

ELLA HOLMALA

ASIAKASVAATIMUKSIEN TÄYTTÄMISEN HAASTEET
TIETOJÄRJESTELMIEN KÄYTTÄJÄKESKEISESSÄ
SUUNNITTELUSSA

Kandidaatintyö
Huhtikuu 2019

TIIVISTELMÄ

Ella Holmala: Vaatimuksien täyttymisen haasteet tietojärjestelmien käyttäjäkeskeisessä suunnittelussa

Challenges in fulfilling requirements in user-centric information system design

Tampereen Yliopisto

Kandidaatintyö, 26 sivua

Maaliskuu 2019

Teknis-taloudellinen TkK-tutkinto-ohjelma

Pääaine: Tietojohtaminen

Tarkastaja: TkT Pasi Hellsten

Avainsanat: Vaatimusmäärittely, tietojärjestelmä, käyttäjäkeskeisyys, suunnittelu

Tietojärjestelmät ovat nykypäivän liiketoiminnan tärkeimpiä apuvälineitä. Tietojärjestelmien avulla voidaan jakaa, tallentaa ja hyödyntää liiketoiminnassa käytettäviä tietoja. Järjestelmien suunnittelun haasteena on ymmärtää asiakkaita ja tarkemmin järjestelmän käyttäjiä. Tässä tutkimuksessa halutaan tarkastella näitä haasteita ja selvittää syitä niiden ilmenemiseen.

Tutkimuksen tietolähteinä toimivat sähköiset kirjat, artikkelit, diplomityöt, konferenssi-julkaisut sekä fyysiset kirjat. Tutkimuksen päätutkimuskysymyksenä oli: ”Mitä haasteita ilmenee vaatimusten yhtensovittamisessa tietojärjestelmien käyttäjäkeskeiseen suunnitteluun?”. Päätutkimuskysymyksen lisäksi tutkimuksessa haluttiin selventää keskeisiä käsitteitä, kuten vaatimusmäärittely ja käyttäjäkokemus. Nämä käsitteet tukevat toisiaan tietojärjestelmien suunnittelussa, jonka takia niitä halutaan täsmentää lukijoille. Lisäksi tutkimuksessa tutkitaan vaatimusmäärittelyn vaiheita sekä tietojärjestelmän kehittämisen prosessia. Tutkimuksen tarkoitus on hahmottaa vaatimusmäärittelyn haasteita sekä vaatimusten yhtensovittamisen haasteita tietojärjestelmien käyttäjäkeskeiseen suunnitteluun.

Tutkimuksen perusteella voidaan tehdä johtopäätös, että vaatimusten yhtensovittaminen tietojärjestelmien käyttäjäkeskeiseen suunnitteluun ei ole yksinkertaista. Vaatimusmäärittely on tietojärjestelmien suunnittelun ensimmäisiä, mutta myös kriittisimpiä vaiheita. Vaatimusmäärittelyssä ilmenevät ongelmat haittaavat tulevia suunnitteluprosessin vaiheita. Haasteina ovat esimerkiksi asiakkaan tietämättömyys omista tarpeistaan tai suunnittelijoiden väärä käsitys asiakkaiden tarpeista, jotka eivät ole todenmukaisia. Tutkimuksen tuloksien avulla tietojärjestelmäsuunnittelijat voivat hahmottaa käytännön suunnitteluratkaisuissaan tutkimuksessa esiin tulevia haasteita ja kehittää suunnittelua enemmän käyttäjäkeskeisemmäksi ja asiakaslähtöisemmäksi.

ALKUSANAT

Tämä kandidaatintyö on tehty tietojohdamisen koulutusohjelmaan keväällä 2019. Tutkimuksen aiheen valitseminen perustui kiinnostukseeni tietojärjestelmistä sekä käyttäjäkokemuksesta. Mielenkiintoni aiheeseen ohjasi yleinen käsitys vaatimusten toteutumattomuudesta tietojärjestelmissä, minkä takia halusin tutkia aiheen haasteita vielä tarkemmin. Kiitos Pasi Hellstenille avustamisesta ja ohjaamisesta työn aikana sekä kurssin pienryhmän osallistujille palautteista ja kehitysehdotuksista.

Tampereella, 11.3.2019.

Ella Holmala

SISÄLLYSLUETTELO

1.	JOHDANTO	1
1.1	Tutkimuksen tausta	1
1.2	Tutkimuksen rajaus ja tutkimuskysymykset	2
2.	TUTKIMUSMENETELMÄN ESITTELY	4
2.1	Tutkimusmenetelmä	4
2.2	Tutkimusaineiston esittely	5
3.	VAATIMUSMÄÄRITTELY	7
3.1	Vaatimusmäärittely käsitteenä	7
3.2	Vaatimusmäärittelyn vaiheet	8
3.2.1	Vaatimuksien syntyminen	9
3.2.2	Vaatimuksien analysoiminen	10
3.2.3	Vaatimuksien vahvistaminen	11
3.2.4	Dokumentointi ja halliinta määrittelyn tukitoimintoina	11
3.3	Sidosryhmät	12
4.	KÄYTTÄJÄKESKEINEN TIETOJÄRJESTELMIEN SUUNNITTELU	13
4.1	Käyttäjäkokemuksen merkitys	13
4.2	Tietojärjestelmien elinkaaren vaiheet	14
4.2.1	Esiselvitys	14
4.2.2	Vaatimusmäärittely	15
4.2.3	Suunnittelu	15
4.2.4	Testaus	15
4.2.5	Käyttöönotto	16
4.3	Ihmisen ja järjestelmän vuorovaikutuksen ergonomia	17
5.	VAATIMUKSIEN TOTEUTUMISEN HAASTEET KÄYTTÄJÄKESKEISESSÄ SUUNNITTELUSSA	19
5.1	Vaatimusmäärittelyn haasteet	19
5.2	Vaatimuksien toteutumisen haasteet	20
6.	YHTEENVETO	22
6.1	Päätelmät	22
6.2	Jatkotutkimusehdotukset	22

KESKEISET KÄSITTEET

B2B:	Yrityksien välistä kaupankäyntiä. Lyhenne on englanniksi ”Business-to-business”.
Graafinen käyttöliittymä:	Käyttäjän tietokonenäytölle tuleva visuaalinen näkymä, jolla voidaan parantaa vuorovaikutusta käyttäjän ja ohjelmiston välillä (Thong et al. 2002). Käsite on englanniksi ”graphical user interface” ja siitä käytetään usein lyhennettä ”GUI”.
Käyttäjä:	Käyttäjät lisäävät, muokkaavat ja hyödyntävät järjestelmän tietoja. Käyttäjät ovat tässä tutkimuksessa keskiössä ja heidän näkökulmaansa painotetaan. Tutkimuksessa puhutaan myös asiakkaista, johon voidaan sisällyttää järjestelmän loppukäyttäjät.
Käyttäjakeskeisyys:	Käyttäjakeskeisyydellä tarkoitetaan suunnittelumenetelmää, missä käyttäjien tarpeet ovat keskiössä. Tutkimuksen aikana puhutaan ihmiskeskeisestä suunnittelusta, joka käsittää käyttäjakeskeisen suunnittelun.
Käyttäjäkokemus:	Muodostuu käyttäjän ennakko-odotuksista ja järjestelmän käytöstä aiheutuvista reaktioista (Bevan, 2019). Käyttäjäkokemus on englanniksi ”user experience” ja siitä usein käytetään lyhennettä ”UX”.
Rajapinta:	Käyttäjälle näkyvä osuus ohjelmistosta. Rajapinnan takana on monimutkainen ohjelman toteutus, josta epäolennaisimmat tiedot ovat piilotettu käyttäjältä. (Haikala & Mikkonen 2011, s.182) Rajapinta on englanniksi ”Interface”.
Tietojärjestelmä:	Tietojärjestelmät käsittävät laitteistot ja ohjelmistot, joiden avulla voidaan käsitellä tietoa eri liiketoimintaprosesseissa (Yeo 2002). Tutkimuksessa kuvataan tietojärjestelmää myös sanalla tuote, koska liiketoiminnallisesta näkökulmasta suunniteltava järjestelmä on asiakkaalle tuleva tuote.
Vaatimusmäärittely:	Selvitys sidosryhmien vaatimuksista esimerkiksi suunniteltavalle järjestelmälle. Vaatimusmäärittelyt ovat englanniksi ”requirements engineering”.

1. JOHDANTO

1.1 Tutkimuksen tausta

Tietojärjestelmät ovat nykypäivän liiketoimintaa tukevia apuvälineitä. Tietoa on mahdollista jakaa tehokkaasti tietojärjestelmien avulla ja ne tukevat liiketoimintaprosessien toimintoja. Järjestelmät varastoivat tietoa ja mahdollistavat tiedon jakamisen eri osastojen välillä. Tietojärjestelmät ovat laajoja kokonaisuuksia, mitkä koostuvat ohjelmistoista ja laitteistoista (Yeo 2002). Järjestelmien isojen hankintakustannuksien takia asiakkaat haluavat tietää paljonko sijoitettu pääoma tuottaa (Petter et al. 2008). Tietojärjestelmän onnistunut suunnittelu on tärkeätä, jotta sijoitetulle pääomalle saadaan vastinetta.

Alkuaikoina ohjelmistoja tuotettiin ilman sen suurempaa suunnittelua (Hoda, 2018). Suunnittelematon tietojärjestelmä voi aiheuttaa merkittäviä määriä kustannuksia organisaatiolle. Ennen kuin tietojärjestelmää aletaan suunnittelemaan, täytyy tietojärjestelmällä olla merkitys. Tietojärjestelmän tulisi ratkaista jokin olemassa oleva ongelma organisaation liiketoiminnassa. Ongelmana voisi olla esimerkiksi asiakkaiden tietojen jakaminen organisaation sisäisesti. Siten asiakashallintajärjestelmä auttaa tiedon jakamista sekä tiedon tallentamista. Järjestelmät eivät kuitenkaan ole yhdenmukaisia jokaisella toimialalla, minkä takia vaatimuksia on tärkeä kartoittaa tarkasti jokaisen asiakkaan kohdalla.

