja käyttäjätarinoiden priorisoinnista, joten on tärkeää, että hän ymmärtää jokaisen käyttäjätarinan. Kirjoittamisessa on hyvä pitää mielessä, että kirjallinen kuvaus ei saisi sisältää liikaa yksityiskohtia, koska niistä on tarkoitus keskustella vasta myöhemmässä vaiheessa. Oleellista on myös tiedostaa, että tärkeämpää kuitenkin kuin itse käyttäjätarinoiden kirjoittaminen, on se, ketkä ovat mukana myöhemmin käytävissä keskusteluissa. (Cohn 2009, luku 7; Mountain Goat Software 2021.)

Kirjallinen kuvaus sisältää tyypillisesti lauseen tai kaksi, ja se kiteyttää toiminnallisuuteen liittyvän tarpeen, joka on kirjoitettu tulevien käyttäjien näkökulmasta. Kuvaus sisältää käyttäjärooliin ja siitä tulisi käydä ilmi myös käyttäjätarinan tuottama arvo. Kirjoittamiseen on olemassa englanninkielinen malli, jota voidaan hyödyntää. Malli auttaa muotoilemaan kuvauksen niin, että se kertoo kenelle toiminallisuus tuottaa arvoa ja miksi. (Cohn 2009, luku 2; Agile Alliance 2021k.) Luukkaisen ja Ilveksen (2021a) mukaan malli ei kuitenkaan sellaisenaan taivu hyvin suomenkielisiin kuvauksiin. Alla olevassa kuvassa (KUVIO 5) on vasemmalla puolella esitetty alkuperäinen englanninkielinen malli ja oikealla puolella suomenkielinen vastine.



KUVIO 5. Kirjallinen kuvaus kertoo kuka kyseistä toiminnallisuutta tarvitsee, mihin ja miksi.

Käyttäjätarinoiden kirjoittamiseen ei ole olemassa yhtä oikeaa tapaa, mutta hyvillä käyttäjätarinoilla voidaan nähdä olevan yhteisiä ominaisuuksia. Bill Wake (2003) on antanut näille ominaisuuksille kirjainlyhenteen INVEST, joka auttaa muistuttamaan mistä ominaisuuksista hyvät käyttäjätarinat koostuvat. Kirjainlyhenne muodostuu englanninkielisten sanojen alkukirjaimista ja näitä sanoja ovat

Independent, Negotiable, Valuable, Estimable, Small, Testable. (Cohn 2009, luku 2; Agile Alliance 2021d.) Ominaisuuksia on avattu seuraavissa kappaleissa hieman tarkemmin.

Ensimmäisenä muistettavana ominaisuutena on se, että käyttäjätarinoiden tulisi olla itsenäisiä (engl. independent) eli toisistaan riippumattomia. Riippumattomuus ei aina ole mahdollista, mutta siihen tulisi pyrkiä. Se helpottaa työskentelyä ja antaa mahdollisuuden toteuttaa käyttäjätarinat missä järjestyksessä tahansa. Käyttäjätarinoiden välinen riippuvuus johtaa helposti priorisointi- ja suunnitteluongelmiin. Toisena muistettavana ominaisuutena on se, että käyttäjätarinoiden tulisi olla neuvoteltavissa (engl. negotiable). Tällä viitataan siihen, että käyttäjätarinat eivät ole kirjallisia kuvauksia vaatimuksista, vaan niiden ideana on auttaa muistuttamaan kehitettävistä toiminnallisuuksista, joista tulee keskustella ennen toteutusta. (Beck & Fowler 2001, luku 11; Wake 2003; Cohn 2009, luku 2.)

Kolmantena muistettavana ominaisuutena on se, että käyttäjätarinoiden tulisi olla arvokkaita (engl. valuable). Käyttäjätarinoiden tehtävänä on kuvata ohjelmiston toiminnallisuuksia, jotka ovat merkityksellisiä ja tuottavat arvoa joko asiakkaan liiketoiminnalle tai ohjelmiston tuleville käyttäjille. Paras tapa varmistaa tämä, on pyytää asiakasta ja ohjelmiston tulevia käyttäjiä kirjoittamaan kuvaukset. Neljäntenä muistettavana ominaisuutena on se, että käyttäjätarinoiden tulisi olla arvioitavissa (engl. estimable). Arvion tulee olla tarkkuudeltaan sellainen, että kehittäjät pystyvät antamaan arvion toteutukseen kuluva ajasta. Käyttäjätarinoiden arvioitavuus liittyy vahvasti niiden kokoon sekä siihen, että ne ovat kaikkien osapuolten näkökulmasta ymmärrettäviä. (Beck & Fowler 2001, luku 11; Wake 2003; Cohn 2009, luku 2.)

Viidentenä muistettavana ominaisuutena on se, että käyttäjätarinoiden tulisi olla pieniä (engl. small). Kuvaus sisältää lauseen tai maksimissaan kaksi – mitä lyhyempi, sen parempi. Mikäli kuvaus todetaan liian suureksi, se voidaan jakaa pienempiin osiin. (Beck & Fowler 2001, luku 11; Wake 2003; Cohn 2009, luku 2.) Liitossanojen kuten "ja" sekä "tai" etsiminen on yksi hyvä tapa etsiä pilkottavia kohteita (Riippi 2018). Koon tulisi olla myös sellainen, että käyttäjätarinan toteutukseen kuluva aika on mahdollista arvioida. Tyypillisesti käyttäjätarinan toteutukseen kuluva aika on muutamasta päivästä viikkoon. Käyttäjätarinan koon tulee olla sellainen, että se voidaan toteuttaa yhden iteraation aikana. Kuudentena muistettavana ominaisuutena on se, että käyttäjätarinoiden tulisi olla testattavissa (engl. testable). Testattavuus liittyy vahvasti siihen, että käyttäjätarina itsessään on tarpeeksi selkeä ja sen arvo on asiakkaan tai ohjelmiston tulevien käyttäjien näkökulmasta tarpeeksi merkityksellinen. Käyttäjätarina voidaan testata hyväksymistestien avulla ja se voidaan

vahvistaa toteutetuksi, kun se testatessa täyttää sille määritetyt kriteerit. Kriteerit auttavat myös tarkentamaan kuvausta ja näin ollen helpottavat työmäärän arviointia. (Beck & Fowler 2001, luku 11; Wake 2003; Cohn 2009, luku 2.)

3.4 Dokumentointi

Käyttäjätarinoiden luonnollisella kielellä kirjoitetut kuvaukset tarjoavat hyvän lähtökohdan dokumentaation tekemiseen (Laine 2021). Kirjalliset kuvaukset dokumentoidaan tuotteen kehitysjonoon, joka voi sijaita esimerkiksi Atlassianin kehittämässä Jira Software -työkalussa. Jira Software on ohjelmistoprojekteissa yleisesti käytetty projekti- ja tehtävähallintaohjelmisto. Käyttäjätarinoita on mahdollisuus lisätä projekti- ja tehtävähallintaohjelmistoon yksitellen, mutta niitä voidaan myös tarvittaessa ladata CSV-tiedostona useampia kerralla. (Patton & Economy 2014, luku 8; Atlassian 2021b.)

Käyttäjätarinoiden kirjalliset kuvaukset tarkentuvat keskusteluiden myötä ja keskusteluita onkin suositeltavaa täydentää erilaisilla muistiinpanoilla. Kirjallisen kuvauksen yhteyteen voidaan kirjoittaa hyväksymistesteihin liittyvät kriteerit, mutta lisäksi Jira Software -työkaluun on mahdollista yhdistää Confluence. Confluence on niin ikään Atlassianin kehittämä työkalu, joka tarjoaa ratkaisun kattavampaan dokumentaatioon. Kyseessä on wikiohjelmisto, johon on mahdollista dokumentoida esimerkiksi kuvia ja videoita sekä muita käyttäjätarinoihin liittyviä muistiinpanoja, jotka auttavat paremmin muistamaan keskusteluiden yksityiskohtia. (Patton & Economy 2014, luku 8; Atlassian 2021a.)

Dokumentoitujen käyttäjätarinoiden tarkkuustasot vaihtelevat niiden prioriteetin mukaan. Kehitysjonon alaosissa olevat matalamman prioriteetin käyttäjätarinat ovat kooltaan isompia ja epämääräisempiä, ja näin ollen ne auttavat vain lähinnä muistuttamaan mihin toiminnallisuuteen ne liittyvät. Ylempänä listalla olevat korkeamman prioriteetin käyttäjätarinat ovat tarkempia ja yksityiskohtaisempia, ja aikaisemmin esitetyt hyvien käyttäjätarinoiden ominaisuudet pätevätkin nimenomaan korkean prioriteetin käyttäjätarinoihin. Kun käyttäjätarina täyttää INVEST-kriteeristön, voidaan todeta, että se on valmis otettavaksi iteraatioon. Myöhemmin kun käyttäjätarina on iteraation aikana toteutettu ja se testatessa täyttää sille määritetyt kriteerit, voidaan se dokumentoida toteutetuksi. (Luukkainen & Ilves 2021a.)

4 KÄYTTÄJÄVAATIMUSTEN KARTOITTAMINEN JA DOKUMENTOINTI

4.1 Käyttäjävaatimusten kartoittaminen

Käyttäjävaatimuksia kartoitettiin Dagmarin Media -yksikön digimainonnan suunnittelijoilta ja tarkoituksena oli selvittää tarpeet uudelle markkinoinnin suunnittelu- ja laskutusohjelmistolle. Käyttäjäryhmäksi valittiin digimainonnan suunnittelijat, koska käyttäjävaatimuksia haluttiin tarkastella mediasuunnittelun ja -oston näkökulmasta, ja muihin Media -yksikön mediaryhmiin, eli printti-, ulko-, elokuva-, televisio- ja radiomainontaan, verrattuna digimainonnan suunnittelu- ja ostoprosessi on monivaiheisin ja näin ollen myös markkinoinnin suunnittelu- ja laskutusohjelmiston kannalta monimutkaisin. Käyttäjäryhmän rajausta tehtiin myös siitä syystä, ettei opinnäytetyö laajenisi kohtuuttomasti. Tästä johtuen Media -yksikön muiden mediaryhmien suunnittelijoiden lisäksi muut Dagmarin yksiköt ja muut sidosryhmät rajattiin opinnäytetyön ulkopuolelle.

