



Käyttöliittymä osana vaatimusmäärittelyä

Lauri Jussila

2020 Laurea



Laurea-ammattikorkeakoulu

Käyttöliittymä osana vaatimusmäärittelyä

Lauri Jussila
Tulevaisuuden innovatiiviset digi-
taaliset palvelut (YAMK)
Opinnäytetyö
Joulukuu, 2020

Lauri Jussila

Käyttöliittymä osana vaatimusmäärittelyä

Vuosi 2020

Sivumäärä 41

Vaatimusmäärittely on olennainen osa sovelluskehitystä. Vaatimusmäärittelyn tarkoitus on luoda sovelluksen toiminnasta ja ominaisuuksista kertova dokumentaatio. Käyttöliittymä on sovelluksen näkyvä osa käyttäjälle. Käyttöliittymä määrittää millaisen kokemuksen käyttäjä saa sovelluksesta.

Tämän opinnäytetyön tarkoituksena on tutkia vaatimusmäärittelyä osana mobiilisovellus projektia ja keskittyä siinä sovelluksen käyttöliittymän määrittelyyn. Opinnäytetyössä analysoidaan määrittelyä projektissa muodostuneita virheitä hyväksi käyttäen. Samalla tarkastellaan virheitä verraten niitä lopulliseen tuotokseen.

Tietoperustana opinnäytetyössä toimii vaatimusmäärittely. Tietoperustaa käsitellään artikkeleiden, tutkimusten ja kirjallisuuden avulla. Lisäksi tietoperustassa käsitellään käyttöliittymän suunnittelua. Tutkimusmenetelmänä opinnäytetyössä käytettiin suunnittelutieteellistä tutkimusta. Suunnittelutieteelliselle tutkimukselle ominaista on luoda suunnitelma ja tietämystä. Näitä suunnitelmia ja tietämys malleja hyödynnettiin opinnäytetyössä

Opinnäytetyön tuloksena luotiin näkemys mobiilisovellus projektin käyttöliittymän määrityksistä ja käyttöliittymän toteutuksesta. Opinnäytetyössä vertailtiin projektissa muodostuneita virheitä vastaavia määrityksiä vasten. Tämän perusteella toimeksiantajalle muodostuu selkeämpi kuva projektin määrityksistä ja toteutuksesta käyttöliittymää kohtaan. Samalla saadaan ymmärrystä tulevaisuuden projekteille käyttöliittymän määrittämisestä. Opinnäytetyössä tehtyjä havaintoja on mahdollista hyödyntää muissa toimeksiantajan projekteissa.

The requirement specification is an integral part of application development. The purpose of specifying the requirements is to create documentation about the operation and features of the application. The user interface is the part of the application visible to the user. The user interface determines what kind of experience the user has with the application.

This Master's thesis aims to examine the requirement specifications as part of a mobile application project and focuses on the application interface definition. The thesis analyzes the defects using the errors formed in the project. At the same time, the mistakes are assessed by comparing them to the final output.

The knowledge base used in the thesis is the specification of requirements. The knowledge base was addressed through articles, research, and literature. Also, the knowledge base deals with user interface design. The research method used in the thesis was design science research. Design research was characterized by the creation of a plan and knowledge. These plans and knowledge models were utilized in the thesis.

As a result of the thesis, a view was created about the mobile application project's user interface definitions and user interface implementation was created. The thesis compared the errors formed in the project against the corresponding determinations. The client gains a clearer picture of the project definitions and implementation towards the user interface. At the same time, an understanding is gained of how to define the user interface for future projects. It is possible to utilize the observations made in the thesis to benefit other projects of the client.

Sisällys

1	Johdanto	7
1.1	Kohdeyritys	7
1.2	Tutkimuksen tarkoitus ja tavoite	7
1.3	Tutkimusongelma	8
2	Tietoperusta	8
2.1	Vaatusmäärittely	8
2.1.1	Vaatimusten määrittelyn vaiheet	10
2.1.2	Vaatusmäärittelyn elinkaari	13
2.2	Vaatusmäärittely ohjaa käyttöliittymäsuunnittelua	17
2.3	Viitekehys mobiilisovelluksen suunnitteluun	18
3	Tutkimusmenetelmät ja toteutus	20
3.1	Suunnittelutieteellinen tutkimus	20
3.2	Viitekehys	20
3.3	Tutkimusviitekehys opinnäytetyössä	22
3.4	Tutkimustyön kuvaus	24
3.5	Tutkimusaineisto	25
3.5.1	Aineiston analysointi	26
3.5.2	Määritysten analysointi	27
3.5.3	Suunnittelun analysointi	29
3.5.4	Toteutuksen analysointi	30
4	Tulokset	31
4.1	Vaatimusten määrittäminen	31
4.2	Käyttöliittymäsuunnittelu	31
4.3	Tulosten synteesi	32
4.4	Tarkistuslista	32
5	Johtopäätökset ja pohdinta	35
5.1	Lähdekritiikki ja eettisyys	37
	Lähteet	38
	Kuviot	41
	Taulukot	41

1 Johdanto

Opinnäytetyö tehdään finanssi- ja vakuutusalan yritykselle. Opinnäytetyössä analysoidaan projektissa tehtyjä vaatimusten määrittämiä ja tutkitaan vaatimusmäärittelyn onnistumista. Yrityksen tavoitteena on tuoda uusia palveluita asiakkaiden käytettäväksi, tarjoamalla heille mobiilipalvelun. Yritys on aiemmin julkaissut mobiilisovelluksia asiakkailleen, mutta tämän opinnäytetyön kohteena olevan sovelluksen on tarkoitus olla aikaisemmat sovellukset yhdistävä palvelu.

Mobiilisovellukset ovat olleet osa jokapäiväistä elämää noin vuosikymmenen ajan. Mobiilisovellusten käyttöliittymät ovat kehittyneet jatkuvasti. Tutkimusta mobiilisovellusten käyttöliittymästä ja siihen liittyvästä määrittämisestä ei kovin laajasti käsitelty aihe. Hyvällä vaatimusmäärittelyllä vältetään virheiden määrää ohjelmistokehitys projekteissa. Ulkoasu on yleisesti suunnittelijan tuotos, jonka kehittäjä tuottaa sovellukseen. Tähän prosessiin on hyvä saada selkeyttä vaatimusmäärittelyn osuudesta osana sovelluskehityksen kokonaisprosessia.

1.1 Kohdeyritys

Kohdeyritys on tässä opinnäytetyössä suomalainen finanssi- ja vakuutusalan yritys. Yritys on tunnettu omalla alallaan ja sillä toimistoja ympäri Suomea. Yrityksen palveluita on kehitetty aktiivisesti viime vuosina ja yhtenä tähän kohdistuvana työnä oli mobiilisovelluksen julkaiseminen.

Opinnäytetyön tekijä työskenteli yhtiössä kyseessä olevan projektin jäsenenä. Pääasialliset tehtävät projektissa olivat osallistuminen vaatimusten määrittämiä koskeviin tehtäviin. Projektissa syntyneiden virheraporttien pohjalta syntyi idea tutkia käyttöliittymän määrittämisestä.

1.2 Tutkimuksen tarkoitus ja tavoite

Opinnäytetyössä on tarkoitus syventää ymmärrystä vaatimusmäärittelyn ja käyttöliittymäsuunnittelun merkitystä mobiilisovelluksen kehittämisprojekteissa. Tavoitteena opinnäytetyössä on kehittää ymmärrystä käyttöliittymän vaatimusmäärittämisestä. Opinnäytetyössä hyödynnetään kohdeyrityksen mobiilisovellus projektia ja siitä saatavia tuloksia. Opinnäytetyöllä koitetaan löytää ratkaisuja, miten käyttöliittymää tulisi määrittää. Sovelluksen käyttöliittymä on sen käyttäjälle näkyvä osa sovellusta. Toteutuksen kannalta on tärkeää luoda sovellus sen näköiseksi, kuin se on suunniteltu.

1.3 Tutkimusongelma

Tutkimuksessa käsitellään sovelluksen käyttöliittymään liittyvää vaatimusten määrittelyä. Tarkoitus on tutkia vaatimusmäärittelyä, joka liittyy sovelluksen käyttöliittymään projektissa. Tutkimuksessa käsitellään mobiilisovelluksen kehitystä ja siihen liittyvää käyttöliittymää. Projektin aikana käyttöliittymän kanssa havaittiin paljon virheitä, mistä syntyi idea projektille. Tutkimuksen ongelma viittaa näihin havaittuihin virheisiin. Tutkimukselle muodostui seuraavia apukysymyksiä:

- Mitä on vaatimusten määrittely
- Miten käyttöliittymän määrittelyä kannattaa toteuttaa

2 Tietoperusta

Tässä luvussa käsitellään opinnäytetyön tietoperustaa. Tietoperusta käsittelee vaatimusmäärittelyä sekä käyttöliittymän suunnittelua.

2.1 Vaatimusmäärittely

Vaatimusmäärittely on ohjelmistotuotannossa tärkein vaihe. (Ul-Arif ym. 2010, 41). Ohjelmistokehityksessä sovelluksen toimintaa kuvataan määrittelyillä, joilla kerrotaan sovelluksen toiminnasta. Määrittelyn tarkoituksena on selvittää sidosryhmien tarpeet ja dokumentoida ne. (Ul-Arif ym. 2010, 42). Vaatimusmäärittelyn prosessiin olennaisena osana kuuluvat seuraavat osuudet vaatimusten analysointi,

Vaatimuksella tarkoitetaan ulkopuolelta tulevaa tarvetta sovelluskehityksessä olevalle sovellukselle. Vaatimukset vaihtelevat tarpeita eivätkä ne ole aina yhtenäisiä. Vaatimukset eivät yksinään ole toimivia ja ne tarvitsevat tuekseen määrittelyä. (Heath 2020, 3). Ohjelmistokehityksessä sovelluksen toimintaa kuvataan määrittelyillä, joilla kerrotaan, miten sovelluksen tulee toimia. Määrittelyt ovat sovelluskehittäjälle ohjeet sovellusta varten. (Dooley 2011, 37). Järjestelmien kehityksessä vaatimukset ilman määrittelyä ovat hyödyttömiä. Yhdessä vaatimus määrittelyllä kyetään antamaan kehitettävänä olevalle järjestelmälle sen toiminnan kannalta merkittävät tiedot. (Heath 2020, 3-4).

Vaatimusten määrittely voidaan jakaa kolmeksi erilaiseksi osa-alueeksi käyttäjä vaatimukset, domain vaatimukset ja ei toiminnalliset vaatimukset. Käyttäjä vaatimuksia voidaan kutsua nimellä toiminnalliset vaatimukset. Toiminnalliset vaatimukset ovat loppukäyttäjälle näkyviä ja käytettäviä. Toiminnalliset määrittelyt kertovat miten lopputuotteeseen kuuluu toimia ottamatta kantaa toteutustapaan. Domain vaatimukset ovat yleisiin tai olemassa oleviin standardeihin liittyviä. Näitä ovat erilaiset standardit tai lait. Yleensä niihin viitataan määrittelyissä.

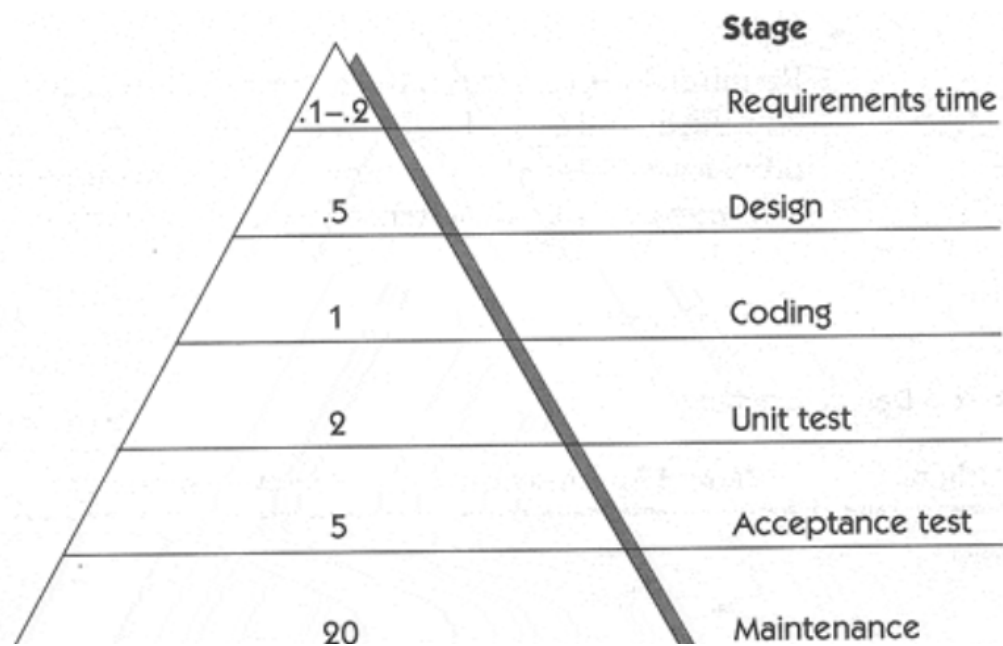
(Dooley 2011, 42). Ei toiminnallisilla määrityksillä puhuttaessa tarkoitetaan asioita, joita sovellus tarvitsee toimiakseen. Ne eivät ole välttämättä käyttäjälle näkyviä. (Dooley 2011, 43).