Vaatimuksia asettavat järjestelmän sidosryhmät. Tärkeimpiä vaatimuksia asettaa asiakas. Muita näkökulmia voivat olla esimerkiksi myyjän tai sijoittajan näkökulma. Tietojärjestelmien kompleksisen luonteen takia osa sidosryhmän vaatimuksista jää toteuttamatta. Vaikka asiakkaiden vaatimuksia pidetään tärkeimpänä, heiltä saavat vaatimukset eivät usein ole sellaisenaan riittäviä, sillä he eivät usein ole edes tietoisia omista tarpeistaan (Buschmann 2011). Vaikeuksia syntyy liiketoimintaongelmien hahmottamisessa sekä niiden ilmaisemisessa. Goldsmithin (2004) mukaan asiakkaiden lisäksi ongelmia ilmenee suunnittelijoiden takia. Suunnittelijat usein kuvittelevat tietävänsä asiakkaiden vaatimuksista, jonka takia he eivät huomioi asiakkaita riittävästi. (Goldsmith 2004)

Tietojärjestelmäprojektit epäonnistuvat huonon suunnittelun takia. Ajankäytön väärä arviointi on yksi syy projektien epäonnistumiseen. (Yeo 2002) Kuutilan (2019) mukaan ajan puutteesta johtuva paine vaikuttaa suunnitteluprosessiin negatiivisesti. Ajan puute vaikuttaa usein esimerkiksi testausvaiheeseen, sillä se on suunnittelussa viimeinen vaihe. Testaaminen on kuitenkin hyvin tärkeä vaihe, koska ilman testausta voi olla hankala löytää järjestelmän virheitä. (Kuutila 2019) Isoin ongelma suunnittelussa on ennalta määritettyjen tavoitteiden saavuttamisessa (Yeo 2002). Asiakkaan ja toimittajan välille syntyy

kommunikaatiokuilu, jonka takia projektit eivät onnistu riittävällä tasolla (Goldsmith 2004).

1.2 Tutkimuksen rajausta ja tutkimuskysymykset

Tutkimuksessa arvioidaan suunnittelijoiden suoriutumista käyttäjien vaatimusten ymmärtämisessä ja selvittää syitä, miksi vaatimukset eivät mahdollisesti täyty. Kommunikointi käyttäjän ja toimittajan välillä ei onnistu tavoitteiden mukaisesti (Goldsmith 2004), minkä takia tutkimuksen taustalla on oletus vaatimusten toteutumattomuudesta tietojärjestelmien suunnittelussa. Tutkimuksessa on tarkoitus keskittyä käyttäjakeskeiseen suunnitteluun ja asiakkaan vaatimusmäärittelyihin.

Tutkimusongelmaksi valitsin vaatimusten yhteensovittamisen tietojärjestelmien suunnittelussa. Päättökysymyksenä on:

- Mitä haasteita on vaatimusten yhteensovittamisessa tietojärjestelmien käyttäjakeskeiseen suunnitteluun?

Alatutkimuskysymyksiä ovat:

1. Mikä on vaatimusmäärittely?
2. Mitä on käyttäjakeskeisyys?
3. Miten käyttäjien vaatimukset toteutuvat tietojärjestelmissä?
4. Missä vaiheessa asiakas otetaan mukaan suunnitteluprosessiin?

Tutkimuksen tavoitteena on selvittää, millä tasolla vaatimukset täyttyvät suunnittelussa. Tutkimuksessa on selvitettävä tietojärjestelmän kehityksen vaiheet ja tavat kerätä vaatimuksia käyttäjiltä. Jos vaatimukset eivät täyty suunnittelussa, on selvitettävä syyt niille. Tutkimuksessa käsitellään vain liiketoiminnassa käytettyjä tietojärjestelmiä. Sen lisäksi tietojärjestelmissä keskitytään ohjelmistoihin. Vaatimuksia tarkastellaan pääasiassa vain asiakkaan näkökulmasta.

Tutkimuksessa ensin käsitellään tietojärjestelmien vaatimusmäärittelyitä. Vaatimusmäärittelyt tukevat suunnittelua, joten vaatimuksia tulisi tutkia ja esitellä ensimmäisenä. Vaatimuksia voisi tarkastella kaikkien sidosryhmien näkökulmista, mutta tässä tutkimuksessa rajataan asiakkaan näkökulmaan. Asiakkaan vaatimukset ovat kuitenkin jokaisen liiketoiminnan lähtökohta, joten tässä kontekstissa olisi mielenkiintoista nähdä millä tasolla asiakkaan vaatimukset yleensä täyttyvät.

Tutkimuksessa on tavoitteena löytää mitä asioita tulisi huomioida tietojärjestelmien suunnittelussa. Tietojärjestelmän suunnittelu on yksi järjestelmäkehityksen vaiheista, minkä takia tässä tutkimuksessa selvitetään kaikki tietojärjestelmäkehityksen vaiheet. Sen lisäksi selvitetään, missä vaiheessa asiakas otetaan mukaan kehitysprosessissa. Tutkimuk-

sessä halutaan selvittää, toteutuvatko asiakkaan vaatimukset suunnittelussa. Koska oletuksena on vaatimuksien toteutumattomuus, halutaan selvittää mitä haasteita toteuttamiseen liittyy.

Tutkimuksen tuloksen tavoitteena on tunnistaa vaatimusmäärittelyn haasteita tietojärjestelmien suunnittelussa. Haasteiden tunnistamisen avulla voidaan keksiä ratkaisuja niille. Ratkaisujen perusteella voidaan tehostaa tietojärjestelmän suunnittelua ja siten saada nopeammin tietojärjestelmiä käyttöön. Tässä tutkimuksessa ei kuitenkaan ole tarkoitus selvittää ratkaisuja, vaan tarkastella pelkästään haasteita.

2. TUTKIMUSMENETELMÄN ESITTELY

2.1 Tutkimusmenetelmä

Tämä tutkimus toteutetaan kirjallisuuskatsauksena. Tiedon lähteinä käytetään seuraavia hakukoneita: Google Scholar, Andor, SpringerLink, IEEE ja Scopus. Hakutulokset on rajattu vain 2000-luvulla julkaistuihin materiaaleihin. Hakulausekkeissa käytettiin Boolean operaattoreita AND, OR ja NOT. Hakulausekkeet olivat lähes kaikki englanninkielisiä.

Taulukko 1. Esimerkkejä hakulausekkeista ja tuloksista

Hakulauseke	Google Scholar	Scopus	Andor	IEEE	Springer-Link
"requirements engineering" AND "agile software development"	5,970	6	906	51	71
"requirements engineering" AND "information system" AND "agile"	4190	4	1811	3	168
"information system" AND "user-centric design"	485	2	335	0	21

Taulukon 1 mukaan lähteitä löytyi eri hakusanoilla runsaasti. Google Scholarista löytyi eniten lähteitä. Scholarista poimituissa lähteissä pyrittiin valitsemaan lähteet, joihin on viitattu eniten. Scopus tuotti paljon vähemmän tuloksia suhteessa muihin tietokantoihin, mutta Scopus tarjosi parempia rajausmahdollisuuksia. Lähteistä pystyttiin esimerkiksi karsimaan maksulliset lähteet, jotka pienensivät lopputuloksen lukumäärää merkittävästi.

Yksi tärkein kriteeri lähteiden valitsemiseen oli lähteiden tuoreus. Lähteiden tuli olla vähintään 2000-luvulla julkaistu. Lähteissä pyrittiin huomioimaan julkaisufoorumien hyväksymiä julkaisuja, sillä niitä ollaan arvioitu tarkemmin ja ovat todennäköisemmin luotettavimpia. Pääasiassa aineistoa valittiin perusteella, kuinka hyvin se sopi tutkimusongelmaan ja tutkimuskysymyksiin.

Sopivan aineiston etsimisessä haasteena oli lähteiden maksullisuus. Tutkimusta piti tehdä yliopiston tiloissa, sillä lähteiden saatavuus oli suurimmaksi osaksi riippuvainen yliopiston verkosta. Haasteita toi myös Tampereen yliopiston uuden kirjaston tietokannan toimimattomuus, sillä osa lähteistä ei ollut saatavilla. Saatavuuden parantamiseksi lähteitä pyrittiin etsimään muista tietokannoista. Scopus osoittautui parhaimmaksi tietokannaksi lähteiden laadun kannalta. Scopuksessa oli mahdollista järjestää lähteitä esimerkiksi viittausmäärän tai julkaisuajan mukaan.

2.2 Tutkimusaineiston esittely

Tutkimusaineistona toimivat pääasiassa artikkelijulkaisut, konferenssijulkaisut, e-kirjat, fyysiset kirjat ja diplomityöt. Lähteet olivat pääasiassa englanninkielisiä, mutta muutama lähde oli suomeksi.

Taulukossa 2 on esitelty tämän tutkimuksen keskeisimmät lähteet.

Taulukko 2: Keskeisten lähteiden tekijät, nimi, kuvaus ja viittausten lukumäärä

Tekijät & Vuosi	Nimi/Otsikko	Lyhyt kuvaus	Viittausten lkm (Google Scholar)
Cadle et al. 2014	”Developing information systems: practical guidance for IT professionals”	Käsittelee tietojärjestelmän kehitystä ja sen elinkaaren vaiheita.	5
Nuseibeh B., Easterbrook S. 2000	”Requirements engineering: A Roadmap”	Esittelee tietojärjestelmien vaatimusmäärittelyn vaiheita ja toimintoja.	2482
Goldsmith R. 2004	”Discovering real business requirements for software project success”	Käsittelee liiketoiminnallisesta näkökulmasta ohjelmistoprojektien vaatimuksia.	44
ISO 9241-210 2010 (Standardi)	”Ihmisen ja järjestelmän vuorovaikutuksen ergonomia”	Sisältää ihmiskeskeisen suunnittelun periaatteita.	336 (viittaukset englanninkieliseen versioon)
Haikala, I. & Mikkonen, T. 2011	”Ohjelmistotuotannon käytännöt”	Esittelee ohjelmistoprojektien keskeisimpiä ominaisuuksia ja projektin vaiheita.	88

Taulukossa 2 esiteltiin tämän tutkimuksen keskeisimmät lähteet. Ohjelmistokehityksestä ja vaatimusmäärittelyistä löytyi paljon lähdemateriaalia. Osa potentiaalisista lähteistä jäi vielä uupumaan huonon saatavuuden takia, mutta niille löytyi korvaavia lähteitä.

Taulukosta 2 nähdään, että vanhempiin julkaisuihin on enemmän viittauksia. Jos lähteiden aikaikkunaa laajensi ennen 2000-luvulla julkaistuihin lähteisiin, lähteitä löytyi paljon enemmän. Tähän tutkimukseen oli tarkoitus löytää mahdollisimman uusia lähteitä, jonka seurauksena vanhemmat lähteet karsittiin. Uusia lähteitä oli hankala löytää, koska useimmat julkaisut olivat sidottu tiettyyn kontekstiin. Tähän tutkimukseen haluttiin löytää yleisiä haasteita vaatimusmäärittelyn yhteensovittamisessa käyttäjäkeskeiseen suunnitteluun, jonka takia niitä ei voitu käyttää.