Osaltaan käyttäjäryhmän valintaan vaikutti myös se, että nykyisen MediaPro -ohjelmiston kehitystä ei ole pystytty digimainonnan suunnittelijoiden näkökulmasta viemään niin pitkälle kuin muiden mediaryhmien osalta. MediaPro on otettu käyttöön ennen digimainonnan aikakautta ja vaikka sitä on vuosien saatossa kehitetty, on se alun perin luotu palvelemaan perinteisiä medioita, joissa muutokset esimerkiksi kanavissa ja ostotavoissa eivät ole olleet niin nopeita kuin digimainonnassa. Digimainonnan kanavia on valtavasti ja niitä syntyy koko ajan lisää, myös toimintavoissa tapahtuu nopeita muutoksia. Alla on kuvattu lyhyesti mitä digimainonta yleisellä tasolla tarkoittaa ja mikä sen osuus on koko mediamainonnasta niin Suomen kuin Euroopan tasolla.

4.1.1 Digimainonta

Digimainonta on digitaalisen markkinoinnin osa-alue, jolla tarkoitetaan mediamainontaa eli kaikkea maksettua näkyvyyttä, joka tapahtuu digitaalisissa kanavissa. Siihen voidaan katsoa kuuluvaksi muun muassa display- ja videomainonta, hakusanamainonta sekä mainonta sosiaalisessa mediassa, kuten esimerkiksi Facebookissa, Instagramissa, YouTubessa, Twitterissä tai LinkedInissa. Digitaalisissa kanavissa asiakaskokemusta voidaan rakentaa systemaattisesti, ja digimainonnan

avulla oikea kohderyhmä voidaan tavoittaa oikeassa paikassa ja oikealla hetkellä kaikissa ostopolun vaiheissa. Osana muita markkinointitoimenpiteitä digimainonnan tavoitteena on toimia tunnettuuden ja myynnin edistäjänä, joko välillisesti tai välittömästi.

Brändillisellä digimainonnalla pyritään ensisijaisesti rakentamaan mielikuvaa, tunnettuutta ja tietoisuutta. Taktisella digimainonnalla puolestaan pyritään saamaan aikaan toimintaa, kuten esimerkiksi tekemään ostos asiakkaan verkkokaupassa tai jättämään tarjouspyyntö asiakkaan verkkosivustolla. Digimainontaa pidetään yleisesti kustannustehokkaana mediamainonnan muotona, koska mainontaa voidaan kohdentaa tarkasti halutulle kohderyhmälle. Mainonnan toimivuutta voidaan digitaalisissa kanavissa mitata hyvinkin tarkalla tasolla ja mainontaa voidaan myös optimoida reaaliajassa. Digimainonnan kehitys on ollut viimeisten vuosien aikana huimaa, myös kanavat ja alustat ovat lisääntyneet eksponentiaalisesti. Teknologisen kehityksen ja digitalisaation mukanaan tuoma mediakulutuksen muutos näkyy osaltaan myös digimainonnan euromääräisissä panostuksissa, joiden osuus kaikesta mediamainonnasta on kasvanut viimeisten viidentoista vuoden aikana tasaiseen tahtiin, edes maailmanlaajuinen koronapandemia ei ole pysäyttänyt kehitystä.

IAB (Interactive Advertising Bureau) Europen julkaiseman vuotuisen Adex -raportin mukaan Euroopassa kaikesta mediamainonnasta digimainontaa vuonna 2020 oli 56,5 prosenttia (IAB Europe 2021; IAB Finland 2021a). Suomessa vastaava luku vuonna 2020 oli IAB Finlandin (2021b) ja Kantarin (2021a) mukaan 47 prosenttia, tarkoittaen sitä, että digimainontaa oli suurin osa kaikesta mediamainonnasta. Viimeisimpien lukujen mukaan vuoden 2021 ensimmäisellä kvartaalilla Suomen mediamainonnasta digimainontaa oli ennätysellisesti jo peräti 53 prosenttia (IAB Finland 2021c; Kantar 2021b). Digimainonnan kasvun ennustetaan edelleen jatkuvan digitaalisen mediakulutuksen kasvun sekä yritysten liiketoiminnan digitalisoitumisen myötä, jota koronapandemia on entisestään kiihdyttänyt. Vuodelle 2021 IAB Europe ennustaa kasvun olevan Euroopassa lähes 10 prosentin luokkaa. (IAB Europe 2021; IAB Finland 2021a; IAB Finland 2021c.)

4.1.2 Digisuunnittelijat

Digimainonnan suunnittelijoilla, tai lyhyemmin digisuunnittelijoilla, tarkoitetaan tässä yhteydessä Dagmarin työntekijöitä, joiden työnkuvaan kuuluu suunnitella ja toteuttaa asiakkaiden mainontaan liittyviä toimenpiteitä digitaalisissa kanavissa. Asiakkailta tarkoitetaan tässä yhteydessä mainostajia

eli Dagmarin asiakkaita, joita ovat eri kokoiset ja eri toimialojen yritykset – niin kotimaiset kuin kansainvälisetkin. Mediasuunnittelun näkökulmasta digisuunnittelijan tehtävänä on löytää sellaiset kanavat ja mediat, joissa määritelty kohderyhmä tavoitetaan tehokkaimmin tavoitteeseen nähden. Mediasuunnittelun lisäksi digisuunnittelija vastaa muun muassa mainonnan toteutuksesta, optimoinnista ja raportoinnista.

Digisuunnittelija huolehtii nimetyistä asiakkaista ja toimii osana asiakkuustiimiä, joka on kasattu asiakkaan ympärille vastaamaan kunkin asiakkaan tarpeita. Asiakkaan puolella on usein markkinointipäällikkö tai muu markkinoinnista vastaava henkilö, jonka kanssa digisuunnittelija tekee yhteistyötä. Digisuunnittelija tekee yhteistyötä usein myös asiakkaiden muiden yhteistyökumppaneiden, kuten esimerkiksi mainostoimistojen kanssa. Lisäksi yhteistyötä tehdään muun muassa eri medioiden, mediatalojen ja teknologiakumppaneiden sekä tietysti Dagmarin omien asiantuntijoiden kanssa, joita ovat muun muassa data-analyytikot, tutkijat, sisällöntuottajat, tekniset asiantuntijat sekä muiden mediaryhmien suunnittelijat.

4.1.3 Kartoituserämenetelmät

Kartoituserämenetelmäksi valittiin tietoperustan pohjalta haastattelut, jotka toteutettiin yksilöhaastatteluina teemahaastatteluiden muodossa, koska tarkoituksena oli ensisijaisesti saada aikaan vuoropuhelu haastateltavien digisuunnittelijoiden kanssa ja syventyä ymmärtämään heitä. Tavoitteena oli hankkia syvälinen ymmärrys haastateltavien digisuunnittelijoiden tarpeista ja arjen haasteista markkinoinnin suunnittelu- ja laskutusohjelmiston käyttöön ja käytettävyyteen liittyen. Lisäksi haastatteluissa pyrittiin pureutumaan siihen, kuinka digisuunnittelijoiden näkökulmasta päällekkäisiä työvaiheita poistamalla voitaisiin sujuvoittaa päivittäistä työtä ja koko suunnittelu- ja ostoprosessia.

Ennen haastatteluja tehtiin valmisteluja laatimalla muun muassa kysymyspatteriston sisältävä kirjallinen haastattelurunko (LIITE 1), jota käytettiin haastattelun tukena. Haastateltavien henkilöiden kanssa sovittiin haastatteluajat ja haastattelut toteutettiin sovittuina ajankohtina. Vallitsevan epidemiatilanteen ja siitä johtuvan toistaiseksi voimassa olevan valtakunnallisen etätyösuosituksen johdosta haastattelut toteutettiin videopuhelun välityksellä. Työkaluna käytettiin Teamsia, koska se on Dagmarilla yleisesti käytössä oleva työkalu. Esittelyihin ei haastatteluiden alkuvaiheessa ollut tarvetta, koska haastattelija ja haastateltavat olivat toisilleen entuudestaan tuttuja. Tästä johtuen myös

ilmapiiriä voisi luonnehtia rennoksi. Alkuun haastateltaville kerrottiin yleisesti opinnäytetyön tavoitteet, ja täsmennettiin vielä mihin haastattelu liittyy ja mikä sen tarkoitus on. Samassa yhteydessä myös korostettiin vastausten merkityksellisyyttä ja luottamuksellisuutta.

Haastatteluiden teemana oli markkinoinnin suunnittelu- ja laskutusohjelmiston käyttö ja käytettävyys, joten ennalta laadittujen kysymysten avulla pyrittiin selvittämään haastateltavilta henkilöiltä muun muassa nykyisen MediaPro -ohjelmiston käyttötilanteita ja käytettävyyttä sekä näihin liittyviä ongelmakohtia. Potentiaalisten automaatiokohteiden näkökulmasta pyrittiin selvittämään muun muassa se, että mihin työtehtäviin haastateltavilta kuluu eniten työaika ja mitkä ovat suunnittelu- ja ostoprosessin pullonkauloja. Haastatteluissa esitettiin avoimia kysymyksiä, joihin ei ollut valmiita vastausvaihtoehtoja. Vastaukset kirjattiin vapaamuotoisesti muistiin ja niihin esitettiin haastatteluiden aikana tarkentavia jatkokysymyksiä, jotka veivät haastatteluita keskustelunomaisesti eteenpäin.