Leffingwell & Widrig ovat kuvanneet vaatimusten määrittelyä pyramidi mallina. Siinä sovelluksen tilaajan tarpeet kuvaavat pientä osaa vaatimusten määrittelystä. Hän nimeää huippua Ongelma tasoksi. Tässä on tarkoitus selvittää tilaajan haluamia toiminnollisuuksia. Kaksi seuraavaa tasoa sisältävät ominaisuudet, joita tilaajan tarpeista muodostuu. Tässä koostetaan laajemmiksi osiksi tarpeet, joita tilaajalla on esittänyt. Alinta tasoa edustaa vaatimusten määrittely. Tässä luodaan ominaisuuksista tarkemmat määritelmät, joilla saadaan luotua sovelluksen tilaajan tarpeita vastaava tuote (Leffingwell & Widrig 1999, 28-31). Kuva 1 esittää Leffingwall & Widrigin pyramidin.



Kuva 1: Leffingwell & Widrig pyramidi.

Tutkimuksen mukaan yleisimmät syyt sovellus kehityksen ongelmiin johtuvat määrittelyistä. Määritykset voivat olla kerätty vajavaisesti tai dokumentointi ei ole riittävällä tasolla. (Leffingwell & Widrig 1999, 17-18). Kuvassa 2 Leffingwell & Widrig esittää yksikkökustannuksen virheen korjaukselle virheen löydettyä eri vaiheessa ohjelmistokehitystä. Ylläpitovaiheessa korjauksen hinta on 20-kertainen kuin löydettyä ohjelmointi vaiheen aikana. Mitä aikaisemmin virheitä löydetään sovelluskehityksessä, pienenee niihin tarvittavat kustannukset. (Leffingwell & Widrig 1999, 20-21). Määrittelyiden virheiden osuuden voidaan arvioida olevan jopa 70 prosenttia uudelleen tehtävän työ osuudesta. Tyypillisesti projektin uudelleen tehtävän työ osuus voi olla jopa 50 prosenttia kaikesta työstä, voi määrittelyiden virheet kuluttaa projektin budjetista 25-40 prosenttia. (Leffingwell & Widrig 1999, 23.)



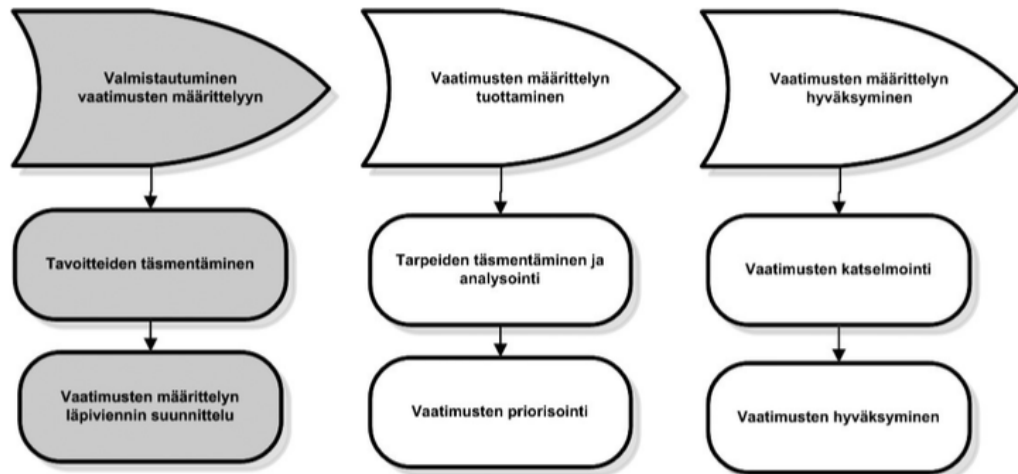
Kuva 2: Leffingwell & Widrigin malli (Leffingwell & Widrig 1999, 20.)

Määrittelyssä ei ole tarkoitus keskittyä kehitettävän kohteen toimintaan. Tarpeettomien käyttöliittymä, toteutus tai testaus tietojen määrittelyä on syytä välttää. Tällä tavoin vaatimukset keskittyvät järjestelmän käyttäytymiseen, (Leffingwell & Widrig 1999, 185.)

2.1.1 Vaatimusten määrittelyn vaiheet

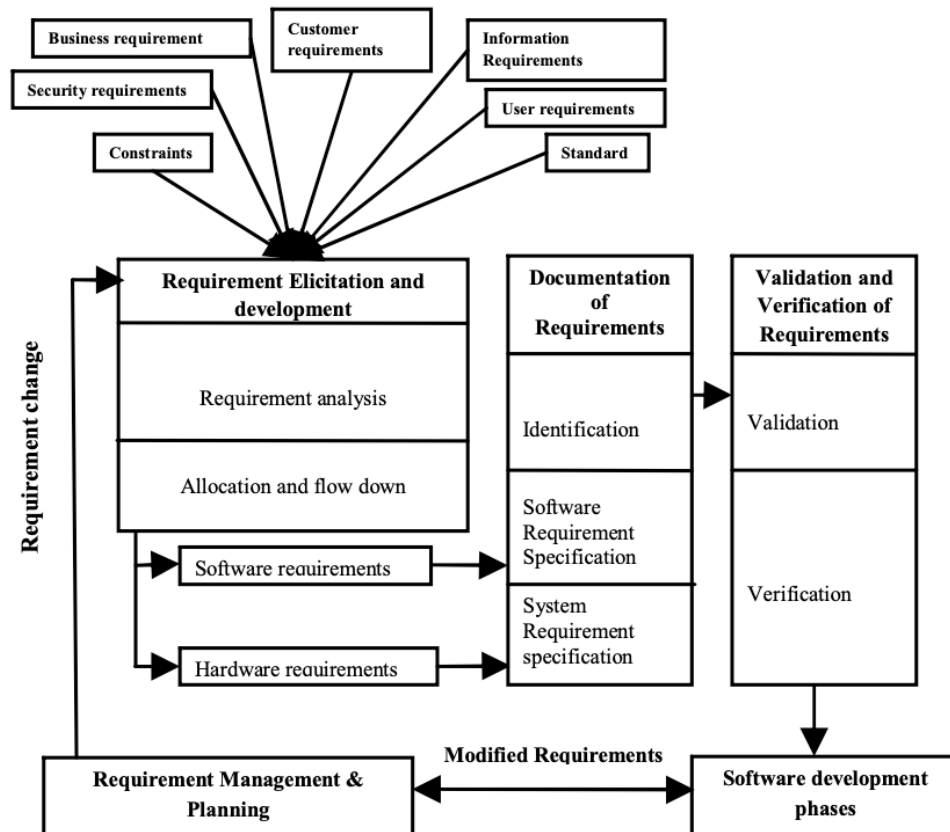
Vaatimusten määrittely on monivaiheinen prosessi. Siitä voidaan erottaa kolme pääaihetta valmistelu, määrittely ja määritysten hyväksyminen. Ensimmäisessä vaiheessa selvitetään tavoitteita, joita ollaan täyttämässä. Kehitettävän järjestelmän tunnistaminen. Työssä voidaan hyödyntää saatavilla olevaa dokumentaatiota. Vaatimustenmäärittely alkaa tavoitteiden täsmäntämisellä, jossa vaatimukseen vaikuttavia tekijöitä arvioidaan. Millaisilla menetelmillä työtä tehdään ja miten arvioidaan lopullisia tuloksia. Dokumentaation suunnittelu ja hyväksyminen osana kokonaisuutta. Läpiviennin suunnittelun lähtökohtana on luoda selvitys, miten määrittely tullaan toteuttamaan. (JUHTA 2018, 11-13). Toisessa vaiheessa oleellista on saada kerättyä informaatiota ja osallistuttua sidosryhmiä. Vaatimusten tarkoituksena on kattaa tulevat tarpeet ja löytää erilaiset riippuvuudet. Erilaisten toimintojen havaitseminen ja niiden jakaminen toiminnallisten ja ei-toiminnallisten vaatimusten välille. Hallinnoinnin kannalta vaatimusten priorisoinnilla on merkitystä. Toteutuksen ohjaus ja kulujen seuranta ovat edellytyksiä kehitystyössä. (JUHTA 2018, 14-15). Kolmannessa vaiheessa vaatimukset esitellään, niiden hyväksymistä varten. Vaatimusten katselmoinnin periaatteena on luotujen vaatimusten

saattaminen hyväksytyiksi. Oleellista on selventää vaatimukset ja korjata niissä havaitut virheet. (JUHTA 2018, 15-16). Kuva 3 esittää Juhta vaatimusmäärittelyn vaiheita



Kuva 3: Juhta vaatimusten määrittelyn vaiheet (JUHTA. 2018, 11.)

Kuvassa 4 esitetään mallia vaatimusten määrittelyn prosessista. Pandey, Suman ja Ramani esittävät tätä mallia vaatimusten määrittelyyn ketterissä ohjelmistoprojekteissa. Vaatimusten määrittely on jaettu pienempiin osakokonaisuuksiin. Kuvan avulla havainnollistetaan vaatimusmäärittelyn prosessi maisuutta ja yhteyttä ohjelmistokehitykseen. Vaatimusten vaiheet esitetään kuvassa tarkemmin asiakas toiveista sovelluskehitykseen saakka. Perinteinen vaatimusmäärittely tehdään projektin alussa, mutta ketterissä malleissa vaatimusten määrittäminen on osa toimintoja koko projektin ajan. (Pandey ym. 2010, 287-288).



Kuva 4: Vaatimusmäärittelyn prosessimalli (Pandey ym, 2010, 288.)

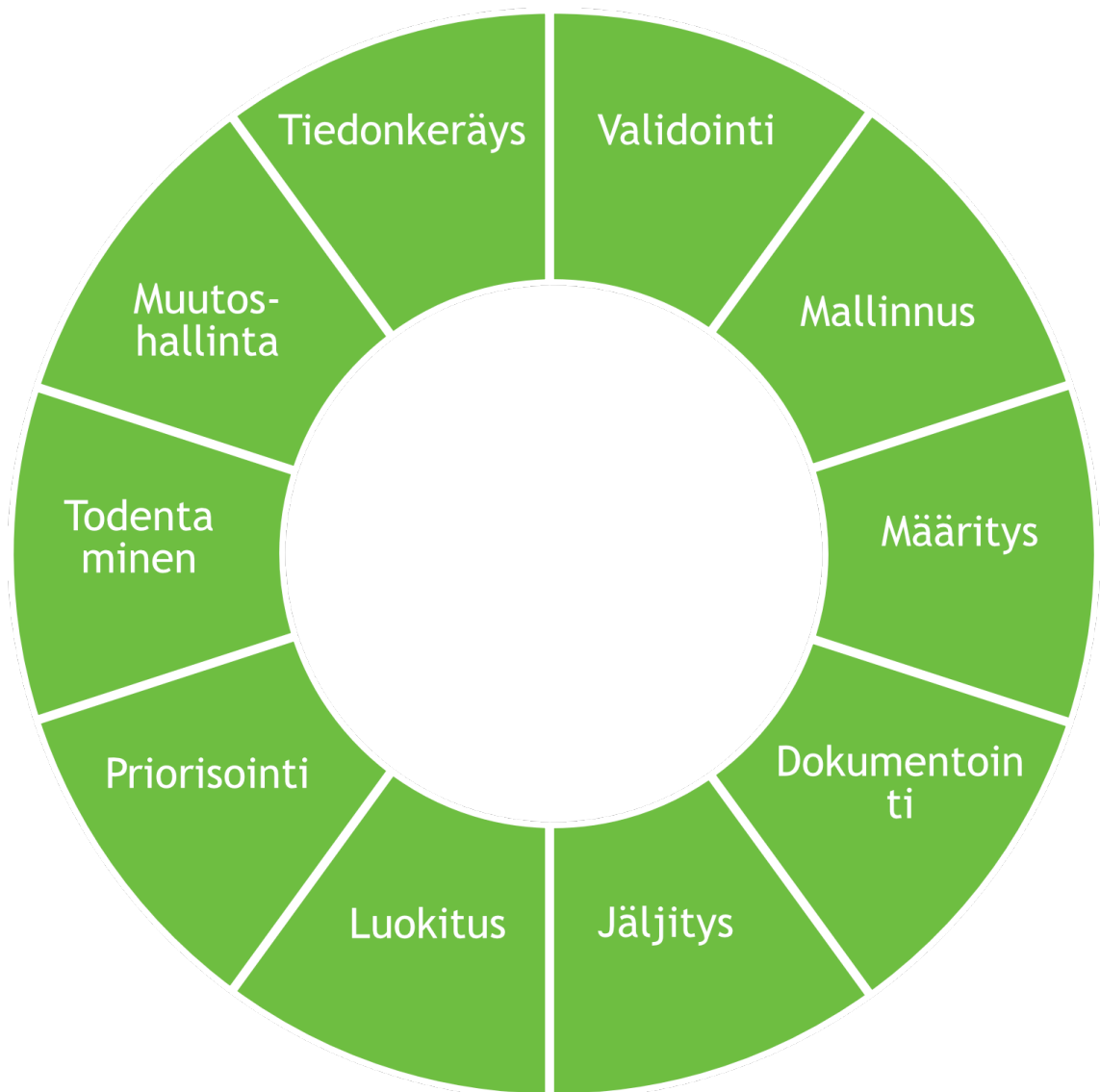
Perusominaisuuksia vaatimukselle Haikala ja Mikkola ovat tehneet neljän kohdan listan. Vaatimuksen tulee olla riittävänä tarkka ollakseen ymmärrettävä. Haikala ja Mikkola mainitsevat myös tarkan ja ymmärrettävän olevan ristiriidassa keskenään. Vaatimuksen pitää olla testattavissa. Jäljitettävyys osa vaatimuksen ominaisuuksista. Haikala ja Mikkola mainitsevat vaatimuksen oltava jäljitettävissä taaksepäin ja eteenpäin. Vaatimuksesta pitää ilmetä alkuperä sekä vaatimuksen teknisyytensä ja testitapaukset, joilla vaatimus on testattavissa. (Haikala & Mikkonen 2011, 63).