3. VAATIMUSMÄÄRITTELY

Tietojärjestelmätoimittajat toimivat B2B-markkinoilla, missä asiakassuhteet ovat tärkeitä. B2B-markkinoilla potentiaalisia asiakkaita on merkittävästi vähemmän kuin kuluttajamarkkinoilla, jolloin asiakassuhteiden ylläpitäminen on kriittisempää. Tietojärjestelmähankkeet ovat kalliita asiakkaille, jonka takia projektin toteuttamisessa on tärkeää vähentää projektin riskejä. Vaatimusmäärittelyn avulla voidaan vähentää epäonnistumisen riskejä.

Tässä luvussa kerrotaan tarkemmin mitä vaatimusmäärittely tarkoittaa ja mitä sidosryhmiä liittyy tietojärjestelmiin. Lisäksi vaatimusmäärittelyn prosessin vaiheista kerrotaan tarkemmin alaluvuissa.

3.1 Vaatimusmäärittely käsitteenä

Vaatimusmäärittelyllä tarkoitetaan eri sidosryhmien tunnistamista ja heidän tarpeiden määrittämistä liittyen suunniteltavaan tuotteeseen (Nuseibeh & Easterbrook 2000). Vaatimuksien selvittämisen jälkeen ne dokumentoidaan, jotta tietoa voidaan myöhemmin hyödyntää (Nuseibeh & Easterbrook 2000). Dokumentit toimivat suuntaviivana suunnittelijoille, sillä ilman dokumentteja suunnittelijat eivät tietäisi käyttäjien vaatimuksista. Eberlein mukaan (2003) määrittelyssä on tarkoitus kuvailla mitä suunnitelman tulisi pitää sisällään, mutta se ei ota kantaa, kuinka ratkaisuja tulisi toteuttaa. Vaatimusmäärittelyn tavoitteena on minimoida uudelleensuunnittelusta tulevia kustannuksia (Eberlein 2003). Minimoimalla ylimääräistä suunnittelua voidaan kustannuksien lisäksi säästää merkittävästi aikaa.

Ennen virallista vaatimuksenkeruuvaihetta suunnittelijoilla on alustava näkemys suunniteltavan järjestelmän vaatimuksista (Sutcliffe 2002, s.19–43). Suunnittelijoiden omat, alustavat käsitykset tarpeista eivät silti ole riittäviä. Sutcliffen (2002, s. 19–43) mukaan kerättyjen vaatimuksien avulla suunnittelija ymmärtää enemmän järjestelmän tavoitteista ja käyttötarkoituksesta. Määrittely vaatii suunnittelijoilta myös ongelmanratkaisutaitoja (Sutcliffe 2002, s. 19–43). Ei ole yhdentekevää tuottaa ratkaisuja kompleksisille vaatimuksille, sillä jokaisella asiakkaalla on uniikit tarpeet.

Vaatimusmäärittelyn tarkoituksena on maksimoida saatava hyöty tulevasta järjestelmästä. Tietojärjestelmähankinnat ovat usein merkittäviä investointeja asiakkaille, jonka takia suunnittelupäätöksiä on tehtävä harkiten (Gable et al. 2008). Onnistuneen järjestelmän suunnittelu alkaa määrittelyillä ja sen avulla voidaan parantaa esimerkiksi käyttäjien tyytyväisyyttä ja suunnittelijoiden työn laatua (Bevan & Maguire 2002). Käyttäjien tyytyväisyys perustuu käyttäjien vaatimuksien toteutumiseen, jolloin vaatimusmäärittelyn tärkeys korostuu. Vaatimusdokumentin avulla suunnittelijoiden työn laatu paranee, sillä

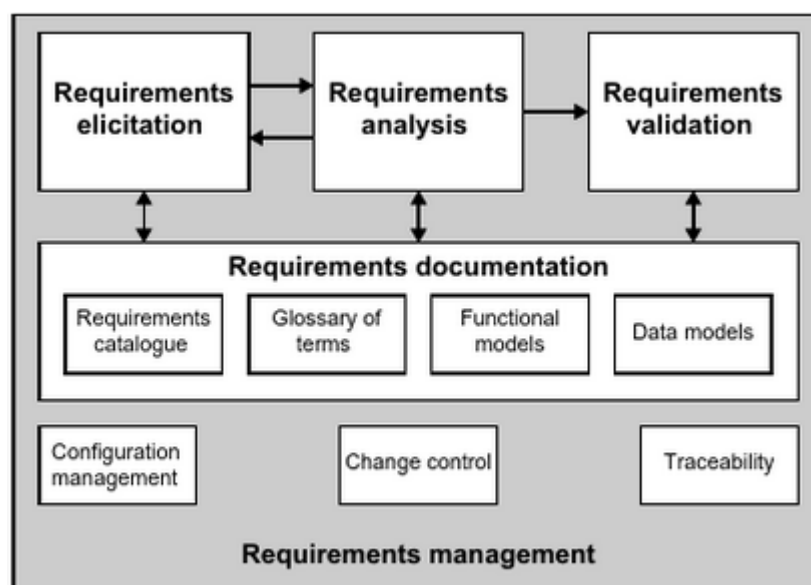
ylimääräiseen vaihtoehtojen pohdintaan ei tarvitse käyttää aikaa, jolloin kustannukset alenevat ja tuottavuus siten paranee.

Sutcliffen mukaan (2002, s. 77–102) kielenkäytön ymmärtäminen on määrittelyssä ensisijaisen tärkeää. Suunnittelijoiden on ymmärrettävä käyttäjien vaatimuksia ja vastavuo- roisesti käyttäjien on ymmärrettävä suunnittelijoiden ajatusta tuotettavasta järjestelmästä. (Sutcliffe 2002, s. 77–102) Yhteisymmärryksellä voidaan luoda ja toimittaa arvoa, mikä on ehdottoman tärkeää kasvavan kilpailun takia (Schön et al. 2017). Mikäli arvolupausta ei pystytä toimittamaan, asiakkaat voivat siirtyä kilpailijalle. Markkinoiden luonteen takia asiakkaan siirtyminen kilpailijalle voi aiheuttaa isoja taloudellisia menetyksiä.

Vaatimusmäärittely on ensimmäisiä vaiheita ohjelmistotuotannossa. Vaatimukset toimivat suunnittelijoiden ohjenuorana, sillä ne määrittävät miten käyttäjät odottavat järjestelmän toimivan (Husnain, 2017). Vaatimusmäärittelyt ovat usein asiakkaan määrittämiä, mutta suunnittelijallakin voi olla vaatimuksia. Suunnittelijan vaatimukset ovat pikemminkin rajoitteita, kuten taloudellisia tai ajallisia rajoitteita.

3.2 Vaatimusmäärittelyn vaiheet

Tietojärjestelmäprojektien onnistumiseen vaikuttaa miten järjestelmä täyttää alkuperäisen tavoitteensa (Nuseibeh & Easterbrook 2000). Vaatimusmäärittelyn tarkoituksena on selkeyttää järjestelmän tavoitetta. Sen vaiheisiin kuuluu vaatimuksien syntyminen, analysoiminen, vahvistaminen ja määrittelyn tukitoiminnot (Eberlein 2003). Määrittelyn tarkoitus on kuvata projektin lopputulos (Haikala & Mikkonen 2011, s. 21). Tietojärjestelmäprojekteissa keskeisimmät vaatimukset tulevat asiakkailta. Vaatimusmäärittelyn prosessissa on kolme vaihetta. Kuvassa 1 on esitetty eri määrittelyn vaiheet ja muita tukitoimintoja.



Kuva 1. Vaatimusmäärittelyn vaiheet (Cadle et al. 2014, s. 61)

Kuvan 1 mukaan prosessi alkaa vaatimuksien syntymisellä. Vaatimuksien syntymisellä viitataan englanninkieliseen käsitteeseen ”Requirements Elicitation”. Cadlen et al. (2014) mukaan alustavien vaatimuksien keruun jälkeen niitä analysoidaan tarkemmin. Tarvittavien vaatimuksien määrittämisen jälkeen vaatimukset vahvistetaan suhteessa suunniteltavaan järjestelmään. Koko prosessin aikana vaatimuksia dokumentoidaan ja hallitaan. (Cadle et al. 2014, s. 61–62) Prosessia tehdään käytännössä iteratiivisesti, jotta muuttuvat vaatimukset voidaan huomioida.

3.2.1 Vaatimuksien syntyminen

Vaatimusmäärittelyprosessi alkaa vaatimuksien syntymisellä. Vaatimuksenkeruu menetelmillä voidaan kerätä tietoa kvalitatiivisesti tai kvantitatiivisesti. (Cadle et al. 2014, s. 64) Kvantitatiivinen menetelmä tarkoittaa määrällisen tiedon keräämistä. Kummallakin tiedonkeruu menetelmällä voidaan saada mahdollisimman kattava kuva tarvittavista järjestelmän ominaisuuksista.

Kvalitatiivisia menetelmiä ovat esimerkiksi haastattelut, työpajat, prototyypit ja tarkkailut. Haastattelu on perinteisin tapa saada tietoa, mutta myös edelleen tärkeimpiä menetelmiä laadullisen tiedon saamiseksi. Työpajoihin osallistuvat tärkeimmät sidosryhmät ja siellä sidosryhmät käyvät keskustelua keskenään. Työpajoissa on oltava ohjaaja mukana, jotta keskusteluissa tulee jokaisen sidosryhmän edustajan tarpeet esille. (Cadle et al. 2014, s. 65) Ohjaajan on myös ohjattava keskusteluja, jotta keskustelut pysyvät olennaisissa puheenaiheissa. Nuseibehin ja Easterbrookin (2000) mukaan prototyyppien avulla voidaan saada alustava kuva kehitettävästä järjestelmästä ja niiden avulla voidaan saada palautetta sidosryhmiltä. Hahmottuminen helpottuu käyttäjien lisäksi myös suunnittelijoille. Cadlen et al. (2014) mukaan muu kvalitatiivinen menetelmä olisi esimerkiksi tarkkailu. Tarkkailujen avulla suunnittelijat voivat ymmärtää paremmin käyttäjien tarpeista. Tarkkailussa voi ilmetä uusia vaatimuksia, jota käyttäjät eivät ole itsekään tajunneet ilmaista. (Cadle et al. 2014, s. 65) Kvalitatiivisten menetelmien avulla saadaan käyttäjien käytännön ongelmia hahmotettua ja sen perusteella voidaan dokumentoida vaatimuksia paremmin.