Haastatteluiden aikana haastateltavien kanssa käytiin läpi myös Dagmarin sisäistä digikampanjan suunnitteluprosessiin liittyvää materiaalia, jonka avulla keskustelua pyrittiin viemään syvemmälle tasolle. Lisäksi haastatteluihin yhdistettiin käyttötilanteiden havainnointia, joka tapahtui haastattelun yhteydessä niin, että haastateltavia pyydettiin jakamaan Teamsin välityksellä oman tietokoneensa näyttöruutu ja näyttämään haastattelussa ilmenneitä käyttöön liittyviä yksityiskohtia nykyisellä MediaPro -ohjelmistolla. Haastateltavien kanssa keskusteltiin myös nykyisen markkinoinnin suunnittelu- ja laskutusohjelmiston käytettävyydestä pääasiassa tuloksellisuuden, tehokkuuden ja tyytyväisyyden näkökulmasta.

Haastatteluihin oli varattu aikaa reilusti, puolitoista tuntia, mutta kysymyksiä oli listattu kysymyspatteristoon sen verran paljon, että kaikkiin kysymyksiin ei ehditty haastatteluiden aikana paneutua niin syvällisesti kuin oli alun perin tarkoitus. Haastattelut etenivät hyvin vapaamuotoisesti ja kysymysjärjestys myös hieman vaihteli haastattelusta riippuen. Haastatteluiden aikana haastateltavat pääsivät vapaasti ilmaisemaan omia ajatuksiaan ja mielipiteitään markkinoinnin suunnittelu- ja laskutusohjelmiston käyttöön ja käytettävyyteen liittyen. Haastatteluiden päätteeksi varmistettiin, että haluaako haastateltava vielä lisätä tai kysyä jotakin. Lisäksi haastateltaville kerrottiin kuinka opinnäytetyöhön liittyvä prosessi haastattelujen jälkeen etenee. Haastattelut päätettiin kiitoksiin.

4.2 Käyttäjävaatimusten dokumentointi

Käyttäjävaatimusten dokumentointi lähti liikkeelle jo haastatteluiden aikana tehdyistä vapaamuotoisista muistiinpanoista, joiden pohjalta saatiin hahmotettua kokonaiskuva nykyisestä suunnittelu- ja ostoprosessista sekä markkinoinnin suunnittelu- ja laskutusohjelmiston roolista osana nykyistä prosessia. Haastatteluiden aikana tehtyjen muistiinpanojen sekä Dagmarin sisäisen digikampanjan suunnitteluprosessiin liittyvän materiaalin pohjalta nykyinen suunnittelu- ja ostoprosessi kuvattiin digisuunnittelijan näkökulmasta kirjallisesti.

Kirjallista kuvausta tukemaan luotiin myös visuaalinen kuvaus markkinoinnin suunnittelu- ja laskutusohjelmiston roolista osana nykyistä suunnittelu- ja ostoprosessia. Visuaalinen kuvaus auttoi tunnistamaan markkinoinnin suunnittelu- ja laskutusohjelmiston käyttöön liittyvät korkean tason käyttäjävaatimukset, jotka kuvattiin ja dokumentoitiin käyttäjätarinoina. Suunnittelu- ja ostoprosessiin sekä markkinoinnin suunnittelu- ja laskutusohjelmiston käyttöön ja käytettävyyteen liittyen tunnistettiin myös kehityskohteita, jotka nousivat esiin haastatteluiden aikana.

4.2.1 Suunnittelu- ja ostoprosessi

Suunnittelu- ja ostoprosessia voidaan tarkastella monesta eri näkökulmasta, mutta tässä kontekstissa sitä tarkastellaan digimainontaan liittyvän yksittäisen mainoskampanjan näkökulmasta. Suunnittelu- ja ostoprosessiin voidaan näin ollen katsoa kuuluvaksi kaikki mainoskampanjan suunnitteluun ja -ostoon liittyvät toimenpiteet, kuten asiakkaan tavoitteisiin ja olemassa olevaan dataan pohjautuva suunnittelutyö sekä suunnittelutyön pohjalta tapahtuva mainoskampanjan toteutus, optimointi, raportointi ja laskutus. Mainoskampanjalla voidaan tarkoittaa joko brändillistä mainontaa tai taktista mainontaa.

Suunnittelu- ja ostoprosessi on monivaiheinen, ja se sisältää useita hallinnollisia ja operatiivisia työtehtäviä, joita hoidetaan usein limittäin. Olennaista on huomioida, että prosessi sisältää paljon muuttujia, ja yksityiskohdat vaihtelevat usein tilanteesta ja hoidettavasta asiakkuudesta riippuen. Karkeasti prosessi voidaan kuitenkin jakaa kolmeen eri vaiheeseen, joita ovat suunnitteluvaihe, rakennusvaihe ja kampanjavaihe. Suunnittelu- ja ostoprosessin eri vaiheita on avattu yleisellä tasolla seuraavissa kappaleissa hieman tarkemmin ja pääasialliset kosketuspisteet markkinoinnin

suunnittelu- ja laskutusohjelmiston kanssa on kuvattu visuaalisesti alla olevassa kuvassa (KUVIO 6).



KUVIO 6. Suunnittelu- ja ostoprosessin sisältämät pääasialliset kosketuspisteet markkinoinnin suunnittelu- ja laskutusohjelmiston kanssa.

Suunnitteluvaihe lähtee liikkeelle briefistä eli asiakkaan toimeksiannosta. Kun intermediatason rooli eri mediaryhmien välillä on asiakasvastaavan johdolla tehty, ja eri mediaryhmien tavoitteet ja roolit ovat mainoskampanjaan liittyen selvillä, on digisuunnittelijan tehtävänä hoitaa intramediataason suunnittelu. Eli toisin sanoen, digisuunnittelijan tehtävänä on suunnitella asiakkaan tavoitteita ja valittua strategiaa vastaavat kanavatason mediavalinnat. Digisuunnittelijan näkökulmasta suunnittelutyö sisältää muun muassa mediavalintojen ja mainosformaattien suunnittelun sekä hintaneuvottelut medioiden kanssa. Tavoitteiden saavuttamista voidaan digimainonnan osalta mitata hyvinkin tarkalla tasolla, joten oikeiden mittareiden määrittely on myös oleellinen osa suunnittelutyötä.

Suunnittelutyön pohjalta tehdään mediasuunnitelma, joka sisältää mainoskampanjaan valitut mediat ja mainosmuodot sekä muut oleelliset tiedot, kuten esimerkiksi aikataulut, näyttömäärät ja budjetit. Mediasuunnitelma tehdään yleensä aluksi Exceliin, jossa sitä on helppo käsitellä ja muokata. Tässä yhteydessä kasataan usein myös Power Point -esitys, jossa mediasuunnitelmaa on avattu tarkemmin ja visuaalisemmin. Excelin pohjalta mediasuunnitelma syötetään [1] markkinoinnin suunnittelu- ja laskutusohjelmistoon viimeistään siinä vaiheessa, kun asiakas on kirjallisesti hyväk-

synyt mediasuunnitelman ja sen edistämiseksi on saatu lupa. Joissain tapauksissa mediasuunnitelma voidaan syöttää suoraan markkinoinnin suunnittelu- ja laskutusohjelmistoon, jolloin Excel-välivaihe jää kokonaan pois.

Kun mediasuunnitelma on syötetty, joko Excel-välivaiheen kautta tai ilman sitä, voidaan valmis julkaisusuunnitelma ladata [2] markkinoinnin suunnittelu- ja laskutusohjelmistosta. Tämä tarkoittaa sitä, että valmis julkaisusuunnitelma ladataan ulos PDF-tiedostona ja toimitetaan sähköpostilla tai muulla sovitulla tavalla asiakkaalle hyväksyttäväksi. Mikäli asiakas on jo hyväksynyt mediasuunnitelman aikaisemmin muulla tavalla, niin tämä vaihe saattaa joissain tapauksissa jäädä välistä. Julkaisusuunnitelma vastaa asiakkaalle myöhemmin toimitettavaa laskua ja se sisältää mediasuunnitelman rivit, jotka siirtyvät asiakkaan myyntilaskulle siinä vaiheessa, kun digisuunnittelija käy siirtämässä ne tilauskantaan.

Asiakkaalta saadun kirjallisen hyväksynnän jälkeen mediasuunnitelmaa voidaan lähteä konkreettisesti edistämään. Mikäli mediasuunnitelma sisältää suoraostoja, voidaan tässä vaiheessa toimittaa medioille määräimet [3] eli vahvistus mainoskampanjasta. Tämä tarkoittaa sitä, että digisuunnittelija lataa markkinoinnin suunnittelu- ja laskutusohjelmistosta PDF-tiedostot ja lähettää ne sähköpostilla medioiden yhteyshenkilöille. Jokainen vahvistus on erillinen tiedosto, joka sisältää uniikin määräysnumeron. Määräysnumero toimii medioiden suuntaan laskutuslupana sovitulle summalle ja sitä tarvitaan myöhemmin medialta tulevan ostolaskun kohdistamisessa. Asiakkaan kirjallisen hyväksynnän jälkeen on myös mahdollista suorittaa ennakkolaskutus [4].

Rakennusvaihe tarkoittaa mediasuunnitelman toteuttamista käytännössä ja tyypillisesti rakennusvaihe lähtee liikkeelle teknisten ohjeiden laatimisesta. Tekniset ohjeet [5] laaditaan mediasuunnitelman pohjalta ja ne toimitetaan asiakkaan mainostoimistolle tai muulle taholle, joka tuottaa mainoskampanjan aineistot. Teknisillä ohjeilla tarkoitetaan dokumenttia, josta ilmenee kampanjaan suunniteltujen medioiden aineistotarpeet sekä niiden tekniset vaatimukset. Tekniset ohjeet laaditaan asiakkuudesta riippuen joko digisuunnittelijan tai teknisen suunnittelijan toimesta. Kun tekniset ohjeet on toimitettu eteenpäin, voidaan rakennusvaihetta jatkaa vaikka aineistoja ei vielä olekaan. Mikäli asiakkaalla on käytössä nimeämistyökalu, on sen täyttäminen seuraava työvaihe. Nimeämistyökalu [6] täytetään mediasuunnitelman pohjalta.