Haikala ja Mikkonen listaavat vaatimuksen dokumentoinnille seuraavat yleispiirteet. Vaatimus dokumentaatiolla tulisi olla luontipäivämäärä. Tekijän nimi pitäisi ilmetä dokumentaatiosta, vaatimuksen alkuperä keneltä vaatimus on saatu. Onko vaatimus uusi vai täydennetäänkö sillä jotain olemassa olevaa vaatimusta. Kuvaus vaatimuksesta apuna voidaan käyttää tekniikoita, joilla tarkennetaan kuvausta kuten käyttäjätarinat tai kaavioita. Miten vaatimus liittyy muihin vaatimuksiin. Mikä on vaatimuksen tarve perä, onko vaatimus valinnainen vai pakollinen. Vaatimuksen testattavuus, miten voidaan todentaa vaatimuksen testaaminen. Arvio tarvittavasta ajasta, joka vaatimuksen toteuttamiseen tarvitaan. (Haikala & Mikkonen 2011, 64).

Vaatimusten määrittelyn parantamiselle Niazi esittää viittä keskeistä prosessi aluetta. Tavoitteita asetetaan useasta eri syystä, niiden tarkoituksena on selkeyttää organisaatiolla. Tavoitteita tarvitaan vaatimusten järjestämisessä ja perustelun tukena. Vaatimusten ymmärtäminen voi olla hankalaa, mutta tarkalla vaatimusten asettamisella kyetään saamaan parempi organisaationallinen ymmärrys. (Niazi 2002, 127-128). Toisessa vaiheessa Niazi esittää vaatimus prosessin syklistyydestä. Kolmivaiheinen sykli, jossa vaatimukset selvitetään, vaatimukset analysoidaan ja vaatimusten arvioiminen. Lähestymisen kannalta Niazi esittää mallin sopivan suurien järjestelmien kehittämisessä. (Niazi 2002, 128). Kolmannessa vaiheessa Niazi esittää sidosryhmien osallistumista vaatimusten määrittelyn prosessiin. Tähän osallistettuna organisaatio on sitoutuneempi muutokseen ja edesauttaa uuden kehityksen läpi saattamista. (Niazi 2002, 128) Neljännessä vaiheessa Niazi selventää vaatimus määrityksen aikana kehitysprosessia vaatimukset eivät ole pysyviä. Muutoksia tapahtuu usein ja muutoksen hallinta on haasteellinen vaihe. Muutoksia pitää jatkuvasti kyetä dokumentoimaan, jotta kehitys prosessi olisi onnistunut. (Niazi 2002, 128). Viidennessä kohdassa Niazi toteaa suunnittelun olevan keskeisessä roolissa muutosprosessia. Niazi toteaa vaatimusten määrittelyn alkavan ilman erillistä suunnitelmaa. Tähän tulisi kiinnittää tarkemmin huomioita ja suunnitelmallisuutta. (Niazi 2002, 129).

2.1.2 Vaatimusmäärittelyn elinkaari

Kokonaisvaltaisen vaatimusten hallinnan tueksi Heath esittää vaatimusten elinkaaren hallinnan mallia. Vaatimusten elinkaaren hallinta antaa kokonaisvaltaisemman kuvan vaatimuksista. Mallia voidaan käyttää hyväksi vaatimus prosessin aikana. (Heath 2020, 152).



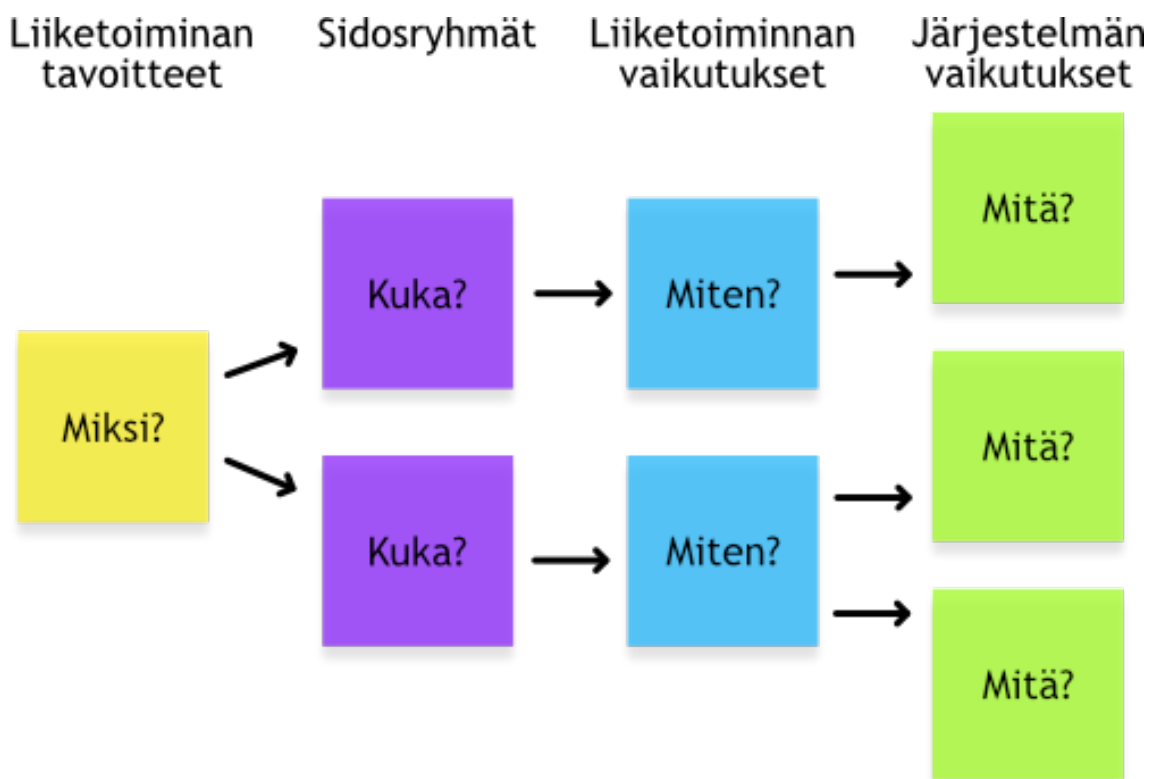
Kuva 5: Vaatimusmäärittelyn elinkaari

Kuvassa 5 esitetään vaatimusmäärittelyn elinkaari Heathin mallin mukaisesti. Hän jakaa vaatimusmäärittelyn 10 eri osuuteen. Vaatimusten validoinnilla varmistetaan vaatimusten olevan todenperäisiä ja täyttävän liiketoiminnan antama tarpeet. (Heath 2020, 153). Mallintaminen lähtee liiketoiminnan tarpeista. Nämä tarpeet esittävät sidosryhmät. (Heath. 2020, 153). Sidosryhmillä tarkoitetaan joukkoa, joka on vuorovaikutuksessa järjestelmän kanssa. Sidosryhmään voi kuulua myös henkilöitä, jotka eivät suoranaisesti ole vuorovaikutuksessa. Sidosryhmien roolit ovat osa kokonaisuutta. (Heath 2020, 5-6).

Heath esittää sidosryhmien vaikutusta vaatimusten mallintamisessa. Hän esittää vaikutus kartan, joka kuvaa vaikutukset nelitasoisessa kartassa. Tätä käyttämällä voidaan vastata perustava laatuksiin kysymyksiin järjestelmästä. Heath esittää seuraava kysymykset (Heath 2020, 20-21).

- Miksi rakennamme järjestelmän?
- Kuka hyötyy siitä?
- Miten sidosryhmät saavuttavat tavoitteet?
- Miten järjestelmä voi auttaa sidosryhmiä saavuttamaan tavoitteet?

Kuva 6 esittää vaikutus kartan Heathin perusteella. Vaikutus kartassa selvitetään liiketoiminnan tavoitteet (miksi) ja sidosryhmät (kuka). Sidosryhmien tunnistavat liiketoiminnalliset vaikutukset. Miten tämä vaikuttaa liiketoimintaan. Mitä tukevia toimenpiteitä organisaation on tehtävä tukeakseen liiketoimintaa. (Heath 2020, 21).



Kuva 6: Vaikutus kartan tarkoitus

Tämän mallin eduksi esitetään viittä kohtaa.

- Keskittyminen: Selkeisiin kysymyksiin vastaaminen
- Jäljitettävyys: Sidosryhmillä olevat tarpeet ja niiden täyttäminen
- Luotettavuus: Kartta on selkeä ja helposti esitettävä
- Yhteistyö ja vuorovaikutus: Jokaiseen tasoon liittyy ihmisten välistä vuorovaikutusta
- Strateginen visio: Ison kuvan hahmottaminen ja laajemman tiekartan mahdollistaminen

(Heath 2020, 21-22).

Vaatimusten validoinnilla varmistetaan vaatimusten täyttävän liiketoiminnalliset vaatimukset. Vaatimukset mahdollistavat liiketoiminnallisten tavoitteiden toteutumisen. (Heath 2020, 154). Määritykset koostuvat annetuista vaateista, jotka pitää pilkkoa pienemmiksi osiksi. Vaatimukset kerätään eri tiedonlähteistä ja muodostetaan yhdeksi yhtenäiseksi kokonaisuudeksi. (Heath 2020, 87). Tärkeintä määrityksissä on kyetä luomaan ymmärrettävä ja toteutettava kokonaisuus. Määrityksiä voidaan jaotella niiden tarpeen tai vaikutuksen mukaan. (Heath 2020, 156).

Dokumentointi on olennaista määrittelyssä. Muutoksien seuraaminen määrittelyssä edesauttaa palaamisessa edellisiin tilanteisiin. Määrittelylle oleellisten aineistojen viittaukset. (Heath 2020, 104). Dokumentointi tarjoaa selkeän näkyvyyden määrittelyihin. Dokumentointia ei ole rajattu pelkästään fyysisiin tulosteisiin, tärkeintä on vain sen olemassaolo ja saavutettavuus. (Heath 2020, 106).

Jäljitettävyydellä voidaan esittää mihin luotavalla vaatimuksella on liiketoiminnallinen vaikutus. Määritystä seuraamalla voidaan esittää vaikuttavuudet ja seuraukset. Samalla voidaan todeta rinnakkaisten määritysten yhtäläisyydet. (Heath 2020, 165).

Vaatimuksien luokituksia on monia ja siihen vaikuttaa vaatimuksen sisältö. Heath nostaa neljä pääkategoriaan joihin vaatimuksia voidaan luokitella. Liiketoiminnalliset määritykset ovat yleisesti tärkeimpiä määrityksiä. Näillä on selkeä hyöty tai tarve sovelluksen kehityksessä. Sidosryhmiltä tulevat vaatimukset Heat osoittaa toiseksi tärkeimmiksi. Ratkaisu määritykset sijoitetaan kolmannelle tasolle. Näillä mahdollistetaan sovelluksen rakentaminen. Siirtymä määrityksiä Heat kuvaa matalimmalla tasolla. Nämä eivät ole kriittisimpiä tehtäviä kehityksen kannalta. (Heath 2020, 156).

Priorisoinnille esitetään neljä pääkategoriaa. Vaatimukset, joiden tuottama arvo on suurin, arvostetaan pääsääntöisesti korkeammiksi. Vaatimusten kustannus on yksi tekijöistä, joka määrittää priorisointia. Kuinka paljon vaatimuksen toteuttaminen maksaa ja kuinka nopeasti vaatimus maksaa itsensä takaisin. Riskillä on merkitystä priorisoinnissa. Korkean riskin määritysten epäonnistuminen voi olla kehityksen kannalta kohtalokasta. Jotkin vaatimukset voivat olla merkitykseltään kiireellisempiä (Heath 2020, 157-158).

Vaatimusten todentamisella varmistetaan vaatimuksen täyttävän sille asetetut tavoitteet. Testauksen yhteydessä vaatimuksen tavoitteita verrataan toteutukseen. Tässä vaiheessa korostuu vaatimuksen dokumentaation laatu. Riippuen käytettävästä projekti mallista muutoksien tuottamiselle on erilaisia ratkaisuja. (Heath 2020, 159).