Kvalitatiivisten menetelmien lisäksi voidaan käyttää kvantitatiivisia menetelmiä. Cadlen et al. (2014) mukaan kvantitatiivisia menetelmiä ovat esimerkiksi kyselyt, dokumenttianalyysit, toimintojen seuraaminen ja rekistereiden läpikäyminen. Kyselyiden avulla saadaan välitöntä tietoa sidosryhmien vaatimuksista, riippumatta heidän maantieteellisestä sijainnistaan. (Cadle et al. 2014, s. 66–67) Kyselyistä saatava tieto voi olla kuitenkin harhaanjohtavaa esimerkiksi huonojen vastausprosenttien tai huonojen kysymysten takia. Olemassa olevien dokumenttien analyysillä pyritään esimerkiksi ymmärtämään mitä dokumentteja on olemassa, minkälaista tietoa järjestelmän tulee tallentaa ja kenellä on oikeus käsitellä tietoihin. Toimintojen seuraamisella voidaan saada yleinen käsitys, mitä järjestelmän toimintojen tulisi sisältää ja mikä on eri työntekijöiden rooli liittyen tulevaan järjestelmään. Lisäksi olemassa olevien rekistereiden läpikäymisen avulla voidaan saada

paljon määrällistä tietoa, esimerkiksi kuinka paljon laskutetaan kuukaudessa. (Cadle et al. 2014) Rekistereiden läpikäymisen perusteella voidaan ymmärtää, millaista dataa järjestelmä käsittelee ja miten dataa tulisi hyödyntää.

3.2.2 Vaatimuksien analysoiminen

Alustavien vaatimuksien löytämisen jälkeen vaatimuksia voidaan analysoida. Vaatimuksia arvioidaan eri kriteerien avulla. Jos kriteerit eivät täyty, ongelmia todennäköisesti ilmenee myöhemmissä järjestelmän kehitysvaiheissa. (Cadle et al. 2014, s. 69–71) Analysointivaiheessa on tärkeintä erotella, mitä vaatimuksia otetaan huomioon ja mitä ei.

Analysoinnissa vaatimuksia luokitellaan esimerkiksi toiminnallisiin ja ei-toiminnallisiin vaatimuksiin. Niitä voidaan kategorisoida myös eri näkökulmiin, esimerkiksi liiketoiminnan tai sidosryhmien näkökulmaan. Vaatimuksia arvioidaan oleellisuuden kannalta, sillä niiden on sisällyttävä projektin laajuuteen. Muita kriteereitä vaatimuksien oleellisuuden kannalta ovat esimerkiksi saavutettavuus, testattavuus, jäljitettävyyys ja johdonmukaisuus. (Cadle et al. 2014, s. 68–71) Suunnittelijan on arvioitava näitä kriteereitä tarkkaan, jotta projektin laajuus ei paisu liian suureksi.

Vaatimuksien priorisointi on yksi tärkeimpiä analysointimenetelmiä, sillä resurssien rajoittuneisuuden vuoksi kaikkia vaatimuksia ei voida huomioida halutulla tasolla (Cadle et al. 2014, s. 71). Eberleinin (2003) mukaan priorisoidut vaatimukset auttavat päätöksentekoa esimerkiksi tilanteissa missä on karsittava vaatimuksia pois ajan puutteen takia. Priorisointiin osallistuvat sekä asiakkaat että suunnittelijat. Asiakkaat voivat ehdottaa suunnittelijoille heidän mielestensä hyödyllisimpiä ominaisuuksia ja ne priorisoidaan tärkeysjärjestykseen. Suunnittelija voi samanaikaisesti tuoda esille ominaisuuksien teknisiä riskejä tai muita vaikeuksia, jolloin asiakas voi arvioida uudelleen tärkeysjärjestystä. Lisäksi suunnittelija voi esittää omia ehdotuksia, jolloin priorisointia tehdään vuorovaikteisesti. (Eberlein 2003) Priorisointia on tärkeä tehdä tiiviissä yhteistyössä, jotta kumpikin näkökulma tulee huomioitua.

Muita analysointimenetelmiä on esimerkiksi mallintaminen ja JAD-menetelmä (*Joint Application Development*). Suosittuja mallintamistekniikoita ovat esimerkiksi data-flow mallit ja oliopohjaiset lähestymistavat. Toinen menetelmä, JAD, on käytännössä johdettu ryhmätapaaminen, missä käydään keskustelua järjestelmän ominaisuuksista. Tapaamisiin osallistuu johtajia, projektipäälliköitä, käyttäjiä, järjestelmäasiantuntijoita ja suunnittelijoita. (Eberlein 2003) Ryhmätapaamisissa on mahdollisuus selvittää mahdollisia konfliktitilanteita eri sidosryhmien kesken ja käydä yleistä keskustelua järjestelmän ominaisuuksista.

3.2.3 Vaatimuksien vahvistaminen

Vaatimuksien vahvistamisen tarkoituksena on hyväksyä määritellyt vaatimukset (Eberlein 2003). Se on määrittelyvaiheen viimeinen vaihe, jolloin arvioidaan ja vahvistetaan dokumentoitu materiaali (Cadle et al. 2014, s. 72).

Cadlen et al. (2014) mukaan arviointiin kuuluu kaksi osa-aluetta: tarkistaminen ja vahvistaminen. Tarkistusvaiheessa varmistetaan, että dokumentointi on tehty oikealla tavalla (Cadle et al. 2014, s. 72–73). Oikea tapa viittaa siihen, että dokumentti on ymmärrettävä sekä selkeästi tehty. Cadlen et al. (2014) mukaan dokumentit tulisi laatia mahdollisimman selkeästi, jotta suunnittelijat ja muut sidosryhmät voivat ymmärtää helposti vaatimuksia. Tarkistamisen jälkeen dokumentit vahvistetaan. Vahvistamisen vaiheessa verrataan, että dokumentin sisältö on täsmällinen käyttäjien tarpeiden kannalta (Cadle et al. 2014, s. 72–73). Vahvistaminen on kriittinen vaihe, sillä virheellinen dokumentaatio voi suunnata tietojärjestelmän kehittämisen väärään suuntaan.

Dokumentoinnin perusteella voidaan kommunikoida kerättyjä vaatimuksia eri sidosryhmien kanssa (Nuseibeh & Easterbrook 2000). Eberleinin (2003) mukaan hyvä dokumentti ei jätä mitään tulkinnan varaan. Dokumentin on oltava ymmärrettävä sekä yhtenäinen. Suunnittelijoiden näkökulmasta dokumentin sisällön on oltava saavutettavissa, jotta järjestelmä voidaan toteuttaa realistisesti. (Eberlein 2003) Dokumentteja on tallennettava turvallisesti ja selkeästi, jotta niitä voidaan löytää myöhemmissä ohjelmistokehityksen vaiheissa. Dokumenteista pitäisi ottaa varmuuskopiot, jotta tiedot ovat saatavilla aina tarvittaessa.

3.2.4 Dokumentointi ja hallinta määrittelyn tukitoimintoina

Dokumentointi on vaatimusmäärittelyn tukitoiminto, sillä dokumenttien avulla voidaan pitää kirjaa kerätyistä vaatimuksista sekä esittää ne selkeästi ymmärrettävässä muodossa (Cadle et al. 2014, s. 73). Dokumenttien tarkoitus on varastoida kerättyä tietoa, jotta niihin voidaan tukeutua myöhemmin. Dokumentointia tehdään koko määrittelyprosessin aikana. Niihin voidaan luoda sanasto termeille, joita lukija voisi tarvita ymmärtääkseen vaatimuksia paremmin (Cadle et al. 2014, s. 74). Dokumentoinnissa voidaan käyttää graafisia esityksiä tai kaavioita kuvaamaan vaatimuksia (Koivula 2014). Visuaaliset lisäykset auttavat lukijoita hahmottamaan kompleksisia vaatimuksia paremmin.

Muutoksien kannalta vaatimuksia tulee myös hallita kokonaisuutena (Cadle et al. 2014, s. 76). Hallinnan tarkoitus on kerätä, varastoida sekä jakaa tietoa (Eberlein 2003). Vaatimuksia on pystyttävä jäljittämään, jos tulee tarve selvittää niiden alkuperä (Cadle et al. 2014, s. 77). Jäljittämistä voidaan tehdä esimerkiksi versionhallinnalla, mihin tallennetaan eri versioita dokumenteista (Eberlein 2003). Muutoksia on myös hallittava, jotta järjestelmäprojektin laajuus ei kasva liian suureksi. (Cadle et al. 2014, s. 76) Laajuutta voi

olla vaikea hallita, sillä muuttuvien vaatimuksien seurauksena on vaikea hahmottaa, milloin laajuus paisuu liian suureksi. Ketterissä menetelmissä vaatimuksia kerätään iteratiivisesti, jolloin muutoksenhallinnan merkitys kasvaa. Vaatimuksien jatkuva hallinta on kriittinen toimenpide, koska on tärkeä tietää mitkä tiedot ovat oleellisia milläkin hetkellä (Schön et al. 2017).

3.3 Sidosryhmät

Tietojärjestelmiin liittyy useita sidosryhmiä. Sidosryhmillä on omat tavoitteet liittyen suunniteltavaan järjestelmään, joten eri sidosryhmien vaatimukset vaihtelevat. (Nuseibeh & Easterbrook 2002) Esimerkiksi rahoittajilla voi olla vaatimus, että järjestelmä suunnitellaan mahdollisimman pienellä budjetilla. Sen sijaan käyttäjien vaatimuksena on mahdollisimman hyvä käyttäjäkokemus. Käyttäjäkokemukseen panostaminen vaatii paljon resursseja, joten rahoittajien ja käyttäjien tavoitteet ovat ristiriitaiset. Mitä enemmän sidosryhmiä on kyseessä, sitä vaikeammaksi toteuttaminen muuttuu. Aina kaikkia vaatimuksia ei ole mahdollista huomioidakaan (Haikala ja Mikkonen 2011, s. 155). Vaatimusmäärittelyssä on löydettävä sopiva tasapaino, jotta tärkeimmät sidosryhmät otetaan huomioon riittävällä tasolla.

Haikalan ja Mikkosen (2011) mukaan sidosryhmistä tehdään tietojärjestelmien kehitysprosessissa sidosryhmäanalyysi. Analyysistä voidaan selvittää, kehen kyseinen projekti vaikuttaa. Tietojärjestelmän sidosryhmiin kuuluu esimerkiksi toimittaja, tilaaja, käyttäjät ja viranomaiset. Tietojärjestelmäprojekteissa usein määritetään ohjausryhmä, johon kuuluu jokaisesta sidosryhmästä edustajia. (Haikala & Mikkonen 2011, s. 153–155) Ohjausryhmän tarkoitus on seurata projektin etenemistä sekä vahvistaa keskeisiä päätöksiä (Haikala & Mikkonen 2011, s. 153). Ohjausryhmä toimii projektiryhmän, sidosryhmien sekä asiakkaan välillä.