Kun aineistot ennen varsinaista kampanjavaihetta saapuvat, otetaan ne työn alle asiakkuudesta riippuen joko digisuunnittelijan tai teknisen suunnittelijan toimesta. Kampanjan trafikointi [7] tarkoittaa muun muassa aineistojen tarkistusta ja testausta sekä mediasuunnitelman pohjalta tapahtuvaa mainoskampanjan rakennusta asiakkaalla käytössä olevaan mainonnanhallintajärjestelmään. Mikäli mediasuunnitelma sisältää suoraostoja, toimitetaan aineistot ja tarvittavat seurannat [8] mediasuunnitelman mukaisesti medioille. Mediasuunnitelman sisältäessä mainontaa esimerkiksi sosiaalisessa mediassa, tulee mainoskampanja rakentaa mediasuunnitelman ja tavoitteiden mukaisesti eri ostojärjestelmiin [9]. Ennen varsinaista kampanjavaihetta tulee viimeistään myös varmistaa, että tarvittavien sivustoseurantojen osalta on kaikki kunnossa.

Kampanjavaihe on vaihe, jolloin mainonta on varsinaisesti käynnissä eli se näkyy eri kanavissa ja medioissa. Mainonnanhallinta- ja ostojärjestelmistä varmistetaan, että mainoskampanja on varmasti käynnistynyt. Kampanjavaiheessa mainonnanhallinta- ja ostojärjestelmissä myös seurataan sitä, että kampanja pyörii mediasuunnitelman ja tavoitteiden mukaisesti. Tarvittaessa reagoidaan, ja ollaan myös esimerkiksi yhteydessä medioihin tai asiakkaaseen. Kampanjasta riippuen asiakkaalle saatetaan toimittaa puolivälissä väliraportti, mutta kampanjan päätyttyä tulokset raportoidaan kattavammin. Kampanjavaiheessa hoidetaan viimeistään kampanjan laskutukseen [10] liittyvät asiat markkinoinnin suunnittelu- ja laskutusohjelmiston avulla. Toteuman mukaan laskutettavat toimenpiteet [11] laskutetaan vasta kampanjan päätyttyä.

4.2.2 Käyttäjätarinoiden kuvaukset

Käyttäjätarinoiden kuvaukset kirjoitettiin tunnistettujen tarpeiden ja selvitettyjen käyttäjävaatimusten pohjalta. Apuna käytettiin mallia, joka auttoi muotoilemaan kuvauksen niin, että se kiteyttää tarpeen, kertoen lukijalle mihin digisuunnittelija kyseistä toiminnallisuutta tarvitsee ja miksi. Kuvauksissa pyrittiin huomioimaan, että toiminnallisuudet ovat digisuunnittelijoiden näkökulmasta merkityksellisiä ja kirjoittamisessa pyrittiin huomioimaan erityisesti se, että kuvaus ei sisällä vielä tässä vaiheessa liikaa yksityiskohtia, koska niistä on tarkoitus keskustella vasta myöhemmässä vaiheessa.

Kirjoittamisen apuna pyrittiin hyödyntämään myös INVEST-kriteeristöä, joka auttoi muistuttamaan mistä ominaisuuksista hyvät käyttäjätarinat koostuvat. Kaikkia hyvän käyttäjätarinan ominaisuuksia

ei kuitenkaan vielä tässä vaiheessa pystytty ottamaan huomioon, koska kuvaukset kirjoitettiin nykyiseen suunnittelu- ja ostoprosessiin pohjautuen korkealla tasolla. Näin ollen kuvauksista voidaan vielä tässä vaiheessa käyttää sanaa epiikki. Kirjotetut kuvaukset eivät välttämättä kokonsa puolesta ole liian suuria, mutta ne ovat vielä toistaiseksi liian epätarkkoja muun muassa testattavuudeltaan sekä työmääräarvioiltaan. Käyttäjätarinoiden kuvauksia (LIITE 2) syntyi yhteensä neljätoista ja ne dokumentointiin Exceliin, joka toimitetaan toimeksiantajalle. Excelissä kuvauksia voidaan tarvittaessa myöhemmin helposti käsitellä ja muokata. Excel mahdollistaa myös sen, että tiedosto voidaan tarvittaessa tallentaa luetteloerottimen sisältäväksi CSV-tiedostoksi, joka voidaan ladata suoraan esimerkiksi Jira Software -työkaluun.

Kirjoitettujen kuvausten osalta voidaan todeta, että ne eivät ole tarkkoja käyttäjävaatimuksia. Niitä voidaan kuitenkin käyttää ohjelmistoprojektin alkuvaiheessa suunnittelutyön apuna, ja projektin edetessä niitä voidaan tarkentaa. Kirjalliset kuvaukset toimivat lähtökohtana dokumentaatiolle ja ne auttavat muistuttamaan toiminnallisuuksista, joista tulee keskustella projektin aikana tarkemmin digisuunnittelijoiden sekä muiden käyttäjäryhmien kanssa, joita toiminnallisuudet koskevat. Keskusteluiden avulla on mahdollista päästä kiinni todelliseen tarpeeseen ja keskusteluiden avulla voidaan myös saavuttaa yhteinen ymmärrys toteutettavista toiminnallisuuksista. Yhteisen ymmärryksen pohjalta voidaan puolestaan löytää paras mahdollinen ratkaisu toiminnallisuuksien toteuttamiseen.

4.2.3 Tunnistetut kehityskohteet

Tunnistetut kehityskohteet liittyivät suunnittelu- ja ostoprosessin sekä markkinoinnin suunnittelu- ja laskutusohjelmiston osalta pääasiassa siihen, että prosessi sisältää paljon manuaalisia työvaiheita eikä nykyinen MediaPro -ohjelmisto juurikaan tue prosessia, vaan enemmänkin ohjaa sitä. Markkinoinnin suunnittelu- ja laskutusohjelmiston rooli osana prosessia on hyvin merkittävä, mutta digisuunnittelijoiden keskuudessa nykyistä MediaPro -ohjelmistoa pidetään kuitenkin lähinnä pelkkänä laskutustyökaluna. Tämä johtuu pääasiassa siitä, että osana nykyistä prosessia MediaPro -ohjelmisto astuu kuvaan vasta suunnitteluvaiheen loppupuolella, tarkemmin sanottuna siinä vaiheessa, kun varsinainen suunnittelutyö on jo tehty ja mediasuunnitelma on jo valmiina. Suunnitteluvaihe ja erityisesti suunnitteluvaiheen alkuvaihe on digisuunnittelijan näkökulmasta koko suunnittelu- ja ostoprosessin kannalta usein eniten aikaa vievin vaihe, joten varsinkin siihen kaivattaisiin enemmän tukea uudelta markkinoinnin suunnittelu- ja laskutusohjelmistolta.

Markkinoinnin suunnittelu- ja laskutusohjelmiston osalta kehityskohteet liittyivät myös vahvasti siihen, että toimintaympäristö ja tarpeet ovat vuosien saatossa muuttuneet. Nykyinen MediaPro -ohjelmisto on suunniteltu alun perin perinteisten medioiden, kuten printti- ja televisiomainonnan tarpeisiin ja se on luotu aikaan, jolloin ennalta laskutettuihin mainoskampanjoihin tuli harvoin muutoksia. Näin ollen esimerkiksi toteuman perusteella jälkikäteen laskutettavan jatkuvan digimainonnan tarpeisiin nykyinen ohjelmisto ei kovin hyvin sovellu. Tässä yhteydessä kehityskohteena nousivat esiin korjauslaskut ja niihin liittyvät käytännöt, joihin digisuunnittelijoilta kuluu toisinaan tuhotomasti aikaa. Tätä pidettiin myös digisuunnittelijoiden keskuudessa haasteena, joka toistuu kerasta toiseen. Vaikka MediaPro -ohjelmistoa on vuosien saatossa kehitetty, niin kaikkia digisuunnittelijoiden tarpeita ei ole pystytty huomioimaan tai ratkaisemaan. Digisuunnittelijat ovatkin joutuneet sopeuttamaan oman toimintansa nykyisen ohjelmiston asettamiin raameihin.

Digisuunnittelijoiden keskuudessa nykyistä markkinoinnin suunnittelu- ja laskutusohjelmistoa pidetään yleisesti toimintavarmana, koska käyttökatkoja ei ole, tai niitä on todella harvoin. Käyttö koetaan kuitenkin epämiellyttäväksi, koska käyttöliittymä itsessään on vanhanaikainen ja ohjelmiston käyttö on muutenkin hidasta. Mikäli digisuunnittelija siirtyy toimistolla esimerkiksi palaveriin ja irrottaa tietokoneen telakasta, niin MediaPro -ohjelmisto lakkaa toimimasta. Ohjelmiston käyttö siis keskeytyy siksi aikaa, kunnes ohjelmiston ja sen antamat virheilmoitukset sulkee ja ohjelmiston avaa uudestaan. MediaPro muodostuu eri moduuleista, joiden välillä operoiminen koetaan yleisesti turhauttavaksi ja tässä yhteydessä kehityskohteina esiin nousivat jälleen korjauslaskut, jotka tulee tehdä erillisessä moduulissa. Hämmennystä aiheuttaa aika ajoin myös eri medioiden monikanavaiset ratkaisut, jotka liittyvät sekä printtimainontaan että digimainontaan, joten myös tämän osalta kaivattaisiin ratkaisua uudelta markkinoinnin suunnittelu- ja laskutusohjelmistolta.