2.2 Vaatimusmäärittely ohjaa käyttöliittymäsuunnittelua

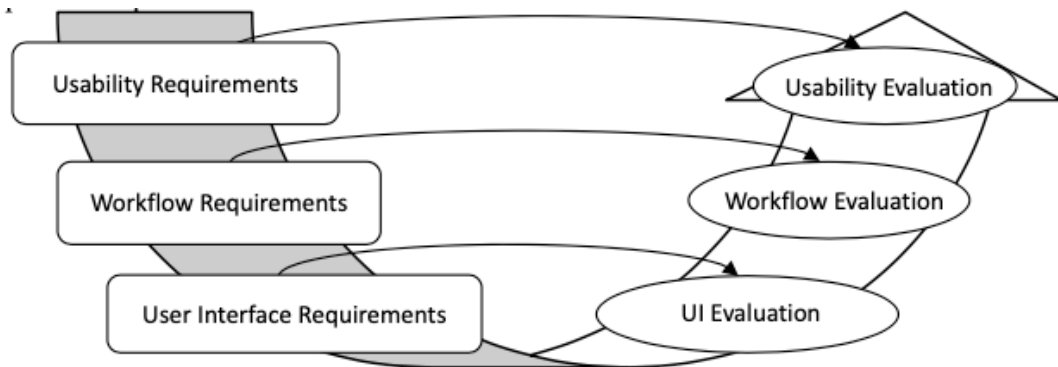
Käyttöliittymäsuunnittelun katsotaan kuuluvan ihmisen ja tietokoneen vuorovaikutusta käsittelevään tutkimusalaan. Tämän tieteen ala tutkii ihmisen ja tietokoneiden vuorovaikutusta. Käyttöliittymä on käyttäjälle näkyvä osa sovelluksesta. Käyttäjät pystyvät käyttöliittymän avulla suorittamaan toimintoja, joita sovellukselle on määritelty. Käyttöliittymällä on mahdollista syöttää tietoa sekä nähdä asioita. (Galitz 2007, 4). Käyttöliittymän suunnittelu on olennainen osa sovelluksen vaihetta, sillä se on käyttäjän näkymä sovellukseen. (Galitz 2007, 5).

Käyttöliittymien suunnittelun periaatteet ovat pääpiirteittäin samanlaisia. (Cooper 2014, 156.) Mobiili käyttöliittymien yleisin käyttötapa on koskettaa näyttöä sormilla. Tämä lähestyminen suunnittelussa on merkittävä eroavaisuus. (Cooper 2014, 433). Mobiilisovellukset poikkeavat verkossa käytettävistä sovelluksista, niiden tarjoamalla ominaisuuksillaan. Mobiilisovelluksen suunnittelussa on kuitenkin huomioitava aluskohtaiset yhteiset elementit, joita sovellukset jakavat. (Wasserman 2010, 3). Haasteensa suunnittelulle tuovat erilaisuudet kehitys ympäristöissä ja valmistajien laitteiden eroavaisuudet. (Min Tun 2014, 69). Mobiili ympäristössä on huomioitava operaattori, käytettävät laitteet, käyttöjärjestelmien versiot ja näytön koko. Nämä ovat muuttujia, joita suunnittelussa on ymmärrettävä. (Mendoza 2013, 8). Suunnittelussa on myös huomioitava loppukäyttäjät ja heidän tarpeensa. Käyttöliittymän suunnittelussa mobiililaitteille ylimääräisistä asioista tulisi karsia ja esittää käyttäjälle vain tarpeellinen informaatio. (McWherter & Gowell 2012, 90).

Käyttäjälähtöisellä suunnittelu metodeilla voidaan edesauttaa vaatimusmäärittelyä. Tässä voidaan hyödyntää tuloksia, joita saadaan käyttäjälähtöisistä tutkimuksista. Käyttäjälähtöinen suunnittelu prosessi sisältää neljä kohtaa. (Xiao 2020, 3).

- Käyttökontekstin määrittely
- Käyttäjien vaatimukset
- Suunnittelun määritelmät
- Suunnittelun arvioiminen

Kuvassa 7 esitetään Käyttäjälähtöisen suunnittelun ja suunnittelun hyväksynnän malli. Tässä mallissa on kuvattu käyttäjälähtöisen suunnittelun prosessia ja samalla hyväksyntä ketjun rakenne. Jokaiselle suunnittelu kohdalle on oma testausvaiheensa. Prosessin tarkoituksena on selkeyttää suunnittelun ja testauksen riippuvuutta. (Xiao 2020, 3-4).

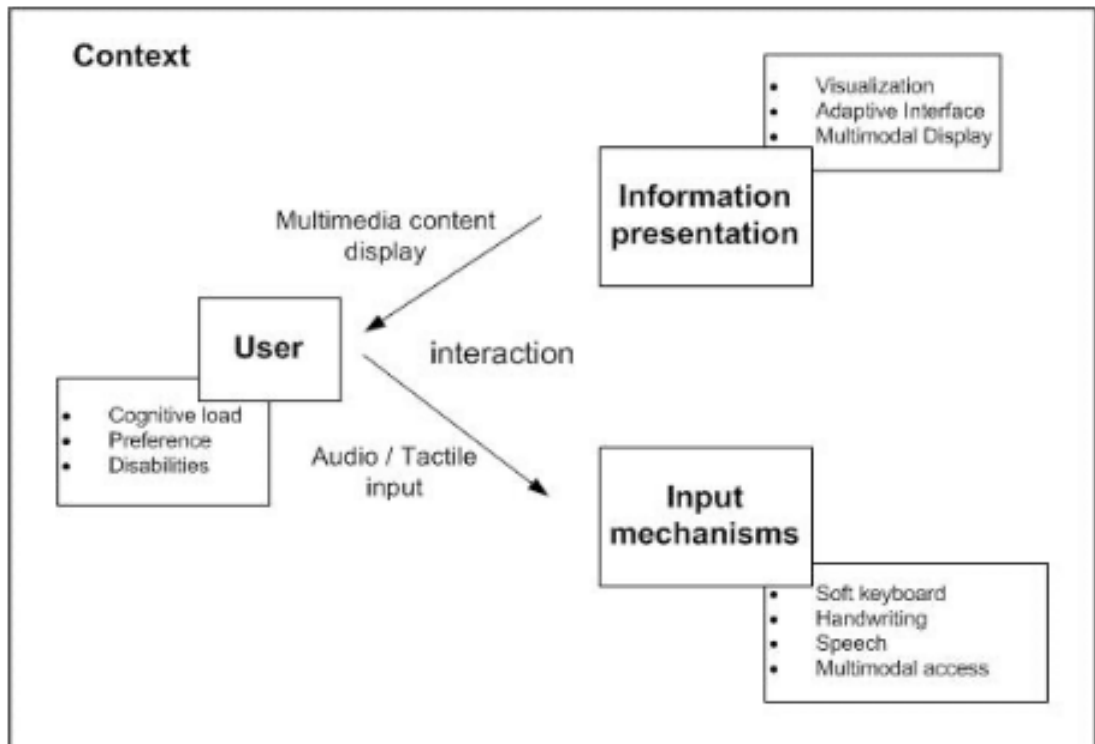


Kuva 7: Käyttäjälähtöisen suunnittelun prosessi (Xiao 2020, 3.)

Käytettävyysvaatimukset kuuluvat ei-toiminnallisiin vaatimuksiin. Niillä voi silti olla riippuvuuksia toiminnallisten vaatimusten kanssa. Näiden onnistumista voidaan mitata kyselyillä, asiakas testeillä sekä asiantuntijoilla arvioimalla. Työnkulun vaatimukset osoitetaan sovelluksen kykyä vastata käyttäjän tarpeita. Käyttötapausten avulla voidaan esittää vuorovaikutusta, jota käyttäjä kokee sovellusta käytettäessä. Käyttötapauksia kuvasta tulisi huomioida käyttäjä vaatimukset. Näitä tapauksia voidaan varmistaa käyttäjien kanssa lopullisen ymmärryksen saavuttamiseksi. (Xiao 2020,4). Käyttöliittymään vaikuttavat määritykset koostuvat käytettävyydestä ja työnkulun havaintojen perusteella, joita kerätään testauksella ja prototyyppien avulla. Käyttöliittymän määrityksillä sovelluksen käyttöliittymän toteutukseen annetaan raamit ja testauksessa voidaan varmistaa sovelluksen täyttävän sille asetetut vaatimukset. (Xiao 2020, 5).

2.3 Viitekehys mobiilisovelluksen suunnitteluun

Adipat ja Zhang esittävät mobiilisovellusten suunnittelun tueksi viitekehystä. He nostavat esille neljä kohtaa joihin suunnittelussa pitäisi keskittyä käyttäjä, sisältö, tiedon esittäminen ja sisällön syöttäminen. (Adipat & Zhang 2005, 2282.) Kuvassa 8 esitetään tarkemmin Adipat ja Zhang viitekehystä. Käyttäjä näkee esitetyn informaation ja pystyy reagoimaan siihen olemassa olevilla välineillä.



Kuva 8: Viitekehys mobiili käyttöliittymän suunnitteluun (Adipat & Zhang 2005, 2282.)

Käyttäjä on suunnittelussa keskiössä ja käyttäjän tarpeita tulisi ymmärtää suunnittelussa. Sisällön esittämisessä käyttäjälle pitäisi selvittää millaisia kohde käyttäjät mahdollisesti ovat ja selvittää heidän tarpeitaan. (Adipat & Zhang 2005, 2282). Adipat ja Zhang listaavat kolme käyttäjään vaikuttavaa asiaa

- Käyttäjäasetukset
- Kognitiivinen kuormitus
- Erityistarpeet

Suunnittelussa tulisi huomioida erilaiset käyttäjät ja tarpeet käyttäjien välillä. Tämän lisäksi osaamisen tasot saattavat myös vaihdella, joka on yksi huomioitava asia. Edistyneimmät käyttäjät kykenevät omaksumaankin muita helpommin uusia käyttöliittymiä. (Adipat & Zhang 2005, 2283). Käyttöliittymän selkeys helpottaa loppukäyttäjän navigointia sovelluksessa. Käyttöliittymän suunnittelussa huomioitavaa on tehdä toiminnot selkeiksi, ettei sovelluksen käyttämisestä tule käyttäjälle hankala toimenpide. (Adipat & Zhang 2005, 2284). Saavutettavuudella taataan sovelluksen mahdollinen käyttö erilaisille ryhmille. Adipat ja Zhang listaavat yleisimmät ryhmät saavutettavuuden piiristä sokeat, värisokeat, kuurot sekä huonokuuloiset ja liikuntarajoitteiset.

Pienien näyttöjen asettamat rajoitteet vaikeuttava suunnittelijoiden työtä. Sisältöjen suunnittelulla voidaan yksinkertaistaa tätä työtä. Adipat ja Zhangin esittävät sisällön esittämiseen kolme näkökulmaa. (Adipat & Zhang 2005, 2285).

- Visualisointi
- Mukautuva käyttöliittymä
- Multimodaalinen näyttö

Laajan sisällön mahduttaminen pieneen tilaan vaatii joustavuutta. Sisällön selaaminen pystysuunnassa on luottavaisempi asia kuin liikkua poikittain. Sisältö tekstien suunnittelussa otettava huomioon pienempien päätelaitteiden tarpeet. Sisällöt kannattaa tiivistää lyhyemmiksi kokonaisuuksiksi. (Adipat & Zhang 2005, 2285-2286). Adaptiivinen käyttöliittymä tarjoaa käyttäjälle mahdollisuuden omien mielenkiintojen nostamiseen esille. Tämän tyyppisille käyttöliittymille ominaista on hallittavuus, ennustettavuus ja läpinäkyvyys. Modaaliselle tiedon esittämiseksi pääsääntöiset tavat ovat visuaalinen esittäminen, ääneen perustuva ja kosketettava käyttöliittymä. (Adipat & Zhang 2005, 2287).

Sisällön syöttämiseksi vaihtoehtoja on monia. Kosketusnäytöillä toimivat näppäimistö, jotka mahdollistavat kirjoittamisen ilman fyysistä näppäimistöä. Käsien kirjoituksen tunnistaminen näytöltä. Ääniohjauksella tapahtuvat toiminnot ilman fyysistä kosketusta. Yhtäaikaiset ohjaukset käyttäen erilaisia komennon tulo kanavia. (Adipat & Zhang 2005, 2288).

3 Tutkimusmenetelmät ja toteutus

3.1 Suunnittelutieteellinen tutkimus

Tietojärjestelmien tutkimusta pidetään soveltavana tutkimuksena. Tyypillistä on lainata teorioita muilta tieteenaloilta tutkimuksiin. Tietojärjestelmien tutkimukseen on viimeaika hyväksytty tulokinnallisten tulosten esittäminen, tämä vaatii tulosten olevan selitettäviä. (March & Smith 1995, 253).

March ja Smithin mukaan tietotekniikan tutkiminen jakaantuu kahteen osaan luonnontieteellinen - ja suunnittelutieteelliseksi tutkimukseksi. Luonnontieteellisessä tutkimuksessa tarkoituksena on ymmärtää todellisuutta. Luonnontieteellisessä tutkimuksessa perustana ovat löydöt ja perustelut. (March & Smith 1995, 253).