Vaatimusmäärittelyssä sidosryhmien tunnistaminen on kriittistä (Nuseibeh & Easterbrook 2000). Käyttäjät ovat yksi tärkeimpiä sidosryhmiä, sillä he ovat järjestelmän välittömiä hyödyntäjiä. Nuseibeh ja Easterbrookin (2000) mukaan käyttäjät ovat interaktiivisen järjestelmän suunnittelussa merkittävässä roolissa. Käyttäjiä voi olla monen tasoisia: aloittelijoita, kokeneita tai satunnaisia käyttäjiä. (Nuseibeh & Easterbrook 2000) Käyttäjäkeskeisellä suunnittelulla voidaan parantaa käyttäjien saamaa kokemusta järjestelmän käytöstä. Aihetta käsitellään tarkemmin luvussa 4.

4. KÄYTTÄJÄKESKEINEN TIETOJÄRJESTELMIEN SUUNNITTELU

Öbrandin (2019) mukaan tietojärjestelmillä on iso vaikutus yrityksen toiminnan tehostamiseen ja ne ovat strategisesti merkittäviä nykypäivän organisaatioissa. Järjestelmien avulla liiketoiminnan toimintoja voidaan organisoida paremmin. Viime vuosina tietojärjestelmät ovat kehittyneet merkittävästi, mutta haluttuja tavoitteita ei olla saavutettu. (Öbrand 2019) Nykypäivänä tietojärjestelmät ovat suurimmaksi osaksi ohjelmistoja, jonka takia ohjelmistotuotannon menetelmiä voidaan soveltaa tietojärjestelmien suunnitteluun.

Ketterät suunnittelumenetelmät ovat suosittuja nykypäivän ohjelmistotuotannossa. Koivulan (2014) mukaan ketteriä ohjelmistomenetelmiä kehitettiin, jotta voitaisiin huomioida vaatimuksien muutoksia paremmin. Ketterissä menetelmissä tehdään jatkuvasti yhteistyötä käyttäjien kanssa ja heiltä palautetta iteratiivisesti. Standardin ISO 9241-210 (2010) mukaan käyttäjäpalautte on tärkeimpiä lähteitä suunnittelussa. Käyttäjien palautteen avulla voidaan arvioida suunnitteluratkaisuja todellisten tilanteiden perusteella. (ISO 9241-210, 2010a) Käyttäjiltä selvitetään liiketoiminnallisia tarpeita ja vaatimuksia, jonka perusteella järjestelmäratkaisut luodaan. (Kirsch 2002)

Tietojärjestelmiä kehitetään asiakkaille hankkeina. Haikalan ja Mikkosen (2011) mukaan järjestelmäprojekteja voidaan kutsua hankkeiksi, koska hankkeet muodostuvat joukoista ideoita, joilla voidaan täyttää strategisia tavoitteita. Hankkeesta vastaa omistaja ja hän on yleensä yksi yrityksen johdon edustajista. Hankkeen toteutuksesta vastaavat hankkeen päällikkö sekä projektiryhmä. (Haikala & Mikkonen 2011, s. 32–33) Projektiryhmä on sidoksissa ohjausryhmään jatkuvasti ja heiltä projektiryhmä saa palautetta hankkeen etenemisestä.

4.1 Käyttäjäkokemuksen merkitys

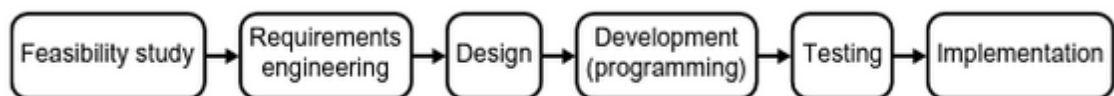
Käyttäjäkeskeisyys tarkoittaa suunnittelussa käyttäjien tavoitteiden ja tarpeiden asettamista keskiöön. Käyttäjäkeskeistä suunnittelua ohjaa jatkuva arviointi loppukäyttäjän näkökulmasta sekä iteratiiviset suunnittelumenetelmät. (Brhel 2015) Käyttäjäkeskeisyyden vaikuttaa vahvasti käyttäjäkokemus. Käyttäjäkokemukseen on perinteisesti ajateltu liittyvän laitteen käytettävyyden, mutta käsitteenä se on paljon laajempi (Sutcliffe 2009). Buschmannin (2011) mukaan käyttäjäkokemus käsittää ihmisen odotukset, käytön aikaiset reaktiot ja käytön jälkeiset reaktiot laitteen käyttämisestä. Järjestelmien suunnittelussa pyritään hyödyntämään käyttäjäkeskeisiä menetelmiä, jotta saavutetaan ensiluokkainen käyttäjäkokemus. (Buschmann, 2011)

Käyttäjäkokemus syntyy ohjelmiston ja laitteiston ulkoisesta olemuksesta, järjestelmän suorituskyvystä ja käyttöä avustavista ominaisuuksista (SFS- EN ISO 9241-210, 2010a). Sutcliffeen (2009) mukaan käyttäjäkokemukseen vaikuttaa käytettävyyden taso, esteettisyys ja hedonismi. Hedonismi on filosofinen näkemys, missä nautinto motivoi ihmisen käyttäytymistä (Veenhoven 2003). Esteettisyyteen ja hedonismiin voidaan vaikuttaa esimerkiksi järjestelmän graafisella käyttöliittymällä. Graafisen käyttöliittymän kautta käyttäjä hyödyntää järjestelmää käytännössä. Käyttöliittymät pyritään suunnittelemaan mahdollisimman mieleisiksi käyttäjille. Esimerkiksi sopivilla värivalinnoilla voidaan vaikuttaa järjestelmän luottavuuteen ja mielekkyyteen. Jos käyttöliittymä on vaikeakäyttöinen ja epämiellyttävä, käyttäjä voi turhautua järjestelmän käytöstä.

Käyttäjäkokemukseen panostaminen on ensisijaisen tärkeää suunnittelussa. Huono käyttäjäkokemus aiheuttaa pitkäaikaisesti negatiivisia tunteita tuotetta kohtaan. Järjestelmätöimittäjien on huomioitava käyttäjäkokemus suunnittelussa, jotta asiakkaita ei menetetä pysyvästi. Käyttäjäkokemuksen parantamiseksi ISO (International Organization for Standardization) on julkaissut standardin, jossa on lueteltu yleisiä käyttäjakeskeisiä suunnittelun periaatteita. Käyttäjää tulisi standardin mukaan ottaa kehitysprosessiin mukaan, sillä he tarjoavat arvokasta tietoa eri tilanteista ja tarpeista (SFS- EN ISO 9241-210, 2010a). Standardia käsitellään tarkemmin alaluvussa 4.3.

4.2 Tietojärjestelmien elinkaaren vaiheet

Ohjelmistoprojektit ovat tyypillisesti pitkäaikaisia hankkeita, jotka voivat kestää useita vuosia (Haikala & Mikkonen 2011, s. 33). Järjestelmän kehittämisen prosessiin kuuluu kuusi vaihetta. Tietojärjestelmän kehittämisen vaiheet ovat esitetty kuvassa 1:



Kuva 1: Tietojärjestelmän suunnittelun vaiheet (Cadle et al. 2014, s. 8)

4.2.1 Esiselvitys

Cadlen (2014) mukaan tietojärjestelmän suunnittelu alkaa esiselvityksellä. Esiselvityksessä selvitetään, onko järjestelmän toteuttamiselle tarvetta. (Cadle et al. 2014, s. 9) Esiselvityksen tehtävä on etsiä paras ratkaisu teknologisiin, organisatorisiin ja taloudellisiin ongelmiin (Koivula 2014). Organisatorisia ongelmia, kuten liiketoimintaprosessien tehotomuutta, voidaan tehostaa uudella järjestelmällä. Järjestelmällä voidaan esimerkiksi parantaa tiedon jakamisen mahdollisuutta. Prosessien tehokkuudella voidaan tehdä merkittäviä taloudellisia säästöjä, jos järjestelmiä hyödynnetään maksimipotentialiinsa. Mitä kompleksisemmasta järjestelmästä on kyse, sitä todennäköisemmin esiselvityksen vaiheeseen kuluu enemmän aikaa (Cadle et al. 2014, s. 9). Uudella teknologialla voidaan

korvata vanhentuneita järjestelmiä. Esiselvityksessä on tarkoitus kiinnittää huomioita olemassa oleviin ongelma-kohtiin ja siten lähteä selvittämään ratkaisua ongelmiin.

4.2.2 Vaatimusmäärittely

Esiselvityksen jälkeen siirrytään vaatimusmäärittelyihin. Kuten luvussa 3 käsiteltiin, vaatimusmäärittelyn tarkoituksena on tukea ymmärrystä siitä, miten suunniteltava järjestelmä tukee asiakkaan liiketoimintaa. Cadlen (2014) mukaan vaatimuksia täytyy selvittää, analysoida ja dokumentoida. Vaatimusdokumentin avulla voidaan vahvistaa, että suunniteltavassa järjestelmässä on huomioitu kaikkien sidosryhmien vaatimukset riittävällä tasolla. Koska vaatimusdokumentti toimii ohjenuorana suunnittelulle, on tärkeää, että vaatimuksia voidaan jäljittää alkuperäiseen lähteeseen jokaisessa suunnittelun vaiheessa. Suunnitteluun vaikuttaa vaatimuksien keräämisen tarkkuus. Mitä tarkemmin määritelmät on kirjoitettu, sitä vähemmän joutuvat suunnittelijat arvailemaan. (Cadle et al. 2014, s. 9) Selkeiden vaatimusdokumenttien avulla suunnittelijoiden työ helpottuu.

4.2.3 Suunnittelu

Vaatimuksien dokumentoinnin jälkeen järjestelmälle suunnitellaan alustavia ratkaisuja ja mitataan niiden yhteensopivuutta vaatimuksien kanssa. Kun alustava ratkaisu on valittu, voidaan siirtyä itse järjestelmän toteutukseen. Toteutusvaiheessa laitteistot ja ohjelmistot joko luodaan itse tai hankitaan toimittajalta. Jos järjestelmä on jo olemassa, sitä voidaan myös konfiguroida. (Cadle et al. 2014, s. 9) Suunnitteluratkaisuissa käytetään malleja ja kaavioita kuvaamaan ratkaisuja karkeammalla tasolla (Pahl 2004). Malleissa voidaan kuvata esimerkiksi, miten tietorakenteet muodostuvat järjestelmässä tai miten vuorovaikutus tapahtuu käyttäjän ja laitteen välillä (Pahl 2004). Mallien on yleisesti tarkoitus helpottaa järjestelmän hahmottamista.