Perustoimintojen osalta nykyistä MediaPro -ohjelmistoa pidetään suhteellisen helppokäyttöisenä sellaisten digisuunnittelijoiden keskuudessa, jotka ovat olleet Dagmarin palveluksessa pidemmän aikaa. MediaPro sisältää kuitenkin paljon sellaisia toiminnallisuuksia, joista kokeneemmillakaan digisuunnittelijoilla ei ole mitään käsitystä. Uudempana digisuunnittelijana ohjelmiston käyttö vaatii alkuun kattavamman perehdytyksen ja tässä yhteydessä esiin nousi huomio siitä, että ohjelmisto ei selitä itseään millään tavalla. Lisäksi käyttöliittymän kuvakkeet koetaan digisuunnittelijoiden keskuudessa osittain hyvinkin epäselviksi ja kehityskohteina mainittiin myös epäselvä fontti sekä fontin koko, jota ei ole mahdollista nykyisessä MediaPro -ohjelmistossa muuttaa. Kehityskohteena mark-

kinoinnin suunnittelu- ja laskutusohjelmiston osalta mainittakoon vielä se, että tilauskantaan siirretyt rivit on toteutettu värikoodein, joten käytettävyyden lisäksi myös saavutettavuuden näkökulmasta todettiin olevan kehitettävää.

Suunnittelu- ja ostoprosessin osalta kehityskohteet liittyivät muun muassa siihen, että prosessin aikana samaa tietoa on monessa eri paikassa, ja tietoa joudutaan siirtämään muun muassa Excelistä manuaalisesti markkinoinnin suunnittelu- ja laskutusohjelmistoon ja tämän jälkeen ainakin osittain samoja tietoja joudutaan siirtämään vielä markkinoinnin suunnittelu- ja laskutusohjelmistosta esimerkiksi mainonnanhallintajärjestelmään. Sen lisäksi, että manuaalinen työ on hidasta, se myös lisää inhimillisten virheiden määrää. Digisuunnittelijoiden työ on pääasiassa hyvin hektistä ja työn alla saattaa olla useita eri kampanjoita, joten suunnittelu- ja ostoprosessin aikana tulevat muutostarpeet saattavatkin inhimillisistä syistä jäädä joskus päivittämättä ja tästä johtuen tiedot eivät toisinaan ole kaikkialla ajan tasalla.

Suunnittelu- ja ostoprosessi sisältää myös useita liikkuvia osia, jotka kaikki vaikuttavat kaikkeen ja näin ollen voidaan todeta, että prosessin eri työtehtävät eivät siis ole toisistaan irrallisia. Digisuunnittelijoiden työtehtävien ja koko prosessin sujuvuuden kannalta oleelliseksi tekijäksi nostettiin asiakkaalta saatu toimeksianto, koska se vaikuttaa kaikkiin prosessin vaiheisiin ja työtehtäviin. Hyvä toimeksianto helpottaa ja nopeuttaa digisuunnittelijan työtä, kun taas puutteellinen toimeksianto vaikeuttaa työtä ja aiheuttaa todennäköisesti haasteita myöhemmässä vaiheessa. Dagmarilta löytyy jo valmiina talotason brief -pohja, jota toimeksiannoissa voidaan hyödyntää, mutta kehityskohteiden näkökulmasta esiin nostettiin kyseisten tietojen saaminen ja hyödyntäminen uudessa markkinoinnin suunnittelu- ja laskutusohjelmistossa.

Potentiaalisten automaatiokohteiden näkökulmasta voidaan todeta, että suunnittelu- ja ostoprosessin työtehtäviä on haastavaa kuvata vaihe vaiheelta alusta loppuun saakka, koska prosessi sisältää paljon muuttujia, ja yksityiskohdat vaihtelevat usein tilanteesta ja hoidettavasta asiakkuudesta riippuen. Lisäksi eri työtehtävät vaativat osittain erityistä tietotaitoa sekä toisinaan myös monimutkaisempaa päättelykykyä. Nykyisessä suunnittelu- ja ostoprosessissa on kuitenkin selvästi nähtävillä työtehtäviä, joiden osalta manuaalista työtä voitaisiin vähentää, jotta aikaa säästyisi aidosti lisäarvoa tuottavampaan työhön. Prosessien uudistaminen ja nykyisen työn tehostaminen vaatiikin rinnalleen uuden markkinoinnin suunnittelu- ja laskutusohjelmiston, joka tukee koko prosessia.

5 POHDINTA

Opinnäytetyön tarkoituksena oli tehdä pohjatyötä ennen Dagmarin ohjelmistoprojektia, ja tutkia käyttäjätarinoita keinona kuvata ja dokumentoida käyttäjävaatimuksia. Tämä edellytti perehtymistä ohjelmistotuotantoon sekä siihen liittyviin käytäntöihin, joten tietoperustassa avattiin alkuun ohjelmistotuotannon käsitettä sekä määrittelyn ja vaatimusten roolia ohjelmistotuotannossa. Konteksti käyttäjätarinoiden käyttöön luotiin tarkastelemalla ohjelmistoprojektia ja erityisesti siihen liittyvää vaatimusmäärittelyä ketterän lähestymistavan näkökulmasta. Tietoperustan tarkoituksena oli selvittää mitä käyttäjätarinat ovat, ja kuinka niiden avulla voidaan kuvata ja dokumentoida käyttäjävaatimuksia osana ketterää ohjelmistoprojektia. Toteutusosassa tietoperusta yhdistettiin käytännön toteutukseen, ja opinnäytetyön tuloksena kartoitetut käyttäjävaatimukset kuvattiin ja dokumentointiin käyttäjätarinoina.

Tavoitteena oli lisätä ymmärrystä ohjelmiston tulevien käyttäjien tarpeista ja arjen haasteista, ja tämän opinnäytetyön osalta tarkasteltavaksi käyttäjäryhmäksi rajattiin Dagmarin Media -yksikön digimainonnan suunnittelijat. Käyttäjäryhmän rajaukseen oli perusteltuja syitä, mutta Dagmarin ohjelmistoprojektin tavoitteet huomioon ottaen, työskentelyprosesseja on tärkeää tarkastella yksittäisen käyttäjäryhmän sijaan laajemmin eri käyttäjäryhmien näkökulmasta. Mikäli käyttäjävaatimuksia tarkastellaan vain yksittäisen käyttäjäryhmän näkökulmasta, niin tavoitteena oleva suunnittelu- ja ostoprossien tehostaminen voi jäädä huomioimatta, koska edellä mainittuihin prosesseihin liittyy useita eri käyttäjäryhmiä. Eri käyttäjäryhmien näkökulmia tarvitaan myös siitä syystä, ettei yksittäisillä käyttäjillä tyypillisesti ole laajaa ymmärrystä kaikista liiketoimintaan vaikuttavista tekijöistä. Eri käyttäjäryhmiä tulisikin osallistaa ristiin esimerkiksi yhteisissä työpajoissa.

Opinnäytetyön aikana käyttäjävaatimuksia kartoitettiin haastatteluiden avulla, mutta myös muiden kartoitusmenetelmien käyttö on Dagmarin ohjelmistoprojektissa perusteltua ja suorastaan välttämätöntä. Yksilöhaastatteluiden avulla saatiin kuitenkin aikaan haluttu vuoropuhelu, joka auttoi lisäämään ymmärrystä digisuunnittelijoiden tarpeista ja arjen haasteista. Haastatteluihin yhdistetyllä käyttötilanteiden havainnoinnilla päästiin puolestaan kiinni erityisesti sellaisiin markkinoinnin suunnittelu- ja laskutusohjelmistoon liittyviin yksityiskohtiin, jotka eivät olisi pelkkien haastatteluiden aikana muuten tulleet esille. Käyttötilanteiden havainnointia voidaan pitää tärkeänä menetelmänä myös siitä syystä, että Dagmarin ohjelmistoprojektin tapauksessa kyseessä on vanhan ohjelmiston korvaaminen uudella ja näin ollen nykyisen MediaPro -ohjelmiston rooli osana projektia on hyvin

merkittävä. Oleellista haastatteluiden kannalta oli oma tuntemus toimialasta sekä aikaisempi tietämys MediaPro -ohjelmistosta. Haastavinta olikin yrittää pysyä haastatteluiden aikana mahdollisimman neutraalina, koska monet vastaukset kirvoittivat paljastamaan myös oman mielipiteen asioista.

Haastatteluiden aikana tuli todettua, että useat digisuunnittelijat olivat sopeutuneet käyttämään nykyistä markkinoinnin suunnittelu- ja laskutusohjelmistoa niin hyvin, että heidän oli jopa hieman haastavaa pohtia asioita eri näkökulmista. Ryhmähaastattelut tai työpajatyyliset innovatiiviset karitoitusmenetelmät voisivatkin sopia paremmin, kun tarkoituksena on löytää uusia näkökulmia. Toisaalta taas ryhmätyöskentely voi joskus haitata uusien ajatusten syntymistä, joten paras tapa voisikin olla se, että haastateltavat olisivat saaneet rauhassa pohtia asioita haastattelun jälkeen yksin, jonka jälkeen olisi otettu uusi haastattelukierros. Opinnäytetyön aikana toteutettuihin haastatteluihin oli varattu reilusti aikaa, mutta se loppui kesken. Kysymyksiä osoittautui olevan liikaa, joten teemaa olisi pitänyt rajata tarkemmin tai haastattelut olisi pitänyt jakaa kahteen osaan, jolloin keskustelua olisi voitu jatkaa toisella kierroksella. Tämä olisi myös mahdollistanut sen, että asioista olisi päästy keskustelemaan entistä syvemmillä tasolla.