3.2 Viitekehys

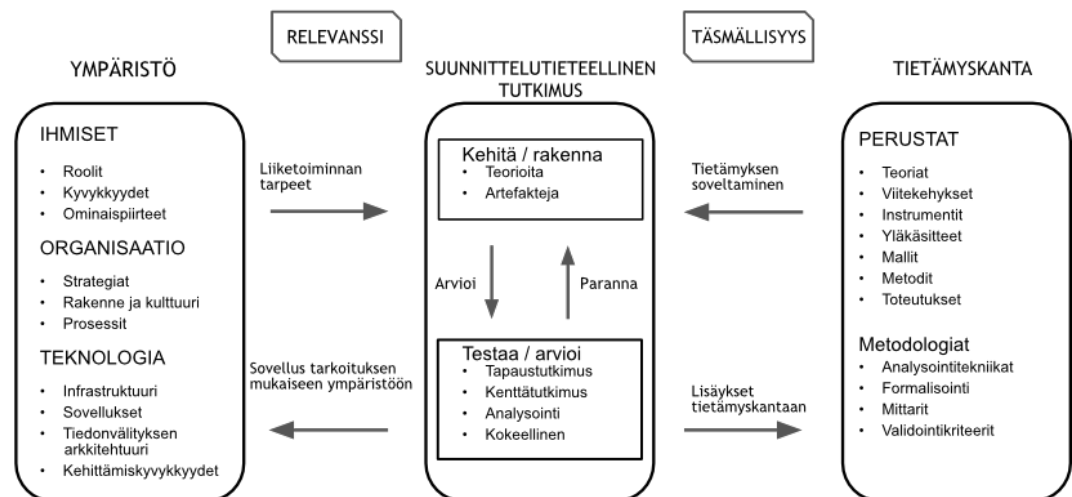
March ja Smith esittelevät tutkimuksessa viitekehysten suunnittelutieteelliselle tutkimukselle. He esittelevät lähestymistavaksi olevan kaksi tapaa rakentaminen ja arvioiminen. He

ovat jakaneet aktiviteetit neljään suoritteeseen, joihin liittyvät rakentaminen ja arvioiminen. (March & Smith 1995, 254 - 255). Taulukko 1 kuvaa March ja Smithin esittämää tutkimuskehystä.

	Suunnittelutiede		Luonnontiede	
	Rakenna	Arvioi	Teorian esittäminen	Perustelu
Käsite				
Malli				
Metodi				
Toteutus				

Taulukko 1: March ja Smithin tutkimuskehys

Suunnittelutieteellinen tutkimus suoritetaan kahdessa toisiaan täydentävässä vaiheessa. Käyttämistiede käsittelee tutkimusta kehittämällä ja perustelemalla teorioita, jotka selittävät tai ennustavat tunnistettuun yritystarpeeseen liittyviä ilmiöitä. Suunnittelutiede kohdistuu tutkimukseen rakentamalla ja arvioimalla esineitä, jotka on suunniteltu vastaamaan yksilöityyn liiketoiminnan tarpeeseen. Käyttämistieteellisen tutkimuksen tavoitteena on totuus.



Kuva 9: Informaatio systeemi tutkimuksen viitekehys

Kuva 9 esittää suunnittelutieteellisen tietojärjestelmän viitekehysten, jonka avulla voidaan selvittää suunnittelutieteellisen tutkimuksen paradigmaa. Ympäristö määrittelee ongelmatilan, se koostuu ihmisistä, organisaatiosta ja rooleista. Se määrittää tavoitteet, tehtävät, ongelmat ja mahdollisuudet, joilla on liiketoiminnan vaatimukset, jotka perustuvat ihmisten

käsitykseen. Näitä arvioidaan suhteessa strategioihin, rakenteeseen, kulttuuriin ja nykyisiin liiketoimintaprosesseihin. Liiketoiminnantarpeet asettuvat tässä teknologiseen infrastruktuuriin, sovelluksiin, kommunikointiarkkitehtuureihin ja kehittämismahdollisuuksiin, jonka perusteella voidaan tutkimuksellista työtä kehittää liiketoiminnan tarpeita vastaaviksi. (Hevner ym. 2004, 273).

Liiketoiminnallisten vaatimusten johdosta suunnittelutieteellinen tutkimus suoritetaan kahdessa vaiheessa, jotka molemmat tukevat toisiaan. Käyttätymistiede lähestyy tutkimusta kehittämällä ja perustelemalla teorioita, niillä on tarkoitus selittää tai ennustaa ilmiöitä, jotka liittyvät yritys tarpeeseen. Suunnittelutieteen lähtökohtana on rakentaa ja arvioida esineitä, jotka on luotu liiketoiminnan tarpeita varten. (Hevner ym. 2004, 273).

Tietopohjalla jakautuu kahteen osaan perusteisiin ja menetelmiin. Perusta sisältää teorian, viitekehyksiä, instrumentteja, rakenteita, malleja, menetelmiä ja toteutuksia. Nämä ovat olennaisia tutkimuksen kehittämis- tai rakennusvaiheessa. Menetelmät toimivat ohjeina tutkimuksen perusteiden ja arvioinnin yhteydessä. Matemaattisia menetelmiä käytetään suunnittelutieteessä laadun ja tehokkuuden arvioinnissa. Käyttätymistieteissä käytettävät tavat ovat tiedon keruuta ja analysointia. Tuloksia voidaan arvioida soveltamalla niitä liiketoiminnan tarpeisiin. (Hevner ym. 2004, 273-274).

3.3 Tutkimusviitekehys opinnäytetyössä

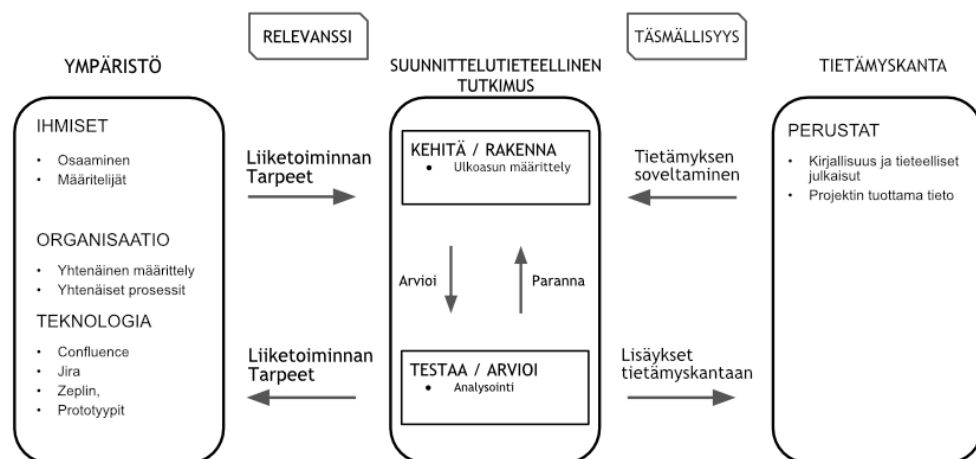
Käyttäen March ja Smith viitekehystä tutkimuksessa,

	Suunnittelutiede		Luonnontiede	
	Rakenna	Arvioi	Teorian esittäminen	Perustelu
Käsite				
Malli	X	X		
Metodi	X	X		
Toteutus				

Taulukko 2: Match & Smith viitekehysten käyttäminen tutkimuksessa.

Opinnäytetyössä on tarkoitus käyttää Hevnerin luomaa suunnittelutieteellisen tutkimuksen mallia. Hevner on luonut teoriansa tueksi seitsemän kohdan ohjeistuksen. Tämän avulla Hevnerin mukaan ensimmäisessä kohdassa tutkimuksen pitää muodostaa artefakti. Tämän määritelmä ei ole tarkka ja artefaktit voivat olla esineitä, malleja tai menetelmiä. (Hevner ym. 2004, 276-277). Toisessa kohdassa mainitaan suunnittelutieteellisen tutkimuksen tarkoituksena olevan hankkia tietoa ja ymmärrystä, joiden avulla teknologia pohjaisten ratkaisujen

kehittämisen ja toteuttamiseen. Tämä paneutuu tähän mennessä ratkaisemattomiin ja tärkeisiin liiketoiminta ongelmiin. Käyttäytymistieteessä tutkitaan tätä kehittämällä ja perustelemalla teorioita, näiden avulla selitetään tai ennustetaan ilmiötä. (Hevner ym. 2004, 278-279). Kolmannessa kohdassa kerrotaan suunnittelu artefaktin arvioinnista. Tässä on kyettävä osoittamaan hyödyllisyys, laatu ja tehokkuus tarkalla tasolla. Tutkimusprosessissa arviointi on tärkeä osa. Liiketoiminnan tarpeista muodostuu arvioinnin kriteerit. Tähän kuuluu myös tekninen infrastruktuuri. (Hevner ym. 2004, 280). Neljännessä kohdassa mainitaan, tutkimuksen tulee osallistua aiheen suunnitteluun, teoria perustaan ja menetelmiin. Tutkimuksessa asetettavan artefaktin tarkoituksena on ratkaista tutkimuksessa asetettu ongelma. Perusteilla on tarkoitus laajentaa nykyistä tietopohjaa hyödyntäen aiempia tutkimuksia ja niiden tuloksia. Arviointimenetelmien kehittäminen ovat osa suunnittelutieteellistä tutkimusta. (Hevner ym. 2004, 281). Viidennessä kohdassa mainitaan suunnittelutieteellisen tutkimuksen ominaisten tutkimusmenetelmien tarkka soveltaminen. Tutkimuksille on tyypillistä tarkasti määritellyt menetelmät, usein nämä ovat matemaattisia menetelmiä. Ominaista määrittää miten hyvin artefakti toimii, eikä teorioida miksi artefakti toimii. (Hevner ym. 2004, 282). Kohdassa kuusi esitetään tehokkaan artefaktin etsiminen edellyttää käytettävissä olevien keinojen käyttämistä haluttujen tarkoitusten saavuttamiseksi ja lakien säilyttämiseksi ongelma ympäristössä. (Hevner ym. 2004, 282-283). Suunnittelutieteelliselle tutkimukselle ominaisena piirteenä ongelma yksinkertaistetaan esittämällä vain osa keinoista. (Hevner ym. 2004, 283-284). Seitsemännessä kohdassa esitetään suunnittelutieteellisen tutkimuksen, on oltava esitettävissä erilaiselle yleisölle. Kohdeyleisön on saatava tarpeeksi informaatio esityksestä kyetäkseen ymmärtämään sen sisällön. (Hevner ym. 2004, 284).



Kuva 10: Tutkimus viitekehys käyttäen Hevnerin mallia

Hevnerin luomasta viitekehuksesta opinnäytetyössä on tarkoitus käyttää ympäristöä ja suunnittelutieteellistä tutkimusta. Kuvassa 10 esitetään tutkimusta varten rakennettu viitekehys,

joka on mukailtu Hevnerin alkuperäisestä mallista. Viitekehyksen avulla on tarkoitus tehdä opinnäytetyössä vaadittava tutkimus.

Tutkimuksen tueksi käytetään Hevnerin luomaa talukukkoa, jonka tarkoitus on tukea suunnittelutielellistä tutkimusta. Taulukosta on jätetty pois viittaukset tietoperustaan. Taulukko 3 esittää tutkimuksessa käytettävän taulukon ja sen sisällön.

Nro	Soveltaminen tutkimuksessa
1	Tutkimuksen artefaktina luodaan ymmärrystä käyttöliittymän määrittämisestä
2	Tutkimuksessa perehdytään vaatimusten määrittelyyn, jonka avulla luodaan artefakti
3	Artefaktia todenpitävyyttä arvioidaan lähdemateriaalia tutkimalla
4	Uutta tietoa koitetaan osoittaa teorian avulla
5	Olemassa olevaa tietoa verrataan materiaalissa oleviin virheisiin
6	Ratkaisu ongelmaa esitetään tutkimuksen tuloksilla
7	Lopputulokset työstä esitellään toimeksiantajalle

Taulukko 3: Hevnerin seitsemän kohdan ohjeistuksen soveltaminen opinnäytetyössä.

3.4 Tutkimustyön kuvaus

Tutkimustyötä aloitettiin tutustumalla suunnittelu tieteelliseen tutkimukseen. Koska sovellusprojekti oli, jatkunut työn aloittamisen aikana yli vuoden aloitettiin työ tarkastelemalla projektin kokonaiskuva. Projektin kehityksessä luotiin uusia määrittämiä, suunnittelua ja kehitystä. Uusia virheitä kirjattiin jatkuvasti. Työssä päätettiin jatkaa virheiden seuraamista aina projektin päättymiseen saakka, jotta kaikki projektissa mahdollisesti ilmenevät uudet virheet saadaan mukaan opinnäytetyöhön.

Nykytilan kuvaamisen jälkeen aloitettiin käymällä läpi sovelluskehityksen aikana kirjattuja virheitä. Kaikista projektin aikaisista virheistä etsittiin niitä, jotka viittasivat tavalla tai

toisella sovelluksen käyttöliittymään. Sovelluksesta jonne virheitä kirjattiin, otettiin ulos excel tiedosto, jota oli nopeampi selata. Tiedosta luettiin rivitasolla virheitä ja niiden kuvauksia. Virheet, joiden teksti sisältö viittasi käyttöliittymään kategorisoitiin erikseen. Näiden virheiden tarkemmat tiedot katsottiin sovelluksesta. Tässä vaiheessa tehtiin vielä karsintaa virheiden suhteen.