Haikalan ja Mikkosen (2011) mukaan suunnittelussa keskeisintä on ohjelmiston arkkitehtuurin suunnittelu. Arkkitehtuurisuunnittelun tarkoitus on tehdä tekninen määritelmä järjestelmän aikaisemmista vaatimusdokumenteista. Kommunikoitavuus on erittäin tärkeää tässä vaiheessa, jotta jokainen kehitystyöhön osallistuva ymmärtää mistä on kyse. Arkkitehtuuri jaetaan pienempiin toteutettaviin komponentteihin, johon määritellään eri rajapintoja. Arkkitehtuurisuunnittelun jälkeen voidaan jakaa ohjelmisto vielä pienempiin ja yksityiskohtaisempiin komponentteihin, jolloin niitä voidaan toteuttaa ohjelmoiden. (Haikala & Mikkonen 2011, s. 177–179) Mitä paremmin arkkitehtuurisuunnitelmat on dokumentoitu, sitä helpompi on suunnittelijan aloittaa järjestelmän toteutusta.

4.2.4 Testaus

Testausvaihe on yksi tärkeimmistä vaiheista tietojärjestelmän kehittämisessä. Cadlen et al. (2014) mukaan testaamisen avulla voidaan varmistaa järjestelmän riittävä toimivuus

ja huomata mahdolliset virheet. Vaikka testausvaihe on kehityksen viimeisimpiä vaiheita, sitä ei tulisi laiminlyödä. (Cadle et al. 2014, s. 9) Epäonnistuneissa tietojärjestelmähankkeissa testaus usein jää hyvin vähälle huomiolle, sillä kiireen ja myöhäisten aikataulujen takia testausvaiheesta joustetaan (Kuutila, 2019). Testaamisen avulla voidaan selvittää mahdollisia epäselvyyksiä, jotka voivat olla suunnittelijan mielestä loogisia, mutta käyttäjälle eivät ole selviä.

Testaamisen vaiheita ovat:

- Testaamisen suunnittelu
- Testiympäristön luominen
- Testin tekeminen
- Tulosten katselu ja arvioiminen (Haikala & Mikkonen 2011, s. 205).

Aluksi testaamisessa suunnitellaan Haikalan ja Mikkosen (2011) mukaan testiprosessi sekä testiympäristö. Suunnitelman avulla voidaan jäljittää, millaisilla testaustavoilla virhe löytyi. Testaamisen tarkoitus on löytää ohjelman koodipohjasta tai ohjelman suorittamisesta ilmeneviä virheitä (Haikala & Mikkonen 2011, s. 205). Ohjelman käyttöliittymästä voidaan etsiä virheitä esimerkiksi kokeilemalla eri linkkejä ja katsomalla toimivatko ne. Haikalan ja Mikkosen (2011) mukaan virheiden jäljitystä kutsutaan englanniksi termillä ”*debugging*”. Testaus on kriittinen järjestelmäkehityksen vaihe, sillä aikataulullisesti onnistuneissa projekteissa se kuluttaa jopa yli puolet ohjelmistoprojektien resursseista. (Haikala & Mikkonen 2011, s. 205) Asiakkaan tyytyväisyyden kannalta on järjestelmän toimivuus tärkeää, minkä takia resurssien kuluminen testaukseen on perusteltua. Käytännössä järjestelmien testaaminen jää liian vähäiselle huomioille, kun aikatauluihin ei pystytä sitoutumaan (Kuutila 2019).

4.2.5 Käyttöönotto

Viimeisenä tietojärjestelmä otetaan käyttöön todellisessa ympäristössä, kun järjestelmä on testattu ja hyväksytty (Cadle et al. 2014). Todellisessa ympäristössä tulee huomioida asioita, mitä testausympäristössä ei ole. Esimerkiksi Cadlen (2014) mukaan olemassa olevan infrastruktuurin huomioiminen on tärkeätä. Jos aikaisemmin on käytetty toista järjestelmää, on oltava huolellinen muutosvaiheessa. Muutoksesta aiheutuvat häiriöt ja katkokset voivat aiheuttaa asiakkaalle taloudellisia menetyksiä, jonka takia muutosprosessi kannattaisi ajoittaa hiljaisemmalle ajanjaksolle kuten viikonlopulle. (Cadle et al. 2014) Lisäksi järjestelmän käyttöönotossa täytyy huomioida muiden järjestelmien sulava integroiminen. Käyttöönoton vaihe tulee suunnitella tarkkaan, jotta voidaan minimoida ylimääräisiä ongelmatilanteita, joita muutoksen tai integroimisen yhteydessä voi tulla.

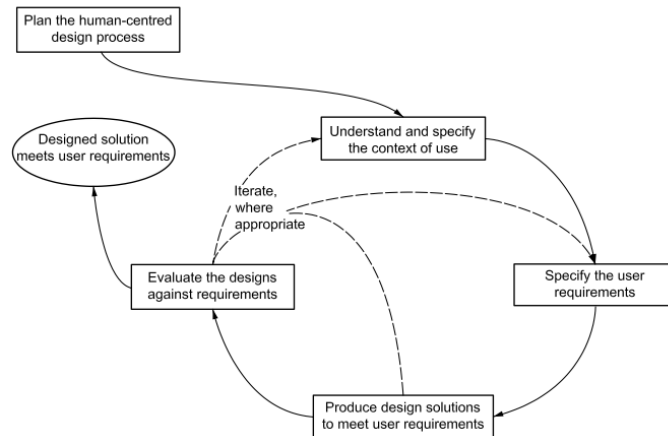
4.3 Ihmisen ja järjestelmän vuorovaikutuksen ergonomia

Standardi SFS- EN ISO 9241-210 käsittelee ihmisen ja järjestelmän vuorovaikutuksen ergonomiaa, mikä käytännössä sisältää ihmiskeskeisen suunnittelun periaatteita. Ihmiskeskeisyyttä voidaan ajatella synonyymina käyttäjäkeskeisyyden kanssa, koska käyttäjät ovat ihmisiä. Standardin mukaan käyttäjiä tulisi ottaa suunnitteluun mukaan, sillä käyttäjiltä voidaan kerätä arvokasta tietoa, mitä suunnittelijat eivät tiedosta. Käyttäjäpalaute on ensisijainen tietolähde ihmiskeskeiseen suunnitteluun. Ihmiskeskeisen suunnittelun on noudatettava monia toimintaperiaatteita, jotta se saavuttaa riittävän tason. Periaatteita ovat:

- Prosessin iteratiivisuus
- Selkeä ymmärrys käyttäjistä, toiminnoista ja ympäristöstä
- Suunnittelussa otetaan käyttäjät mukaan koko prosessin ajan
- Suunnittelun pohjana toimii käyttäjäkokemus kokonaisuudessaan
- Suunnittelua ohjaa ja tarkentaa käyttäjäkeskeinen arviointi
- Suunnittelussa on oltava monialaisia osaajia. (SFS- EN ISO 9241-210, 2010a).

Iteratiivinen työskentelytapa sallii muutoksien tekemisen pitkän suunnitteluprosessin. Iteratiivisuudella huomioidaan käyttäjiä paremmin. Iteratiivisuus antaa myös suunnittelijoille mahdollisuuden huomata omat virheensä. Virheitä voi syntyä, jos ei ole tehty riittävästi pohjatyötä käyttäjien kanssa. Suunnittelijoiden on ymmärrettävä, millaisessa kontekstissa käyttäjät toimivat. Käyttäjiä tulisi ottaa suunnittelussa alusta saakka mukaan ja heidän kanssaan pitäisi tehdä yhteistyötä läpi prosessin. Prosessin aikana tulisi tehdä jatkuvaa käyttäjäkeskeistä arviointia, jotta voidaan minimoida virheiden määrää. Käyttäjäkeskeisellä arvioinnilla päästään todennäköisimmin haluttuun lopputulokseen, kun käyttäjien vaatimukset ja tarpeet asetetaan prioriteetiksi. Monialaisilla osaajilla voidaan varmistaa järjestelmän onnistumisen mahdollisuutta, kun eri suunnittelunäkökulmat otetaan huomioon.

Kuten aikaisemmin mainittiin, ihmiskeskeinen suunnitteluprosessi on iteratiivinen. Kuvasta 2 nähdään ihmiskeskeisen suunnittelun prosessin kulku:



Kuva 2: Ihmiskeskeisen suunnittelun prosessi (SFS- EN ISO 9241-210, 2010b).

Kuvan 2 mukaan ensimmäisenä tehdään alustava suunnitelma prosessista. Suunnitelman valmistuttua voidaan aloittaa ensimmäinen iteratiivisen prosessin vaihe: kontekstin ymmärtäminen. Kontekstin ymmärtämisen jälkeen tehdään käyttäjien vaatimusten määrittely. Määrittelyn jälkeen tuotetaan alustava ratkaisu järjestelmästä ja ratkaisua arvioidaan vaatimuksiin peilaten. Näitä vaiheita iteroidaan niin kauan, kunnes saadaan ratkaisu, joka täyttää käyttäjän vaatimukset riittävällä tasolla.

5. VAATIMUKSIEN TOTEUTUMISEN HAASTEET KÄYTTÄJÄKESKEISESSÄ SUUNNITTELUSSA

Ohjelmistotekniikassa perusongelmana on täyttää järjestelmän vaatimusten ja käytettävän teknologian välistä kuilua (Haikala & Mikkonen 2011, s. 177). Suunnittelija tekee ratkaisuja oman kokemuksen, ohjelmistonkehitysmenetelmien ja oman luovuuden avulla. (Sutcliffe 2002, s. 19–43) Suunnittelijoiden oma kokemus ja luovuus ei kuitenkaan ole riittävää ratkaisujen tekemisessä, sillä asiakkaan vaatimuksia pitäisi ottaa paremmin huomioon.

5.1 Vaatimusmäärittelyn haasteet

Vaatimukset eivät ole aina itsestään selviä. Muutoksenhallinta on yksi vaatimusmäärittelyn haaste (Nusebeih & Easterbrook 2000). Suunnitteluprosessin iteratiivisuus voi aiheuttaa paljon muutoksia, ja sen myötä on pystyttävä hallitsemaan näitä muutoksia. Muutoksien aiheuttamaa hankkeen laajuuden kasvamista on myös hankala hallita. Muutoksien laajuudelle on löydettävä tasapaino, sillä muutoksia ei voida tehdä loputtomasti.