Haastatteluiden pohjalta selvitettyjen käyttäjävaatimusten muotoileminen käyttäjätarinoiksi oli alkuun yllättävän vaikeaa, mutta lopulta kuvauksia syntyi neljätoista. Haastavinta oli löytää tarkkuustaso, jolla kuvaukset kannattaa kirjoittaa. Kirjoitetut kuvaukset perustuivat nykytilaan ja ne kuvaavat abstraktiotasolla tavoitteita, joita digisuunnittelijoiden on markkinoinnin suunnittelu- ja laskutusohjelmistolla kyettävä suorittamaan. Kuvausten osalta voitiin todeta, että niitä voidaan käyttää Dagarin ohjelmistoprojektissa suunnittelun apuna muistuttamaan toiminnallisuuksista, joista tulee keskustella tarkemmin digisuunnittelijoiden sekä muiden käyttäjäryhmien kanssa, joita toiminnallisuudet koskevat. Keskusteluiden avulla on mahdollista saavuttaa yhteinen ymmärrys siitä, mitä ollaan tekemässä ja miksi. Yhteisen ymmärryksen pohjalta voidaan puolestaan löytää paras mahdollinen ratkaisu toiminnallisuuksien toteuttamiseen.

Käyttäjätarinoista ei ollut aikaisempaa tietämystä, joten aihealueeseen perehtyminen syvensi omaa ymmärrystä ja ammatillista osaamista. Käyttäjätarinoiden ymmärtäminen vaatii kuitenkin laajempaa ymmärrystä ketterästä ohjelmistokehityksestä ja erityisesti vaatimusmäärittelystä. Vaatimusmäärittely on yksi haastavimmista osa-alueista, ja se myös aiheutti yksittäisistä asioista eniten hämmennystä, koska se tarkoittaa eri lähteissä hieman eri asioita, ainakin käytetyt termit vaihtelivat lähteestä riippuen. Opinnäytetyössä päädyin lopulta käyttämään johdonmukaisesti samaa suo-

menkielistä termistöä kuin Haikala ja Mikkonen Ohjelmistotuotannon käytännöt -kirjassaan. Käyttäjätarinoihin perehtyminen auttoi ymmärtämään, että niillä on ohjelmistoprojektin kannalta monia etuja, mutta yksinään ne eivät kuitenkaan riitä, koska onnistuakseen vaatimusmäärittely edellyttää muun muassa avointa kommunikaatiota, jonka taustalla on osapuolten välinen luottamus. Työväli-
neenä käyttäjätarinoita täytyy myös osata käyttää oikein, joten oleellista on ymmärtää niiden todellinen tarkoitus. Kuten eräs alan asiantuntijakin totesi, niin hyväkään työkalu osaamattomissa käsissä ei tuota optimaalisia tuloksia.

Opinnäytetyötä tehdessä pohdin sitä, että mikäli käyttäjätarinoiden todellista tarkoitusta ei ole ymmärretty oikein, niin ne saattavat näyttäytyä vain tietyn mallin mukaan kirjoitettuina kuvauksina. Haikalan ja Mikkosen (2011, 84) mukaan kaikkien kehitysjonossa olevien tehtävien muotoileminen käyttäjätarinoiksi voi vaatia melkoista luovuutta. Tämä antaa ikään kuin ymmärtää, että kaikki tehtävät tulisi muotoilla kehitysjonoon käyttäjätarinoiksi. Oman käsityksen ja oppiman mukaan kaikkia kehitysjonossa olevia tehtäviä ei kuitenkaan ole tarkoituksaan kirjoittaa käyttäjätarinoiden muotoon, vaan käyttäjätarinat toimivat nimenomaan tietyn tyyppisissä käyttäjien tarpeisiin perustuvissa vaatimuksissa, jotka kuvaavat ohjelmiston toimintaa tulevien käyttäjien näkökulmasta. Toisin sanoen, kaikkea ohjelmiston kehitykseen liittyvää työtä ei siis ole tarkoituksenmukaista tai mahdollista ilmaista käyttäjätarinoina, joten osa tehtävistä lisätään kehitysjonoon muun tyyppisinä kuvauksina.

Käyttäjätarinoiden kirjoittamiseen käytettävää mallia on Luukkaisen ja Ilveksen (2021a) mukaan kritisoitu muun muassa sen takia, että malli kiinnittää liikaa huomioita siihen, kuinka käyttäjätarinat kirjoitetaan. Heidän mukaansa kyseinen malli onkin menettämässä suosiotaan. Vaikka malli saattaa aiheuttaa sen, että keskittyminen menee varsinkin alkuun kokemattomalla henkilöllä liiaksi kuvauksen muotoiluun, niin mallilla on selvästi nähtävissä selkeä tarkoitus. Tarkoituksena on auttaa muotoilemaan kuvaus niin, että se kiteyttää kaiken oleellisen eli kenelle toiminnallisuutta ollaan tekemässä, mitä sen avulla halutaan saada aikaan ja miksi sitä ylipäätään ollaan tekemässä. Kirjallinen muoto ei kuitenkaan ole käyttäjätarinoiden olennaisin asia, mutta sen tehtävänä on siirtää painopiste käyttäjiin, pois teknisestä toteutuksesta. Tarkoituksena on keskittyä ohjelmiston tuleviin käyttäjiin ja asettaa heidän tarpeensa keskustelun keskipisteeseen.

Opinnäytetyö oli laaja oppimistehtävä, joka vahvisti käsitystä siitä, että monimutkaisten ohjelmistojen ja kokonaisten tietojärjestelmien kehittäminen on haastavaa työtä ja vaatii osaamista useilta eri osa-alueilta. Opinnäytetyö tehtiin päivätöiden ohessa, joten prosessin aikana suurimmat haasteet

olivat aikatauluun liittyviä ja alun perin suunniteltuun aikatauluun tulikin viivästyksiä. Dagmarin ohjelmistoprojektin aikataulu muuttui kuitenkin niin ikään prosessin aikana ja tästä johtuen opinnäytetyö valmistui juuri ajallaan. Tavoitteena oli, että opinnäytetyö voisi toimia toimeksiantajan apuna ohjelmistoprojektin valmisteluvaiheessa. Valmisteluvaihe on juuri aloitettu ja sen pohjalta tullaan tekemään tarkempi projektisuunnitelma. Opinnäytetyöprosessin aikana Dagmarin ohjelmistoprojektiin löydettiin myös kumppani, jonka kanssa yhteistyö on päästy aloittamaan. Toimeksiantajan lisäksi myös kehitykseen valitun yhteistyökumppanin on mahdollista hyödyntää opinnäytetyötä sekä käyttäjätarinoiden kirjallisia kuvauksia.

Ohjelmiston tulevana käyttäjänä on äärimmäisen kiinnostavaa päästä läheltä seuraamaan kuinka ohjelmistoprojekti lähtee liikkeelle ja näkemään muun muassa se, kuinka tulevia käyttäjiä tullaan projektiin osallistamaan. Opinnäytetyön aihe oli mielenkiintoinen ja kokonaisuudessaan opinnäytetyöprosessi oli todella opettavainen, se myös loi perustan ammatillisen asiantuntijuuden kasvulle. Opinnäytetyön kirjallinen raportti oli koko prosessin ajan jatkuvan muutoksen alla ja ohjelmistotutannon termein voidaankin todeta, että se eteni ”code-and-hack” -periaatteella eli työtä kasvatettiin, laajennettiin, muutettiin ja korjattiin, kunnes lopputulos oli tyydyttävä. Opinnäytetyön aikana hienointa oli huomata kuinka useat tutkinto-ohjelman aikana opitut asiat nivoutuivat yhteen ja asettuivat samalla kontekstiin.

Opinnäytetyön aikana selvisi mitä käyttäjätarinat ovat, ja kuinka niiden avulla voidaan kuvata ja dokumentoida käyttäjävaatimuksia osana ketterää ohjelmistoprojektia. Käyttäjätarinoita tarkasteltiin kuitenkin melko suppeasta näkökulmasta, joten syventyminen niihin vielä tarkemmin sekä käyttäjätarinakarttaan voisi olla tämän opinnäytetyön jatkokehityskohteena. Näin jälkikäteen pohdittuna opinnäytetyön näkökulma olisi kuitenkin voinut olla enemmän esimerkiksi käyttäjätutkimuksen menetelmissä, joten tutustuminen tarkemmin eri menetelmiin sekä käyttäjäkeskeiseen suunnitteluprosessiin olisi myös hyvä jatkokehityskohde. Opinnäytetyötä tehdessä kävi lisäksi ilmi, että käyttäjätarinoista voivat hyötyä myös muun muassa käyttäjäkokemussuunnittelijat, joten tästä näkökulmasta aihealueeseen olisi myös ehdottomasti mielenkiintoista perehtyä.

LÄHTEET

Agile Alliance 2021a. Agile 101. Hakupäivä 28.2.2021. <https://www.agilealliance.org/agile101>.

Agile Alliance 2021b. Epic. Hakupäivä 18.4.2021. <https://www.agilealliance.org/glossary/epic>.

Agile Alliance 2021c. Incremental development. Hakupäivä 28.2.2021. <https://www.agilealliance.org/glossary/incremental-development>.

Agile Alliance 2021d. INVEST. Hakupäivä 5.4.2021. <https://www.agilealliance.org/glossary/invest>.

Agile Alliance 2021e. Iteration. Hakupäivä 28.2.2021. <https://www.agilealliance.org/glossary/iteration>.

Agile Alliance 2021f. Iterative development. Hakupäivä 28.2.2021. <https://www.agilealliance.org/glossary/iterative-development>.

Agile Alliance 2021g. Product backlog. Hakupäivä 5.4.2021. <https://www.agilealliance.org/glossary/backlog>.

Agile Alliance 2021h. Story mapping. Hakupäivä 18.4.2021. <https://www.agilealliance.org/glossary/storymap>.

Agile Alliance 2021i. The three C's. Hakupäivä 5.4.2021. <https://www.agilealliance.org/glossary/three-cs>.

Agile Alliance 2021j. User stories. Hakupäivä 28.2.2021. <https://www.agilealliance.org/glossary/user-stories>.

Agile Alliance 2021k. User story template. Hakupäivä 6.4.2021. <https://www.agilealliance.org/glossary/user-story-template/>.