Virheiden keräämisen yhteydessä käytettiin kommenttikenttään kyseisen virheen kommentointiin. Näistä kommentteista saatettiin osissa tapauksia nähdä, miten virheen korjaaminen oli edennyt. Näitä havaintoja verrattiin keskenään. Sovellusten virheitä kategorisoidaan syyn mukaan, josta virhe oli aiheutunut. Kategorisoinnissa hyödynnettiin sovelluksesta luotuja vaatimusmäärittelyksiä, joihin kyseinen havaittu virhe kohdistui. Sovellus määrittelyksistä saatiin ymmärrystä virheen alkuperästä. Määrittelysten versionhallinnasta oli hyötyä tutkittaessa virheen alkuperäistä syytä, jossain kohdin määrittelyksiä oli korjattu virheen ilmentymisen jälkeen, tai määrittelyksiä oli kokonaan muutettu. Sovelluksen kaikista näytöistä piirretyt havainne kuvat, oli kirjattujen virheiden vertaileminen suunnitelmiin helppoa.

Tätä jatkettiin tutkimalla tarkemmin sovelluksessa tehtyjä vaatimuksia. Sovelluksen vaatimuksia käytiin läpi tarkastellen, miten siinä on otettu huomioon käyttöliittymä. Käyttöliittymä määrittelyksistä, tarkasteltiin käyttöliittymä kuvauksen kirjoitusta ja tutkittiin näitä kirjoitettuja määrittelyksiä kuvia vasten. Tässä kohden tarkasteltiin määrittelyn paikkaansa pitävyyttä, jotta sitä vasten voitiin toteuttaa sovelluskehitystä. Käyttöliittymän määrittelyksiä vertailtiin keskenään havaitakseen eroavaisuuksia. Pääosin koko projektin aikana pysyttiin samassa määrittelyn mallissa.

Projektissa havaittuja virheitä vertailtiin myös erikseen suoraan piirrettyjen ruutu kuvien kanssa. Tässä tutkittiin tarkemmin yksittäisten elementtien toteutusta suunnittelun perusteella. Projektissa on luotu suunnittelun yhteydessä yhtenäisiä komponentteja, joita on ollut tarkoitus käyttää läpi toteutuksen. Tässä vaiheessa oli kiinnostuttu komponenttien toteutuksesta projektin eri vaiheiden aikana.

Projektin alussa luotiin käyttöliittymäprototyyppejä sovelluksesta. Näillä pääasiallinen tarkoitus oli sovelluksen testaaminen asiakkailla. Käyttöliittymä prototyyppejä verrattiin sovelluksesta luotuihin näyttö kuviin ja toteutukseen. Näistä verrattiin eroavaisuuksia keskinäisestä toteutuksesta.

3.5 Tutkimusaineisto

Tutkimuksellinen aineisto saadaan sovelluskehitys projektissa raportoiduista virheistä. Virheellä tarkoitetaan poikkeamaa sovelluksessa, joka ei ole määrittelysten mukainen. (Haikala & Mikkonen 2011, 206). Virheen syy ei ole aina yksiselitteinen. Virheet voivat johtua huonosta ohjelmoinnista tai suunnittelusta. Vaatimuksien puuttuminen tai puutteet niissä voivat johtaa

virheisiin. (Runeson ym. 2006, 83). Virheitä sovelluksessa raportoitiin koko sovelluskehitys projektin ajan. Syy projektissa ilmeneviin virheisiin johtui puutteista tai epäselvistä vaatimus määrityksistä. Vaatimus määrittelyissä ilmenevät virheet ovat suoraan yhteydessä virheisiin sovelluksen ohjelmointiin. (Runeson ym. 2006, 83). Sovelluksen käyttöliittymän suunnittelussa tapahtui myös virheitä. virheistä sovelluksen käyttöliittymän suunnittelussa sekä sovelluksen ohjelmoinnissa syntyneet virheet. Suunnitteluun liittyvien virheiden korjaaminen on nopeampaa, kun virheet huomataan ennen sovelluskehitystä. (Runeson ym. 2006, 83).

Virheitä kirjattiin koko projektin ajan virheiden hallinnan työkaluun. Pääasiallinen vastuu virheiden raportoinnista oli sovellus testaaajille. Virheitä kirjasivat myös muutkin projekti organisaation jäsenet. Tästä aineistosta opinnäytetyöhön on rajattu virheet, jotka koskevat sovelluksen käyttöliittymää. Sovelluksessa raportointiin myös muita virheitä, kuin käyttöliittymään vaikuttavat.

Opinnäytetyössä tutkitaan määrittelyjen toteutusta havaittuihin virheisiin, jotka liittyvät käyttöliittymään. Opinnäytetyössä katsotaan vaatimusten määrittely ja niihin liittyviä käyttöliittymän virheitä. Projektin tuottamassa aineistossa on tutkittavissa vaatimusmäärittelyt.

3.5.1 Aineiston analysointi

Tutkimuksessa selvitettiin sovelluksesta kirjattujen virheiden tiedosta käyttöliittymään liittyvät virheet. Raportoituja virheitä vertailemalla määrityksessä olevaan kuvaan analysoitiin, johtuiko virhe sovelluksen suunnittelusta, määrittelystä vai toteutuksesta. Virheitä kategorioitiin näihin luokkiin kokonaismäärien ymmärtämiseksi. Taulukko 4 esittää miten projektissa käyttöliittymään liittyvät virheet jakautuivat. Pääosa havaituista virheistä oli peräisin toteutuksesta. Toteutus ei kyennyt toteuttamaan suunnitellun kaltaista ulkoa-asua sovellukselle. Yksittäisiä virheitä kirjattiin koko projektin aikana. Pääasiallinen ajankohta oli uuden kehitys-version julkaisun jälkeen tapahtuva kehitettyjen ominaisuuksien katsastaminen. Suunnittelun osuus johtuvista virheistä oli toiseksi suurin. Tähän voidaan katsoa kuuluvan virheitä suunniteltaessa sovellusta, jonka pitää toimia monen eri valmistajan laitteissa mukautua erikokoisiin näytön resoluutioihin. Testauksen yhteydessä saatettiin havaita suunnitellun asian sopimattomuus sovellukseen suunnitellun laisena. Vaatimusmäärityksiin liittyvät virheet on vaikeampi havaita. Tässä tutkimuksessa havaittiin neljä virhettä, jotka liittyivät vaatimusmäärittelyyn. Yleisin syy näihin oli jonkin asian huono toimivuus, jolloin luotiin uusi tai täydentävä määrittely asiasta.

Syy	Määrittely	Suunnittelu	Toteutus
Kpl	4	11	91

Taulukko 4: Virhe kirjausten jakautuminen

Projektissa virheitä kirjattiin usean henkilön toimesta. Toteuttajan testaajat tekivät sovelluksen ensimmäisen testauksen ja kirjasivat siinä havaitsemansa virheet. Näissä virheissä ei pääasiallisesti kiinnitetty huomiota pientä tarkkuutta vaativiin käyttöliittymällisiin kohtiin. Testaus ei tässä kohtaa kiinnittänyt käyttöliittymään tarkemmin huomiota. Testaus keskittyi pääasiallisesti täyttämään määrittelykselle annetusta hyväksyntä kriteeristön. Hyväksyntä kriteeristöllä kerrotaan, mitkä asiat tulee täytyä kehitystyössä. Näiden kohtien täytyessä voidaan katsoa työn olevan valmis. (Neugent 1982, 443). Asiakkaan tekemässä testauksessa pääsääntöisesti raportoitii käyttöliittymään vaikuttavista virheistä. Tätä testaus tekivät projektissa määrittelijät, suunnittelijat sekä hyväksymistä testaajat. Määrittelijöiden ja suunnittelijoiden kiinnostus kehitetyn osan toiminnasta edesauttoi havaitsemaan pääosan virheistä.

3.5.2 Määrittelysten analysointi

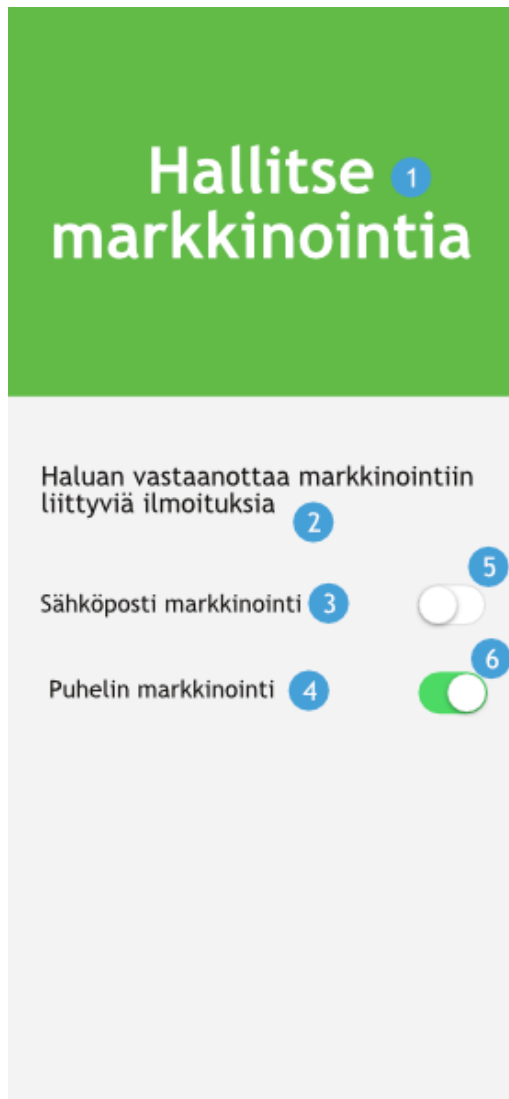
Projektissa käytetty määrittelymalli mukaili SAFe:n mallia. Määrittelyt olivat kolme tasoisia. Ylintä tasoa määrittelyssä edusti SAFe:n mukainen Epic, tämä sisältää jokin merkittävän kokonaisuuden, jota toteutetaan pienemmissä osissa. Epic kuuluu määrittämään MVP (Minimum Viable Product, pienin mahdollinen tuote, joka syntyy epicin valmistuttua. (SAFe, epic). Projektin mallissa toimitettiin yhden epicin alla, joka oli myös projektin MVP. Aina ensimmäiseen julkaisuun ja sen jälkeen työskentely tapahtui pääsääntöisesti yhden epicin alla.

Epicin alla projektissa luotiin featureita, jotka ovat pienempiä kokonaisuuksia. Nämä jakavat epicin tehtävät pienemmiksi toteutettaviksi kokonaisuuksiksi. (SAFe). Näillä täytetään liiketoiminnallisia kokonaisuuksia. Olennaista featurelle on sen hyötyjen maininta ja hyväksymiskriteeri. (Measey 2015, 81). Featureiden malli muuttui projektin aikana. Aluksi projekti käytti featurea epicin tavoin ja myöhemmin siirtyi käyttämään, sitä suoranaisesti epicin alla.

Featureiden alle projektissa luotiin user storyja. User story on osa featurea, jolla on tarkoitus toteuttaa jokin featuressa määritetty toiminto. Näille ominaista on sisältö, joka tulisi olla toteutettavissa määritetyn ajanjakson sisällä. User storyn ollessa osa featurea, voi featurella olla useita user storyjä. User storyssä kuvataan kuka - mitä -miksi mallilla sisältö. (SAFE. story).

Sovelluksen käyttöliittymän määrittelyssä pääasiallinen työ oli merkitä suunnittelu tiimin piirtämään kuvaan numeroilla kentän merkitystä, jos sillä oli osuutta määrittelyyn kanssa. Riippuen käyttötapauksesta kuvia saattoi olla yksi tai useampi. Kuten kuvassa 11 havainto

esitetään millä tavalla projektin aikana käyttöliittymään liittyviä määrittelyjä luotiin projektissa. Kuvaan lisättiin numero kohtia, joiden tarkemmat kuvaukset esitettiin määrittelyissä taulukkonäkymässä. Taulukoissa kerrotaan mitä kuvassa olevalla numerolla tarkoitettiin. Riippuen numeron kohdasta merkitys saattoi olla teksti resurssin avaimen nimi tai rajapinnan sisältämän arvon kenttä. Toiminnollisuuksia kuvaavissa kohdissa saatettiin mainita painikkeen toiminta tai tapahtuva toiminto. Määrittely ei ottanut kantaa kenttien tekstien kokoon tai tyyliin. Sovellus kehittäjillä oli mahdollisuus tutustua, miten elementit oli suunniteltu. Marginaalit oli mahdollista tarkistaa ennen kehitystä ja sen aikana sovelluksesta, jonne tehdyn suunnittelukuvat ladattiin. Samassa määrittelyssä saattoi olla erilliset kuvat kahdelle eri käyttöjärjestelmä versiolle. Näissä kuvissa pyrittiin samaan numerointiin helpottaakseen kehitystä.