Sidosryhmien aiheuttama kompleksisuus tuo vaikeuksia vaatimusmäärittelyyn (Bevan & Maguire 2002). Määrittäjöille on haasteellista priorisoida vaatimuksia optimaalisella tavalla. Kompleksisten järjestelmien kehittämisessä suunnittelijat joutuvat nojautumaan vain sidosryhmien antamiin tietoihin, sillä suunnittelijoilla ei todennäköisesti ole resursseja perehtyä monenlaisiin toimialoihin kovinkaan syvällisesti. Sidosryhmien kasvavat tarpeet vaikeuttavat suunnittelijoiden tehtäviä (Pahl 2004). Suunnittelijoiden on tehtävä itsenäisesti päätöksiä, mitä vaatimuksia karsitaan.

Vaatimusmäärittelyn vaikeimpia haasteita ovat äänettömien vaatimusten selvittäminen (Cadle et al. 2014, s. 64). Käyttäjillä on usein olemassa tarpeita kehitettävään järjestelmään, joita he eivät ymmärrä (Bevan & Maguire 2002). Tarpeita voi olla vaikea ilmaista selkeästi tai käyttäjän tarpeet voivat olla epämääräisiä (Nusebeih & Easterbrook 2002). Myös käyttäjien tarpeita ei osata dokumentoida oikeassa muodossa, minkä seurauksena käyttäjäkeskeisyys kärsii suunnittelussa (Bevan & Maguire 2002). Käyttäjien vaatimusten epämääräisyys voi johtua muista tekijöistä, joita ei voida hallita (Nusebeih & Easterbrook 2002). Ohjelmistojen kompleksisuuden takia käyttäjät eivät välttämättä osaa kuvitella vaatimuksiaan, jonka takia niitä on vaikea ilmaista ääneen.

5.2 Vaatimuksien toteutumisen haasteet

Vaatimuksien yhteensovittaminen käyttäjäkeskeiseen suunnitteluun ei ole yhtä helppoa kuin miltä se kuulostaa. Järjestelmät epäonnistuvat, koska kehityksessä suunnittelijat eivät ymmärrä käyttäjien tarpeita riittävällä tasolla (ISO 2010a). Goldsmithin (2004) mukaan teknisten ihmisten ja käyttäjien kommunikointi on yksi merkittävin ongelma vaatimuksien toteutumisessa. Suunnittelijoiden hankalat asenteet voivat vaikuttaa suunnitteluun, sillä heitä ei kiinnosta kaupallinen aspekti suunniteltavasta tuotteesta. (Goldsmith 2004). Suunnittelijoilta puuttuu kokonaisvaltainen ajattelutapa järjestelmän vaikutuksesta ja laajuudesta. Suunnittelijat kuvittelevat tietävänsä käyttäjien vaatimuksista, minkä seurauksena tietojärjestelmäprojektit eivät onnistu (Goldsmith 2004). Tietojärjestelmän kehittämisessä on vaikeuksia keskittyä kokonaiskuvaan, kun kompleksisista järjestelmistä kyse (Schön et al. 2017).

Yeon (2002) mukaan suurin osa projekteista epäonnistuu, koska alkuperäisesti määritettyyn projektin budjettiin tai aikatauluun ei pystytä sitoutumaan. Alamin (2016) artikkelissa sanottiin, että vuonna 2012 tehdyn tutkimuksen mukaan vain 16,2 % tietojärjestelmäprojekteista toimitettiin onnistuneesti vaatimusten, aikataulun ja budjetin suhteen. Yli puolet projekteista olivat osittain epäonnistuneita ja 31 % olivat täysin epäonnistuneet. (Alami 2016) Onnistuneiden projektien osuus on suhteellisesti hyvin pieni verrattuna epäonnistuneisiin projekteihin.

Yksi syy epäonnistumiselle oli järjestelmien yhteensopimattomuus käyttäjien vaatimusten kanssa (Jiang & Klein 2000). Joskus ongelmat kasvavat niin suuriksi, että projekteja jätetään kesken. Projektien hylkääminen aiheuttaa merkittäviä määriä kustannuksia organisaatioille (Yeo 2002). Projekteissa ei arvioida tarpeeksi realistisesti niiden saavutettavuutta. Lisäksi suunnittelun ongelmana ovat liian nopeat kehityksen syklit, missä aikaa ei anneta tarpeeksi käyttäjien tarpeiden analysoimiseen (Bevan & Maguire 2002). Tarpeiden selvittämiseen ei käytetä tarpeeksi aikaa, jonka seurauksena järjestelmät eivät saavuta tavoitteitaan.

Tietojärjestelmäprojektien ongelmat eivät synny teknisistä syistä vaan enemmänkin sosiaalisista tai poliittisista syistä (Yeo 2002). Alami (2016) väitti projektien epäonnistuvan, koska projektiekosysteemit eivät ole tasapainossa. Huonot johtamistaidot vaikuttavat järjestelmien epäonnistumiseen. Johtajien projektinhallinta on heikkoa, vaikka projektinhallinnan oppeja on dokumentoitu ja helposti saatavilla. (Alami 2016) Projektipäällikkö on vastuussa hankkeen toteutumisesta, joten hänen tulisi huolehtia projektiryhmän suoriutumisesta. Jiang ja Kleinin (2000) mukaan ongelmia syntyy myös projektiryhmän roolien hahmottamisessa, sillä projektiryhmissä ei määritetä tarpeeksi selkeästi henkilöiden rooleja. Lisäksi yleisen asiantuntemuksen puute vaikuttaa projektin tehokkuuteen negatiivisesti. (Jiang & Klein 2000)

Organisaatiokulttuuri vaikuttaa voimakkaasti halukkuuteen jakaa tietoa ja ylipäättänsä asenteeseen liittyen tietojärjestelmiin (Dorothy & Kayworth 2006). Esimerkiksi asiakashallintajärjestelmissä on tarkoitus selvittää sekä jakaa tietoa asiakkaista. Myyjien palkkaus perustuu provisiopalkkaukseen, jolloin heillä voi olla kannusteita pitää asiakastietoja itsellään. Vaikka myyjät ovat käyttäjiä, heillä on selkeä kannustin olla jakamatta tietoa asiakashallintajärjestelmään. Organisaation sisäisesti pitäisi välttää tämän kaltaisia tilanteita esimerkiksi vaihtamalla palkitsemistapoja.

Käyttäjäkokemuksen puute vaikuttaa tietojärjestelmän onnistumiseen (Jiang & Klein 2000). Sutcliffen (2009) mukaan huonosti käytettävä järjestelmä jää käyttäjien muistiin negatiivisessa merkityksessä. Aikaisemmin kokeiltuja, huonosti käytettäviä tuotteita asosioidaan tulevaisuudessa näihin huonoihin kokemuksiin ja niitä voi olla vaikea muuttaa (Sutcliffe 2009). Käyttäjätyytyväisyyttä tulisi huomioida, sillä käyttäjät ovat lopulta järjestelmän hyödyntäjiä.

6. YHTEENVETO

6.1 Päätelmät

Kuten tutkimuksen alussa oletettiin, tietojärjestelmien suunnittelussa on vaikeuksia yhteensovittaa asiakasvaatimuksia. Tietojärjestelmien suunnittelussa on vaikeuksia pysyä alkuperäisessä budjetissa ja aikataulussa. Ylittyminen johtuu enimmäkseen uudelleensuunnittelun takia. Myös vaatimusten riittämättömästä määrittelystä seuraa epäonnistuneesti suunniteltuja järjestelmäratkaisuja. Suunnittelijat usein kuvittelevat tietävänsä asiakkaiden tarpeista, vaikka käytännössä totuus on aivan toisenlainen. Suurin osa tietojärjestelmäprojekteista jätetään kesken tai epäonnistuvat alusta saakka, koska suunnittelua ei tehdä oikein.

Ennen tietojärjestelmien suunnittelua tehdään selvitys asiakkaan vaatimuksista. Koska kaikkia vaatimuksia ei voida huomioida, on kiinnitettävä paljon huomiota vaatimusten analysointivaiheeseen. Vaatimuksia on priorisoitava tärkeysjärjestykseen ja niistä on otettava vain olennaisimmat huomioon, jotta laajuus ei kasva liian suureksi. Kasuvat sidosryhmien tarpeet tuovat haasteita suunnitteluun.

Nykypäivänä tuotekehityksessä asiakkaan tarpeiden täyttäminen sekä arvon toimittaminen ovat nousseet tärkeimmiksi näkökannoiksi liiketoiminnassa (Schön et al. 2017). Käyttäjäkokeumuksella voidaan vaikuttaa asiakkaan tyytyväisyyteen ja sillä suuri merkitys järjestelmän menestymiseen. Käyttäjakeskeisen suunnittelun pohjalla toimivat käyttäjien vaatimukset, jonka takia vaatimusmäärittely on ehdottoman tärkeää suunnittelussa (Bevan & Maguire 2002). Tietojärjestelmähankkeet ovat merkittäviä investointeja asiakkaille, minkä takia suunnittelun tulisi onnistua. Tietojärjestelmän suunnittelussa pitäisi olla käyttäjät keskiössä, jotta saadaan maksimoitua järjestelmän hyöty.

6.2 Jatkotutkimusehdotukset

Tässä tutkimuksessa saadaan yleinen käsitys asiakasvaatimusten yhteensovittamisen haasteista käyttäjakeskeisessä suunnittelussa. Jatkotutkimuksissa on tarve käyttää ajankohtaisempia lähteitä, sillä suurin osa tämän tutkimuksen tutkimusaineistosta on julkaistu 2000-luvun alussa. Jatkotutkimuksiin on tarve lisätä tarkempia käyttäjakeskeisiä menetelmiä, jos niitä on olemassa. Käyttäjakeskeisestä suunnittelusta tulisi verrata ketterien ohjelmistokehitysmenetelmien piirteisiin, sillä ketterät menetelmät ovat nykypäivänä suosittuja ohjelmistotuotannossa. Jatkotutkimuksissa tulisi myös esitellä ratkaisuja löydettyihin haasteisiin.

LÄHTEET

Alami, A. (2016). *Why Do Information Technology Projects Fail?* Saatavilla [www-osoitteessa: https://www.sciencedirect.com/science/article/pii/S1877050916322918](https://www.sciencedirect.com/science/article/pii/S1877050916322918). Viitattu 2.3.2019.