Alfame 2021. Vaatimusmäärittely ketterässä ohjelmistokehityksessä - 4 askelta onnistuneeseen määrittelytyöhön. Ladattava opas. Pdf-tiedosto. Hakupäivä 24.2.2021. <https://www.alfame.com/laataa-vaatimusmaarittely-ketterassa-ohjelmistokehityksessa-opas>.

Atlassian 2021a. Confluence. Hakupäivä 31.7.2021. <https://www.atlassian.com/software/confluence>.

Atlassian 2021b. Jira Software. Hakupäivä 31.7.2021. <https://www.atlassian.com/software/jira>.

Beck, Kent, Beedle, Mike, Bennekum, Arie van, Cockburn, Alistair, Cunningham, Ward, Fowler, Martin, Grenning, James, Highsmith, Jim, Hunt, Andrew, Jeffries, Ron, Kern, Jon, Marick, Brian, Martin, Robert C., Mellor, Steve, Schwaber, Ken, Sutherland, Jeff & Thomas, Dave 2001a. Manifesto for agile software development. Hakupäivä 28.2.2021. <https://agilemanifesto.org/>.

Beck, Kent, Beedle, Mike, Bennekum, Arie van, Cockburn, Alistair, Cunningham, Ward, Fowler, Martin, Grenning, James, Highsmith, Jim, Hunt, Andrew, Jeffries, Ron, Kern, Jon, Marick, Brian, Martin, Robert C., Mellor, Steve, Schwaber, Ken, Sutherland, Jeff & Thomas, Dave 2001b. Principles behind the Agile Manifesto. Hakupäivä 28.2.2021. <https://agilemanifesto.org/principles.html>.

Beck, Kent & Fowler, Martin 2001. Planning extreme programming. Boston: Addison-Wesley. Hakupäivä 6.4.2021. O'Reilly. Vaatii käyttöoikeuden.

Brooks, Frederick 1987. No silver bullet: essence and accidents of software engineering. Computer 20 (4), 10–19. Hakupäivä 16.7.2021. 10.1109/MC.1987.1663532.

Capgemini 2019. Johdon opas onnistuneeseen liiketoimintaprosessien automatisointihankkeeseen. Ladattava opas. Pdf-tiedosto. Hakupäivä 2.5.2021. <https://www.capgemini.com/fin/2019/01/blog-alykas-automaatio-viisi-vinkkia/>.

Cohn, Mike 2009. User stories applied: for agile software development. 13th printing. Boston, MA: Addison-Wesley. Hakupäivä 24.2.2021. O'Reilly. Vaatii käyttöoikeuden.

Dagmar 2021a. Dagmar. Hakupäivä 24.2.2021. <https://www.dagmar.fi/dagmar/>.

Dagmar 2021b. Dagmar Drive. Hakupäivä 24.2.2021. <https://www.dagmar.fi/dagmar-drive/>.

Dagmar 2021c. Dagmar Media. Hakupäivä 24.2.2021. <https://www.dagmar.fi/dagmar-media/>.

Dagmar 2021d. Dagmar Stories. Hakupäivä 24.2.2021. <https://www.dagmar.fi/dagmar-stories/>.

Eficode 2021. Ketterä ohjelmistokehitys. Hakupäivä 30.5.2021. <https://www.eficode.com/fi/palvelut/suunnittelu-kettera-ohjelmistokehitys>.

Garreta-Domingo, Muriel 2021. User stories: as a [UX designer] I want to [embrace agile] so that [I can make my projects user-centered]. Interaction Design Foundation. Hakupäivä 13.4.2021. <https://www.interaction-design.org/literature/article/user-stories-as-a-ux-designer-i-want-to-embrace-agile-so-that-i-can-make-my-projects-user-centered>.

Garstin, Katie 2020. Why you should consider hybrid project management. Nine Labs. Hakupäivä 31.5.2021. <https://www.ninelabs.com/insights/why-you-should-consider-hybrid-project-management>.

Gyursanszky, Anthony 2020. Totta vai tarua - nämä neljä asiaa vievät ohjelmistokehityksen tehot. Codento. Hakupäivä 24.2.2021. <https://blog.codento.com/totta-vai-tarua>.

Haikala, Ilkka & Mikkonen, Tommi 2011. Ohjelmistotuotannon käytännöt. 12. uudistettu painos. Helsinki: Talentum.

Hanington, Bruce 2003. Methods in the making: a perspective on the state of human research in design. Design Issues 19 (4), 9-18. Hakupäivä 4.7.2021. 10.1162/074793603322545019.

Holopainen, Ville, Kallio, Antti & Kämäräinen, Jussi 2020. MediaPron uudistus 16.11.2020. Sisäinen lähde. Dagmar.

Härkönen, Kristiina 2014. Onnistuneen tietojärjestelmähankkeen menestystekijät. Gofore. Hakupäivä 7.8.2021. <https://gofore.com/onnistuneen-tietojarjestelmahankkeen-menestystekijat/>.

IAB Europe 2021. Adex benchmark 2020 study. Pdf-tiedosto. Hakupäivä 25.6.2021. https://iabeuropa.eu/wp-content/uploads/2021/05/IABEurope_AdexBenchmark_2020_2021.pdf.

IAB Finland 2021a. IAB Europe Adex 2020 - Digitaalisen mediamainonnan määrä Euroopassa +6,3 % kasvussa. Hakupäivä 25.6.2021. <https://www.iab.fi/uutiset-blogi/uutiset-ja-tiedotteet/iab-europe-adex-2020-digitaalisen-mediainonnan-maara-euroopassa-63-kasvussa.html>.

IAB Finland 2021b. Verkkomainonta kasvoi 2,2% haastavassa tilanteessa 2020 - verkkomainonta jo lähes puolet koko mainoskakausta. Hakupäivä 25.6.2021. <https://www.iab.fi/uutiset-blogi/kvartaalitiedotteet/verkkomainonta-kasvoi-22-haastavassa-tilanteessa-2020-verkkomainonta-jo-lahes-puolet-koko-mainoskakausta.html>.

IAB Finland 2021c. Verkkomainonta mukavassa kasvussa vuoden ensimmäisellä kvartaalilla. Hakupäivä 25.6.2021. <https://www.iab.fi/uutiset-blogi/uutiset-ja-tiedotteet/verkkomainonta-mukavassa-kasvussa-vuoden-ensimmaisella-kvartaalilla.html>.

IEEE 610.12 1990. IEEE Standard glossary of software engineering terminology. Hakupäivä 24.2.2021. 10.1109/IEEESTD.1990.101064.

Innanen, Piia 2021. Palvelumuotoilun menetelmä - kyselytutkimukset. Palvelumuotoilu Palo. Hakupäivä 4.7.2021. <https://www.palvelumuotoilupalo.fi/blogi/2021/1/22/palvelumuotoilun-menetelma-kyselytutkimus>.

Jeffries, Ron 2001. Essential XP: Card, Conversation, Confirmation. Hakupäivä 5.4.2021. <https://ronjeffries.com/xprog/articles/expcardconversationconfirmation/>.

Juvonen, Rami 2018. Ohjelmistoprojektin sudenkuopat ja miten ne vältetään. Helsinki: Books on Demand. Hakupäivä 3.4.2021. E-kirja. Vaatii käyttöoikeuden.

Kantar 2021a. Mediamainonnan määrä 2020. Lehdistötiedote. Pdf-tiedosto. Hakupäivä 25.6.2021. https://www.kantar.fi/sites/default/files/lehdistotiedote_mediamainonnan_maara_2020.pdf.

Kantar 2021b. Mediamainonnan määrä 2021 Q1. Kvartaalitiedote. Pdf-tiedosto. Hakupäivä 25.6.2021. https://www.kantar.fi/sites/default/files/mediamainonnan_maara_q1_2021.pdf.

Kauppalehti 2021. Kasvajakonsernien joukko kutistui – M&M listasi Suomen suurimmat toimistoryhvät. Hakupäivä 24.2.2021. <https://www.kauppalehti.fi/uutiset/kasvajakonsernien-joukko-kutistui-mm-listasi-suomen-suurimmat-toimistoryhmat/5114676b-9e71-498d-aae4-9bad09944926>.

Komulainen, Sari 2021. Palvelumuotoilun menetelmä - havainnointi. Palvelumuotoilu Palo. Hakupäivä 4.7.2021. <https://www.palvelumuotoilupalo.fi/blogi/2021/2/17/palvelumuotoilun-menetelma-havainnointi>.

Kulmala, Tatu 2016. Ketterät IT-sopimukset. Lukander Ruohola HTO. Hakupäivä 3.4.2021. <https://www.lrhto.fi/artikkelit/ketterat-it-sopimukset/>.

Kuoppa, Katri 2021. Sovellussuunnittelija. Salomaa Yhtiöt Oy. MediaPron esittely 17.3.2021.

Kähönen, Päivi 2016. Vaatimusmäärittelyopas. Pasaati. Ladattava opas. Pdf-tiedosto. Hakupäivä 24.2.2021.

Laine, Henri 2021. Havainnoja ketteryydestä ja ohjelmistojärjestelmien dokumentaatiosta. Digia. Hakupäivä 31.7.2021. <https://blog.digia.com/havainnoja-ketteryydesta-ja-ohjelmistojarjestelmien-dokumentaatiosta>.

Leffingwell, Dean 2011. Agile software requirements: lean requirements practices for teams, programs, and the enterprise. Upper Saddle River, NJ: Addison Wesley. Hakupäivä 24.2.2021. O'Reilly. Vaatii käyttöoikeuden.

Lucia, Andrea & Qusef, Abdallah 2010. Requirements engineering in agile software development. Journal of Emerging Technologies in Web Intelligence 2 (3), 212–220. Hakupäivä 9.5.2021. 10.4304/jetwi.2.3.212-220.