Kuva 11: Havainnollistava kuva käyttöliittymän määrittelystä

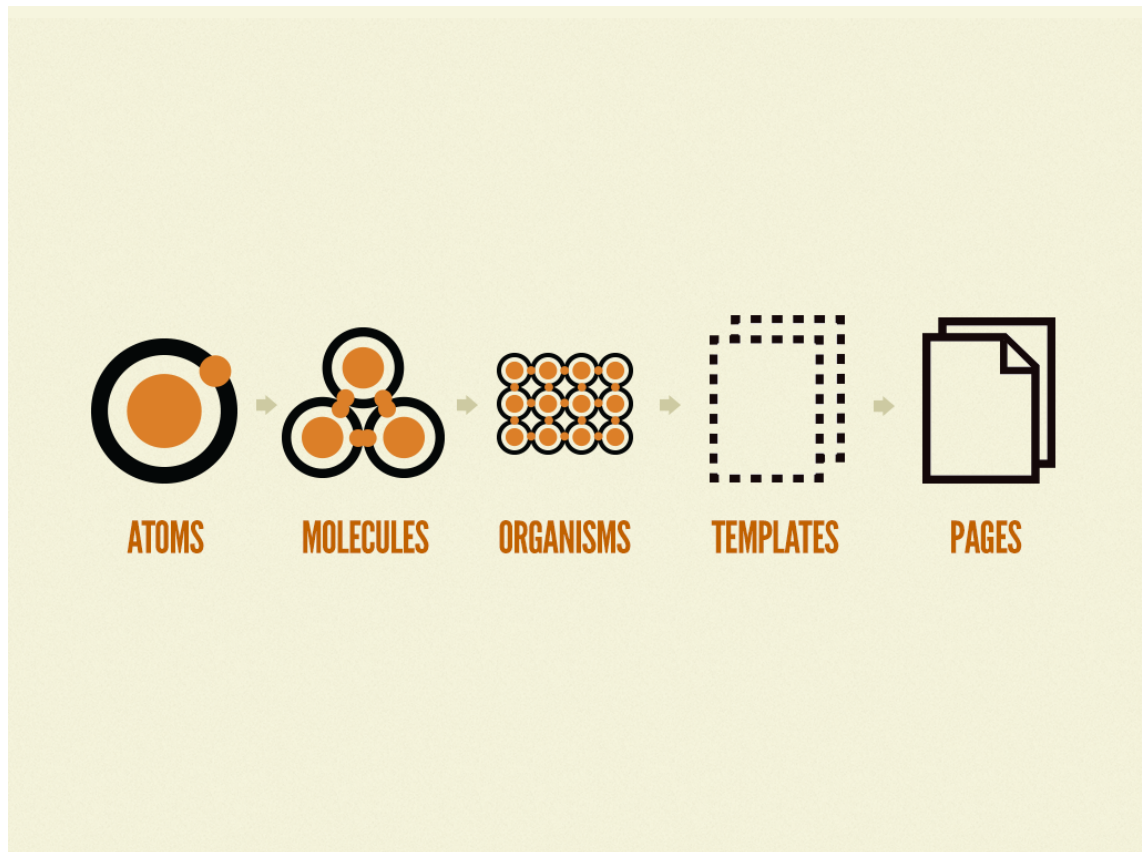
3.5.3 Suunnittelun analysointi

Suunnittelussa sovelluksen kuvat ladattiin palveluun, jossa ne olivat esillä. Sovelluksen kaikista ruuduista tehtiin. Sovelluksen suunnittelua tuki käyttäjä tutkimuksissa saatu palaute sovelluksesta ja hyväksi havaittuja toimintoja käytettiin sovelluksessa. Sovelluksesta tehtiin kahdelle eri käyttöjärjestelmälle. Tämän vuoksi käyttöliittymän suunnittelussa jouduttiin tekemään poikkeuksia käyttöjärjestelmien vuoksi. Alkuperäisen suunnitelma mukaan sovelluksessa olisi ollut samanlainen Käyttöliittymä molemmilla käyttöjärjestelmillä.

Suunnittelussa lähdettiin rakentamaan mallia, jossa samoja elementtejä voidaan uudelleen käyttää. Ideologiana ei haluttu, jokaisen kuvan olevan itsenäinen. Uudelleen käytettävät elementit toistuvat jokaisessa kuvassa. Pääasiassa molemmat käyttöjärjestelmät hyödynsivät samoja elementtejä, mutta muutamissa tapauksissa jouduttiin tekemään poikkeuksia.

Suunnittelua tehtiin koko projektin aikana. Suunnittelussa pystyttiin havaitsemaan toteutus tiimin kyvykkyksiä ja mukauttamaan suunnittelua toteutusta tukevaksi. Suunnittelijat ymmärsivät millaisia toteutuksia ei välttämättä ole mahdollista saada suoritettua.

Sovelluksen käyttöliittymän kehittämisessä lähtökohtana oli Atomic Design metodologia. Tämä on käyttöliittymän suunnittelu menetelmä ja sen tarkoituksena on auttaa käyttöliittymän suunnittelussa. Tämä prosessi koostuu viidestä vaiheesta, jotka jaetaan atomeiksi, molekyyleiksi, organismeiksi, pohjiksi ja sivuiksi. Prosessimainen toteuttaminen ei ole tämän metodologian keskiössä, tämä tarjoaa tukea suunnittelulle. Jokainen sivu koostuu pienemmistä asioista ja atomic designin avulla voidaan ymmärtää mitä nämä pienemmät kokonaisuudet ovat ja mikä niiden vaikutus on. Käyttöliittymän suunnittelussa uudelleen käytettävyys helpottaa työtä. (Frost 2020). Käyttäen atomic design periaatteita suunniteltiin eri tasoisia komponentteja, joita hyödyntäen suunniteltiin sovelluksen käyttöliittymä. Kuvassa 12 esitetään atomic designing periaatetta suunnittelussa.



Kuva 12: Atomic design prosessi (Frost. 2020, <https://atomicdesign.bradfrost.com/images/content/atomic-design-process.png>.)

3.5.4 Toteutuksen analysointi

Sovelluksen kehityksessä toimi kaksi ryhmää, jotka molemmat keskittyivät omaan käyttöliittymän kehitykseen. Osa ryhmän jäsenistä toimi asiakkaan tiloissa ja osa ulkomailla eri aikavyöhykkeellä kuin toimittajan tiloissa olevat. Tiimit kävivät keskenään tiedonvaihtoa käyttäen pikaviestimiä ja pitämällä neuvotteluja. Sovelluskehityksen johtaminen tapahtui molemmissa ryhmissä toimittajan sijainnista. Kehityksellä ei ollut kokonaisuuden johtamista toimittajan puolesta, molempien ryhmien johtavat kehittäjät ohjasivat työtä ja ryhmiään. Ottaen huomioon kahden eri käyttöjärjestelmän eroavaisuudet, joutuivat kehitysryhmät tekemään välillä toisistaan poikkeavia ratkaisuja. Hajautetulla ohjelmistokehityksen projektilla parhaiden tulosten saavuttaminen ei ole yksinkertaista. Vaatimusten esittäminen ymmärrettävästi hankaloituu johtuen eroavaisuuksista ympäristöissä. (Ali & Lai 2017, 208). Hajautetussa ohjelmistokehitys projektissa eri aikavyöhykkeillä jakautuminen tuo projektille lisää kehitys tunteja, mutta samalla kasvokkain tapahtuvat kontaktit vähenevät projektin aikana. (Marcirio & Majdenbaum 2020, 18-19). Projektissa asiakkaan tiloissa olleen kehitysryhmien kanssa päivittäinen kommunikointi oli nopeaa ja tehokasta, ulkomailla toimivien henkilöihin nähden. Ohjelmistoprojekteissa keskinäinen kommunikointi ja tiedon jakaminen ovat keskeisessä osassa.

Ihmisten on tiedettävä mikä tehtävä on kullakin työn alla. Tämä on myös projektin johtamisen kannalta haasteellista. (Marcirio & Majdenbaum 2020, 18-19).

4 Tulokset

4.1 Vaatimusten määrittäminen

Tutkimuksen ensimmäisessä ala kysymyksessä kysyttiin mitä on vaatimusten määrittely. Tähän haettiin vastausta teoriaosuudessa. Toisena kysymyksenä selvitettiin käyttöliittymän määrittäystä mobiilisovellukseen.

Vaatimusmäärittelystä käydään tietoperustassa läpi käsitettä ja sen tarkoitusta. Tutkittaessa vaatimusmäärittelyn prosessia Leffingwell & Widrig pyramidi tuo esille, miten vaatimukset luovat perustan tarvittavilla ominaisuuksille tarpeille. Yksi tarve vaatii toteutuakseen useamman ominaisuuden, jotka tuovat tälle ominaisuudelle lisää vaatimuksia. Vaatimusten määrittelyn on prosessi, jonka aika vaatimuksia tarkennetaan, analysoidaan ja esitetään. Vaatimusmäärittelyn tuloksena on sovellus kehitykselle toteuttavat määreet sovelluksen toiminnasta.

Vaatimusmäärittelystä muodostuu eri teorioiden mukaan malli, jolla on kolme päävaihetta. Vaatimukset analysoidaan, eli selvitetään millaisiin asioihin, on löydettävä ratkaisu. Tässä kohtaa kerätään tietoa sidosryhmiltä. Saatujen vaatimusten perusteella voidaan jaotella vaatimuksia toiminnallisiksi ja ei toiminnallisiksi vaatimuksiksi. Vaatimusten dokumentointi mahdollistaa vaatimusten siirtämisen sovelluskehitykselle. Dokumentointi on myös oleellista vaatimuksen jäljitettävyyden ja sovellus testauksen kannalta. Sovellustestaus on riippuvainen vaatimus dokumentaatiosta. Dokumentaation hyödynnettävyyttä varten onkin syytä kiinnittää huomioita laatuun. Dokumentaation luomiseen voidaan osallistaa useampia tahoja, joka sitouttaa myös tiiviimmin osallistujia. Valmiit määrittelyt esitetään sidosryhmille ja varmistetaan niiden täyttävän sidosryhmien tarpeet. Oleellista on hyväksynnän yhteydessä todeta virheettömyys. Vaatimusten korjaaminen tässä vaiheessa, aiheuttaa pienimmän vaikutuksen kokonaistyömäärään.

Vaatimusten elinkaaren malli Heathin esittämänä kuvaa millaisista asioista vaatimus koostuu ja mitä siinä huomioitavana. Nämä kohdat sijoittuva erivaiheisiin vaatimusmäärittelyksen prosessia. Tämän avulla voidaan selkeämmin työskennellä vaatimusmäärittelyksen prosessin kanssa.

4.2 Käyttöliittymäsuunnittelu

Tutkimuksen toisena ala kysymyksenä selvitettiin miten käyttöliittymän määrittely kannattaa toteuttaa. Tähän esitetään vastausta tietoperustassa. Xiaon prosessi malli esittää käyttöliittymän suunnittelulle mallia, jolla on yhteyttä sovelluksen testaamiseen. Käyttöliittymän

vaatimukset ovat osa toiminnallisia vaatimuksia, joka olisi huomioitava määrittelyn yhteydessä. Tätä ei kannata sekoittaa käytettävyysvaatimusten kanssa. Adipat ja Zhang viitekehyksen avulla huomioida onko käyttöliittymä- vai käytettävyysvaatimus kyseessä.

Tärkeäksi tekijäksi nostetaan käyttäjän huomioiminen suunnittelussa. Käyttäjistä mainitaan edistykselliset ja henkilöt, jotka eivät ole edistyksellisempiä käyttäjiä. Tässä kohtaa on kyettävä huomioimaan pääkäyttäjäryhmät sovellukselle. Suunniteltaessa mobiilisovellusta teoria nostaa suurimmaksi haasteeksi päätelaitteen ruudun koon. Päätelaitteiden kirjo on laaja ja niillä on erilaisia ominaisuuksia. Päätelaitteen käyttötapa suunnittelussa nostetaan esille. Onko tarkoitus toimia vaaka- vai pysty formaatissa.

4.3 Tulosten synteesi

Tuloksina tutkimuksen kysymykseen käyttöliittymään liittyvä vaatimusten määrittely. Kohde projektin käyttämää käyttöliittymän mallia ei teoriaosuudessa suoranaisesti mainita. Projektin käyttämä malli on ollut hyvin selkeä toteuttavalle osapuolelle.

Käyttöliittymän vaatimusten määrittämiseen voidaan soveltaa tavallista määrittelyn prosessia. Kuitenkin Leffingwell & Widrig nostavat esille käyttöliittymän jättämisen pois vaatimusten määrittelystä ja keskittyttävän toiminnollisuuksien määrittämiseen. (Leffingwell & Widrig 1999, 185). Xiaon mukaan osana määrittelyä ovat käyttöliittymä sekä käytettävyys. (Xiao 2020, 5). Leffing ja Widrig mainitseva tarpeiden ja ominaisuuksien muodostavan vaatimukset.

Tärkeämpänä löydöksenä opinnäytetyössä voidaan nostaa käyttöliittymän suunnittelun ja kehityksen yhteyden. Hyväksyttäessä tehtyä kehitystä lisäämällä ulkoasun vastaavuuden yhdeksi kriteereistä voidaan saavuttaa parempia tuloksia. Leffingwell ja Widring nostavat tämän esille mallissaan, joka kuvaa sovelluskehityksen ongelmia. Virheiden korjaaminen hyväksymistauksen yhteydessä on projektille helpompaa, kuin ylläpitovaiheessa. (Leffingwell & Widrig 2020, 20). Xiaon mallissa testaus liitetään osaksi kokonaisuutta. Näin ollen käyttöliittymän määrittelyyn ei synny suoranaista ohjeistusta. Vaatimuksen dokumentaatiossa on kyettävä kuvaamaan riittävällä tasolla käyttöliittymä ja sen tarpeet.