Bevan, N. (2009). *What is the difference between the purpose of usability and user experience evaluation methods?* Allen institute for artificial intelligence. Saatavilla [www-osoitteessa: https://pdfs.semanticscholar.org/cba7/4036995821ca560d31bf397c695a460a63a5.pdf](https://pdfs.semanticscholar.org/cba7/4036995821ca560d31bf397c695a460a63a5.pdf). Viitattu: 2.2.2019.

Bevan, N. & Maguire, M. (2002). *User requirements analysis*. Saatavilla [www-osoitteessa: https://link.springer.com/chapter/10.1007/978-0-387-35610-5_9](https://link.springer.com/chapter/10.1007/978-0-387-35610-5_9). Viitattu 17.2.2019.

Brhel et al. (2015). *Exploring principles of user-centered agile software development: A literature review*. Saatavilla [www-osoitteessa: https://www.sciencedirect.com/science/article/pii/S0950584915000129](https://www.sciencedirect.com/science/article/pii/S0950584915000129). Viitattu 25.2.2019

Buschmann, F. (2011). *Unusable Software Is Useless, Part 1*. Vol. 28. Saatavilla [www-osoitteessa: https://search-proquest-com.lib-proxy.tuni.fi/docview/818397850/fulltextPDF/C2F2C9D5DC904794PQ/1?accountid=14242](https://search-proquest-com.lib-proxy.tuni.fi/docview/818397850/fulltextPDF/C2F2C9D5DC904794PQ/1?accountid=14242). Viitattu: 2.2.2019.

Cadle, J., Ahmed, T., Cox, J., Girvan, L., Paul, A., Paul, D., Thompson, P. (2014). *Developing information systems: practical guidance for IT professionals*. E-kirja. BCS Learning & Development Limited. Saatavilla [www-osoitteessa: https://ebookcentral.proquest.com/lib/tampere/detail.action?docID=1713962](https://ebookcentral.proquest.com/lib/tampere/detail.action?docID=1713962). Viitattu: 24.2.2019.

Dorothy E., Kayworth T. (2006). *Review: a review of culture in information systems research: toward a theory of information technology culture conflict*. Saatavissa [www-osoitteessa: https://dl.acm.org/citation.cfm?id=2017316](https://dl.acm.org/citation.cfm?id=2017316). Viitattu 3.3.2019

Eberlein A., Maurer F., Paetsch F. (2003). *Requirements engineering and agile software development*. MIS Quarterly. Vol. 30. Saatavilla [www-osoitteessa: https://ieeexplore.ieee.org/abstract/document/1231428](https://ieeexplore.ieee.org/abstract/document/1231428). Viitattu 2.3.2019.

Gable, G., Sedera D., Chan T. (2008). *Re-conceptualizing information system success: The IS-impact measurement model*. EBSCO Industries. Vol. 9. Saatavilla [www-osoitteessa: http://eds.b.ebscohost.com/abstract?site=eds&scope=site&jrnl=15369323&AN=34101995&h=ZmguvF-deCF%2fkLXylcmkqK4yJJsmwV5gThC3WNUvCzWb6rtu8k%2beiqUbtpOR7e6mDlu3OBNw%2bu7rAUDLJdU9Ghg%3d%3d&crl=f&resultLocal=ErrCrlNoResults&re](http://eds.b.ebscohost.com/abstract?site=eds&scope=site&jrnl=15369323&AN=34101995&h=ZmguvF-deCF%2fkLXylcmkqK4yJJsmwV5gThC3WNUvCzWb6rtu8k%2beiqUbtpOR7e6mDlu3OBNw%2bu7rAUDLJdU9Ghg%3d%3d&crl=f&resultLocal=ErrCrlNoResults&re)

sultNs=Ehost&crlhashurl=login.aspx%3fdirect%3dtrue%26profile%3de-host%26scope%3dsite%26authype%3dcrawler%26jrnl%3d15369323%26AN%3d34101995. Viitattu 26.2.2019.

Goldsmith, R. (2004). *Discovering real business requirements for software project success*. E-kirja. Artech House. Saatavilla www-osoitteessa: <https://ebookcentral.proquest.com/lib/tampere/reader.action?docID=227685>. Viitattu: 20.2.2019.

Haikala, I. & Mikkonen, T. (2011). *Ohjelmistotuotannon käytännöt*. 12. uud. p. Helsinki: Talentum.

Hoda, R. et al. (2018). *The Rise and Evolution of Agile Software Development*. IEEE Computer Society. Saatavilla www-osoitteessa: <https://ieeexplore-ieee-org.libproxy.tuni.fi/stamp/stamp.jsp?tp=&arnumber=8409911>. Viitattu: 2.2.2019.

Husnain, S. (2017). *Defining supplier relationship management system requirements for an EPC firm: a case study*. Diplomityö. Tampereen teknillinen yliopisto. Saatavilla osoitteessa: <https://dspace.cc.tut.fi/dpub/handle/123456789/25509>. Viitattu 1.2.2019.

Jiang, J., Klein, G. (2000). *Software development risks to project effectiveness*. Saatavilla www-osoitteessa: <https://www.sciencedirect.com/science/article/pii/S0164121299001284>. Viitattu 3.3.2019.

Kirsch L. J., Sambamurthy V., Ko D., Purvis R. L. (2002). *Controlling Information System Development Projects: The view from the client*. Vol. 52. Saatavilla www-osoitteessa: <https://pubsonline.informs.org/doi/abs/10.1287/mnsc.48.4.484.204>. Viitattu 4.3.2019.

Koivula, E. (2014). *Yliopistokirjastojen kirjastotietojärjestelmän vaatimusten kirjoitus ja priorisointi*. Diplomityö. Tampereen teknillinen yliopisto. Saatavilla www-osoitteessa: <https://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/22734/koivula.pdf?sequence=3&isAllowed=y>. Viitattu 5.3.2019.

Kuutila, M. et al. (2019). *Time pressure in Software Engineering: A Systematic Literature Review*. Cornell University. Saatavilla www-osoitteessa: <https://arxiv.org/abs/1901.05771>. Viitattu: 2.2.2019.

Nuseibeh B., Easterbrook, S. (2000). *Requirements Engineering: A Roadmap*. ICSE '00 Proceedings of the Conference on The Future of Software Engineering. Saatavilla www-osoitteessa: <https://dl.acm.org/citation.cfm?id=336523>. Viitattu: 20.2.2019.

Pahl, C. (2004). *Adaptive development and maintenance of user-centric software systems*. Vol. 46. Saatavilla www-osoitteessa: <https://www.sciencedirect-com.libproxy.tuni.fi/science/article/pii/S0950584904000722>. Viitattu 9.2.2019.

- Petter, S., DeLone, W., McLean, E. (2008). *Measuring information systems success: models, dimensions, measures, and interrelationships*. Taylor & Francis Group. Saatavilla [www-osoitteessa: https://orsociety.tandfonline.com/doi/abs/10.1057/ejis.2008.15#.XIJGCCgzY2w](https://orsociety.tandfonline.com/doi/abs/10.1057/ejis.2008.15#.XIJGCCgzY2w). Viitattu 8.3.2019.
- Schön et al. (2017). *Key Challenges in Agile Requirements Engineering*. Saatavilla [www-osoitteessa: https://link.springer.com/chapter/10.1007/978-3-319-57633-6_3](https://link.springer.com/chapter/10.1007/978-3-319-57633-6_3). Viitattu 24.2.2019.
- SFS-EN ISO 9241-210 (2010a). *Ihmisen ja järjestelmän vuorovaikutuksen ergonomia. Osa 210: Vuorovaikutteisten järjestelmien käyttäjäkeskeinen suunnittelu*. Suomen Standardisoimisliitto. Saatavilla [www-osoitteessa: https://online.sfs.fi/fi/index/hakutulos.html.stx](https://online.sfs.fi/fi/index/hakutulos.html.stx). Viitattu: 24.2.2019.
- SFS-EN ISO 9241-210 (2010b). *Ergonomics of human-system interaction. Part 210: Human-centred design for interactive systems (ISO 9241-210:2010)*. Saatavilla [www-osoitteessa: https://online.sfs.fi/fi/index/hakutulos.html.stx](https://online.sfs.fi/fi/index/hakutulos.html.stx). Viitattu: 24.2.2019.
- Sutcliffe, A. (2002). *User-Centered Requirements Engineering*. Springer. Saatavilla [www-osoitteessa: https://link.springer.com/content/pdf/10.1007%2F978-1-4471-0217-5.pdf](https://link.springer.com/content/pdf/10.1007%2F978-1-4471-0217-5.pdf). Viitattu: 20.2.2019.
- Sutcliffe A. (2009). *Designing for user engagement: aesthetic and attractive user interfaces*. Morgan & Claypool. Saatavilla [www-osoitteessa: https://tuni.finna.fi/Record/tamcat.531455](https://tuni.finna.fi/Record/tamcat.531455). Viitattu 3.3.2019.
- Thong, J., Hong, W., Tam, K.-Y. (2002). *Understanding user acceptance of digital libraries: what are the roles of interface characteristics, organizational context, and individual differences?* Vol. 57. Saatavilla [www-osoitteessa: https://www.sciencedirect.com/science/article/pii/S1071581902910244](https://www.sciencedirect.com/science/article/pii/S1071581902910244). Viitattu 8.3.2019.
- Veenhoven, R. (2003). *Hedonism and Happiness*. Journal of Happiness Studies. Vol. 4. Saatavilla [www-osoitteessa: https://link.springer.com/article/10.1023/B:JOHS.0000005719.56211.fd](https://link.springer.com/article/10.1023/B:JOHS.0000005719.56211.fd). Viitattu 8.3.2019.
- Yeo, K. (2002). *Critical failure factors in information system projects*. International Journal of Project Management. Vol. 20. Saatavilla [www-osoitteessa: https://www.sciencedirect.com/science/article/pii/S0263786301000758](https://www.sciencedirect.com/science/article/pii/S0263786301000758). Viitattu 26.2.2019.
- Öbrand L., Augustsson N.-P., Mathiassen L., Holmström J. (2019). *The interstitiality of IT risk: An inquiry into information systems development practices*. Information Systems Journal. Vol. 29.

Saatavilla www-osoitteessa: <https://www.scopus.com/record/display.uri?eid=2-s2.0-85042562406&origin=resultslist&sort=cp-f&src=s&st1=The+interstitial-ity+of+IT+risk%3a+An+inquiry+into+information+systems+development+prac-tices&st2=&sid=56524187832a241dcbcc0a663fef79f3&sot=b&sdt=b&sl=104&s=TI-TLE-ABS-KEY%28The+interstitiality+of+IT+risk%3a+An+inquiry+into+infor-mation+systems+development+practices%29&relpos=0&citeCnt=1&searchTerm=>. Viitattu 4.3.2019.