Luukkainen, Matti & Ilves, Kalle 2021a. Ohjelmistojen vaatimusmäärittely, tuotteen ja sprintin hallinta. Ohjelmistotuotanto avoin yliopisto. Hakupäivä 25.2.2021. <https://ohjelmistotuotanto-hy-avoin.github.io/osa2/>.

Luukkainen, Matti & Ilves, Kalle 2021b. Ohjelmistotuotanto, ketteruus, Scrum. Ohjelmistotuotanto avoin yliopisto. Hakupäivä 27.2.2021. <https://ohjelmistotuotanto-hy-avoin.github.io/osa1/>.

Mountain Goat Software 2021. User stories. Hakupäivä 18.4.2021. <https://www.mountaingoatsoftware.com/agile/user-stories>.

Palvelumuotoilu Palo 2021. Palvelumuotoilun menetelmä - haastattelu. Hakupäivä 4.7.2021. <https://www.palvelumuotoilupalo.fi/blogi/2021/2/17/palvelumuotoilun-menetelma-haastattelu>.

Patton, Jeff & Economy, Peter 2014. User story mapping: discover the whole story, build the right product. Sebastopol, CA: O'Reilly Media. Hakupäivä 28.2.2021. O'Reilly. Vaatii käyttöoikeuden.

Pohjonen, Risto 2002. Tietojärjestelmien kehittäminen. Jyväskylä: Docendo. Hakupäivä 7.8.2021. Ellibs. Vaatii käyttöoikeuden.

ProjectTOP 2021. Vaatimusmäärittelyn 5 ongelmaa ja yksi ratkaisu. Hakupäivä 1.6.2021. <https://projecttop.com/vaatimusmaarittelyn-5-ongelmaa-ja-yksi-ratkaisu/>.

Riippi, Juha 2018. Et ikinä usko mitkä kolme asiaa menevät pieleen Vincitin asiakasprojekteissa! Vincit. Hakupäivä 18.7.2021. <https://www.vincit.fi/fi/et-ikina-usko-mitka-kolme-asiaa-menevat-pieleen-vincitin-asiakasprojekteissa/>.

Robertson, Suzanne & Robertson, James 2013. Mastering the requirements process: getting requirements right. Third edition. Upper Saddle River, NJ: Addison Wesley. Hakupäivä 24.2.2021. O'Reilly. Vaatii käyttöoikeuden.

Ruuska, Ari 2020. Vältä Agilen kantapää – huomioi nämä seikat ketterän projektin ostamisessa. Wunder. Hakupäivä 26.5.2021. <https://wunder.io/fi/artikkelit/valta-agilen-kantapaa-huomioi-nama-seikat-ketteran-projektin-ostamisessa/>.

Schön, Eva-Maria, Thomaschewski, Jörg & Escalona, María José 2017a. Agile requirements engineering: a systematic literature review. Computer Standards & Interfaces 49, 79–91. Hakupäivä 8.5.2021. <https://doi.org/10.1016/j.csi.2016.08.011>.

Schön, Eva-Maria, Thomaschewski, Jörg & Escalona, María José 2017b. Identifying agile requirements engineering patterns in industry. In proceedings of the 22nd European conference on pattern languages of programs. Association for Computing Machinery 26, 1–10. Hakupäivä 8.5.2021. <https://doi.org/10.1145/3147704.3147733>.

Sofokus 2021. Ohjelmistokehitys. Hakupäivä 30.5.2021. <https://www.sofokus.com/fi/expertise/ohjelmistokehitys/>.

Tirkkonen, Janne 2019. Tiesitkö, että myös asiakasta tarvitaan ohjelmistokehityksessä? Alfame. Hakupäivä 14.4.2021. <https://www.alfame.com/blog/myos-asiakasta-tarvitaan-ohjelmistokehityksessa>.

Vainikainen, Kimmo 2020. 2020-luku avaa uuden teknologisen aikakauden. Talentree. Hakupäivä 24.2.2021. <https://talentree.fi/softa/uusi-teknologinen-aikakausi/>.

Vehviläinen, Lassi 2021. Näin teet käyttäjälähtöistä suunnittelua etätyöaikana. Fraktio. Hakupäivä 1.5.2021. <https://www.fraktio.fi/blogi/kayttajalahtoinen-suunnittelu-etatyoaikana>.

Visual Paradigm 2021a. Theme vs epic vs user story vs task. Hakupäivä 18.7.2021. <https://www.visual-paradigm.com/scrum/theme-epic-user-story-task/>.

Visual Paradigm 2021b. What is user story? Hakupäivä 28.2.2021. <https://www.visual-paradigm.com/guide/agile-software-development/what-is-user-story/>.

Wake, Bill 2003. INVEST in good stories, and SMART tasks. XP123. Hakupäivä 24.2.2021. <https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>.

Wiegers, Karl Eugene & Beatty, Joy 2013. Software requirements. Third edition. Redmond, WA: Microsoft Press. Hakupäivä 24.2.2021. O'Reilly. Vaatii käyttöoikeuden.

Young, Ralph Rowland 2004. The requirements engineering handbook. Boston: Artech House. Hakupäivä 26.2.2021. ProQuest Ebook Central. Vaatii käyttöoikeuden.

Alkusanat

- Haastattelun tarkoitus
- Luottamuksellisuus
- Luvan pyytäminen nauhoittamiselle

Kysymykset

1. Voisitko kuvailla lyhyesti keskeisimmät työtehtäväsi osana nykyistä suunnittelu- ja ostoprosessia?
2. Jos pohdit nykyistä suunnittelu- ja ostoprosessia, niin mikä tai mitkä työtehtävät vievät sinulta eniten aikaa? Miksi?
3. Millainen rooli markkinoinnin suunnittelu- ja laskutusohjelmistolla on osana mainitsemiasi työtehtäviä ja ylipäätään osana suunnittelu- ja ostoprosessia?

– Digikampanjan suunnitteluprosessi (sisäinen materiaali) –

4. Voisitko kuvailla hieman tarkemmin missä tilanteissa käytät markkinoinnin suunnittelu- ja laskutusohjelmistoa? Ja kuinka usein?
5. Millä tavalla markkinoinnin suunnittelu- ja laskutusohjelmisto tukee nykyistä suunnittelu- ja ostoprosessia?
6. Kuinka suunnittelu- ja ostoprosessia voitaisiin ohjelmiston avulla mielestäsi tukea paremmin?
7. Jos pohdit markkinoinnin suunnittelu- ja laskutusohjelmiston käyttöä, niin mihin asiaan tai asioihin kuluu eniten aikaa? Miksi?
8. Mitkä ovat mielestäsi haasteet ohjelmiston käytössä? Miten nämä vaikuttavat työntekoon ja arjen sujuvuuteen?
9. Onko jotain haasteita, jotka toistuvat kerrasta toiseen?
10. Mitä muuttaisit joko markkinoinnin suunnittelu- ja laskutusohjelmistossa tai ylipäätään suunnittelu- ja ostoprosessissa, jotta työt sujuisivat jouhevammin?
11. Kuinka manuaalista työtä voitaisiin mielestäsi poistaa?

12. Mitkä ovat haasteet ohjelmiston käytettävyyteen liittyen?

- Tuloksellisuus – Saatko asiat tehtyä mahdollisimman virheettömästi ja ongelmattomasti?
- Tehokkuus – Saatko asiat tehtyä aikaa tuhlaamatta?
- Tyytyväisyys – Mitä mieltä olet ohjelmistosta ja kuinka miellyttävää sitä on käyttää?

13. Mitkä ovat mielestäsi markkinoinnin suunnittelu- ja laskutusohjelmiston hyvät puolet? Mitä haluaisit ehdottomasti säilytettävän uudessa ohjelmistossa?

14. Tuleeko sinulla jotain muuta mieleen mitä olisi hyvä ottaa huomioon tai nostaa esiin?

Loppusanat

- Vapaat kommentit ja varmistaminen haluaako haastateltava vielä lisätä tai kysyä jotain
- Kertominen kuinka opinnäytetyöhön liittyvä prosessi etenee
- Kiitokset

	A	B
1	Käyttäjärooli	Käyttäjätarina
2	Digisuunnittelija	Haluan luoda uuden kampanjan, jotta saan julkaisusuunnitelman tehtyä.
3	Digisuunnittelija	Haluan lisätä uuden rivin, jotta se näkyy julkaisusuunnitelmalla.
4	Digisuunnittelija	Haluan toimittaa julkaisusuunnitelman asiakkaalle, jotta saan siihen hyväksynnän.
5	Digisuunnittelija	Haluan toimittaa määräimen medialle, jotta saan kampanjan vahvistettua.
6	Digisuunnittelija	Haluan siirtää rivin tilauskantaan, jotta se näkyy asiakkaan myyntilaskulla.
7	Digisuunnittelija	Haluan siirtää rivin jäädytettynä tilauskantaan, jotta ostolasku pystytään tarkistamaan ja maksamaan.
8	Digisuunnittelija	Haluan tarkastella mediasuunnitelmaa, jotta saan tekniset ohjeet tehtyä.
9	Digisuunnittelija	Haluan tarkastella mediasuunnitelmaa, jotta saan nimeämisexcelin tehtyä.
10	Digisuunnittelija	Haluan tarkastella mediasuunnitelmaa, jotta saan kampanjan trafikoitua.
11	Digisuunnittelija	Haluan tarkastella mediasuunnitelmaa, jotta saan kampanjan rakennettua ostojärjestelmään.
12	Digisuunnittelija	Haluan tarkastella onko korjauslaskuehdotuksia tullut, jotta saan hyvityksen tehtyä.
13	Digisuunnittelija	Haluan tarkastella onko korjauslaskuehdotuksia tullut, jotta saan lisäveloituksen tehtyä.
14	Digisuunnittelija	Haluan tarkastella määräimen tilaa, jotta tiedän voinko kohdistaa siihen ostolaskun.
15	Digisuunnittelija	Haluan suodattaa kampanjoita, jotta saan haettua tietyn kriteerin mukaan.