Tuloksia verratessa käytettyyn tutkimusmenetelmään huomataan tietämuskantaan soveltuvan tulosten lisääminen. Opinnäytetyössä tuotetaan uutta tietoa käyttöliittymän vaatimusmäärittelyyn nostamalla määrittelyn dokumentaation suhteen vaatimuksista muodostuvassa sovelluksen tuotantoon hyväksymisessä. Saatuja tuloksia voidaan käyttää Hevnerin mallissa uuden artefaktin analysoinnissa.

4.4 Tarkistuslista

Hevner esittää arvioinnin olevan olennainen osa suunnittelua tieteellistä tutkimusta. Hevner esittää tarkistuslistan käyttämistä osana suunnittelu tieteellistä tutkimusta. Tarkistuslistan

halutaan selvittää täyttääkö projekti tutkimus tärkeimmän näkökohdat. (Hevner & Chatterjee 2010, 19). Taulukossa 5 esitellään Hevnerin tarkastuslista.

Nro	Kysymys	Vastaus
1	Mikä on tutkimuskysymys?	Tutkimuksen kysymyksenä on selvittää käyttöliittymän määrittelyä.
2	Mikä on artefakti? Miten artefakti ilmenee?	Artefaktina toimii mobiilisovelluksen käyttöliittymän määrittely
3	Mitä suunnitteluprosesseja käytetään artefaktin rakentamiseen?	Suunnitelmaa arvioidaan teorian sisältöä peilaten. Lähdekirjallisuuteen ja tutkimuksellisiin artikkeleihin perehtyminen. Projektin aikaisten virhe raporttien läpikäyminen ja niiden analysointi. Tutkimukselle relevanttien virheiden sisällön vertaaminen projektissa luotuihin vaatimusmäärittelyihin. Synteesin muodostaminen virheiden ja määrittelyjen välillä. Tulosten analysointi. Tulosten esittäminen
4	Kuinka artefakti ja suunnittelu-prosessi luovat tietopohjan? Mitkä teoriat tukevat artefaktin suunnittelua ja suunnittelu-prosessia, jos sellaisia on?	Tietopohjalla luodaan ymmärrystä vaatimusmäärittelystä sekä käyttöliittymän suunnittelusta. Eryityisesti vaatimus määrittelyä koskevat artikkelit ja tutkimukset luovat tutkimukselle uskottavuutta. käyttöliittymän suunnitteluun liittyvät artikkelit ja tutkimustiedot tukevat synteisiä.
5	Mitä arviointeja tehdään sisäisten suunnittelu syklien aikana? Mitä suunnittelu parannuksia tunnustetaan jokaisen suunnittelu syklin aikana?	Tutkimuksessa käytettävien virheraporttien läpikäyminen ja niiden soveltuvuuden tarkistaminen tutkimuksen käyttöön. Koska virheitä on tehty useamman henkilön toimesta niiden suodattamisessa pyrittävä käyttämään mahdollisimman vähän suodattimia. Valittujen virheraporttien virheiden vertaaminen virhettä koskevaan vaatimusmäärittelyyn. Koska

		yhdestä määrittelystä voi olla useampi virhe on selvitettävä samankaltaiset virheet ja tunnistettava virheet, jotka viittaavat samaan määrittelyyn.
6	Kuinka artefakti tuodaan sovel- lus ympäristöön ja miten se testataan kentällä? Mitä mittai- reita käytetään osoittamaan artefaktin hyödyllisyys ja pa- rannus aikaisempiin esineisiin verrattuna?	Artefakti on tässä kohtaa ymmärrys käyttöliittymän määrittelystä. Artefaktin pohjalta voidaan selittää vaatimusmäärittelyjä tekevien henkilöiden kanssa, mitkä asiat ovat relevanttia ottaa määrittelyssä huomioon ja mitkä asiat eivät välttämättä ole määrittelijän vastuulla. Artefaktia voidaan arvioida, jos seuraavat projektit hyödyntävät opinnäytetyössä havaittuja asioita. Seuraavien projektien tuloksista voidaan tehdä vastaava tutkimus ja katsoa onko tuloksissa eroa.
7	Mitä uutta tietoa tietämyskan- taan lisätään ja missä muo- dossa?	Käyttöliittymä vaatimuksille kannattaa luoda hyväksyntä kriteeristön, jolla tarkistetaan vastaako toteutettu työ haluttua tahtotilaa.
8	Onko tutkimuskysymystä käsi- tely tyydyttävästi?	Kysymykseen haetaan vastauksia vaatimusmäärittelyn teoriasta sekä sitä täydennetään käyttöliittymän suunnittelun teorialla. Ensimmäinen alakysymys oli mitä on vaatimusmäärittely. Tähän vastataan teoriaosan alussa. Toinen alakysymys oli miten käyttöliittymän määrittelyä kannattaa toteuttaa. Teoriaosan lopussa käsitellään käyttöliittymän suunnittelua ja prosessia. Tässä käsitellään vaatimusmäärittelysten näkökulmasta suunnittelua

Taulukko 5: Suunnittelu tieteellisen tutkimuksen tarkistuslista

5 Johtopäätökset ja pohdinta

Opinnäytetyön tuloksena saatiin luotua ymmärrystä käyttöliittymän vaatimusmäärittämisestä. Käyttöliittymän määrittelyn uutta tietoa voidaan kohden yrityksen toiminnoissa hyödyntää tulevissa projekteissa. Opinnäytetyössä käytettiin kahta alatutkimuskysymystä. Näihin tutkimuksen alakysymyksiin luodaan teoriaosuudessa ymmärrystä, joka yhdistetään tutkimuksellisen työn kanssa tulokset osuudessa. Tutkimuksellisen tehtävän kysymyksenä esitettiin käyttöliittymään liittyvä vaatimusmäärittely. Tähän esitetään vastausta tulokset osion synteessissä.

1. Mitä on vaatimusten määrittely
2. Miten käyttöliittymän määrittelyä kannattaa toteuttaa

Vaatimusmäärittelyä tutkittiin teoriaosuudessa. Tähän esitettiin teoriaa vaatimusmäärittelystä selvittämällä vaatimusmäärittelyn olemusta ja tarvetta. Vaatimusten määrittely selvittää tarpeita ja toiveita. Näiden pohjalta se muodostaa dokumentaation, kehitystyötä varten. Vaatimusmäärittelyssä edetään prosessimaisesti. Vaatimukset eivät ole kaikki samanlaisia ja siksi määrittelyssä on huomioitava mitä asia ollaan käsittelemässä.

Käyttöliittymän määrittelyyn havaitaan vaikuttavan mobiililaitteiden koko ja niiden poikkeamat toisistaan. Käyttöliittymän tarkastaminen osana testausta. Käyttäjälähtöisyyden olevan osa kokonaisuutta. Käyttöliittymän määrittely ei ole aivan yksiselitteinen kysymys. Käyttöliittymä on sovelluksen käyttäjälle näkyvä osa, joten sen merkitys sovelluksen menestymiselle on olennaista.

Opinnäytetyössä tutkitussa projektissa, havaittiin suunnittelijoiden olevan vastuussa käyttöliittymätyön toteutuksen valvomisesta. Johtopäätösten perusteella ei ole väliä, mikä taho valvoo käyttöliittymän toteutusta. Tärkeä asia on linkittää käyttöliittymän hyväksyntä osana kokonaisuhyväksyntää toteutukselle. Tällöin on selvää eri osapuolille, milloin toteutettu työ voidaan hyväksyä ja siirtää tuotantoon. Vaatimusmäärittelyn aikana tärkeää on huomioida käyttöliittymää koskevat määritykset ja kategorisoida ne oikein.

Vaatimusmäärittelyn prosessissa käyttöliittymä vaatimukset kulkevat samalla kaavalla. Tärkeää on ymmärtää syötteitä ja luoda niistä toimivan dokumentaatio. Sidosryhmien osallistaminen määrittelyn eri vaiheisiin tuo yleisesti parempia tuloksia.

Projektissa käyttöliittymän toteutuksen seuraamista hoiti pääosin suunnittelutiimi, joka osasi ohjata toteuttavaa tiimiä parhaansa mukaan. Suunnittelu tiimin vastuulla oli antaa tarkempia ohjeita sovelluksen yksityiskohdista toteutukselle. Suunnittelun perustana ollut atomic design malli ei selkiytynyt kehitys tiimille. Sovelluksessa havaittiin komponentteja, jotka eivät olleet keskenään yhtenäisiä. Kehitys tiimeillä ei maantieteellisen hajautuksen vuoksi ollut mahdollisuutta käydä kasvokkain keskusteluja ja ne hoidettiin pääsääntöisesti puhelimitse. Kehitystiimin tiedonvaihto oli tiimin vetäjien vastuulla.

Teoria osuus osoitti Leffingwell & Widrig toteavat myöhempi virheen havaitseminen projektissa lisäävät sen kustannusta. Projektin aikana havaituista virheistä valtaosa keskittyi toteutuksen jälkeen tehtyihin havaintoihin. Virheitä tässä opinnäytetyössä tarkasteltiin toteutuksen jälkeen. Projektin aikana havaittuja virheitä määrittelyssä tai suunnittelussa ennen toteutukseen menemistä ei ole tarkasteltu. Pienin osa virheistä johtui määrittelystä, jonka kustannukset korjauksessa ovat suurimmat. Kaikkia virheitä ei ole raportoitu projektin aikana. Osa havaituista virheistä annettiin suullisesti ja niiden korjaaminen jäi toteuttavan tiimin vastuulle. Osan virheistä ollessa raportoimattomia ei voida sanoa, mikä olisi ollut todellinen virheiden määrä.

Muutoksia projektin suunnittelussa aiheutti lisää työtä määritykselle. Määritysten muutokset kirjattiin aikaisemmin luotuihin määrityksiin. Määrityksiä paranneltiin läpi projektin. Määritykset kerättiin olemassa olevan tiedon perusteella ja tieto kerättiin lisää määrittelyn aikana. Tiedon keräyksen yhteydessä voidaan validoida määrityksen tyyppiä. Määritysten läpikäyminen ja hyväksyminen ovat viimeinen vaihe ennen kehityksen aloittamista.

Tutkimusmenetelmäksi opinnäytetyössä valikoitui suunnittelutieteellinen tutkimus. Tällä menetelmä vaikutti oleva sopiva lähestymistapa opinnäytetyössä tehtävään tutkimukseen. Suunnittelutieteellisen tutkimuksen metodit tarjosivat tutkimukselle uudenlaisen näkökulman. Tutkimuksellisen työn kulku oli määritelty tämän tutkimusmenetelmän osalta selkeästi ja tutkimusta oli loogista tehdä. Hevnerin malli tarjoaa selkeän ohjeistuksen, jota voi seurata työn edistyessä. Ympäristöstä saadaan tutkimukselle tarpeita, joiden avulla luodaan artefakti. Tietämiskantaan saadaan tutkimuksen pohjalta lisättyä uutta tietoa. Tutkimuksen tulokset voidaan siirtää osaksi tietokantaa ja hyödyntää suunnittelutieteellisessä tutkimuksessa. Tieto koostuu tutkimuksen tuloksista, käyttöliittymää koskevana vaatimusmäärittelyn tietona. Tutkimus on luokassaan ainoa, joka käsittelee käyttöliittymän määrittystä.

Opinnäytetyössä esiteltiin esimerkki, jonka mukaan käyttöliittymää oli määritelty. Teoriaosuudessa ei löydetä suoraan viittauksia tapaan. Teorian mukaan on tärkeämpää dokumentoida vaatimukset ja esitellä ne hyväksytysti. Sidosryhmien ja liiketoiminnan tarpeiden täyttäminen on oleellinen osa. Kuvan perusteella voidaan kuitenkin luoda hyväksyntää varten kriteerejä ja verrata määrittelyn kuvaa lopulliseen tuotokseen.

Lopputuloksena voidaan sanoa, ettei projektissa syntyneitä virheitä käyttöliittymän toteutuksessa voida pitää määrittelyn tuloksena. Määritykset ovat välttäneet ottamasta suoraan kantaa käyttöliittymään. Niissä on tulkittu kohtia ja kuvattu niiden toiminnallisuutta tai merkitystä. Paikoitellen käyttöliittymän suunnittelussa ei ole huomioitu eri laitteiden ruutukokoa, jolla on osassa tapauksia vaikuttavuutta toteutukseen. Opinnäytetyö työn tuloksina voidaan pitää suunnittelun osallistumista vaatimusten määrittelyyn tavalla, jossa vaatimusmäärittelijä kertoo toiminnollisuuksista suunnittelijalle ennen käyttöliittymän suunnittelua. Suunnittelijan

tulisi luoda selkeät ohjeistukset käyttöliittymän toiminnollisuuksista. Käyttöliittymä kuvien dokumentointi vaatimusmäärittelyn vaiheessa ja näiden liittäminen osaksi lopullista sovelluksen hyväksyntä kriteeristöä. Tällöin käyttöliittymän hyväksyntään syntyy selkeät kohdat, jotka sovelluksen on täytettävä ollakseen kelvollinen julkaisua varten. Näihin asioihin kiinnittämällä huomiota voidaan edes auttaa ei suunnitellun kaltaisin käyttöliittymän julkaisua.

Kohde organisaatio tekee omille palveluilleen jatkuvaa kehitystä. Opinnäytetyön pohjalta voidaan esittää testaukseen uutta kohtaa. Tässä kohdassa verrattaisiin käyttöliittymää määrittely dokumentaatioissa olevaan tuotokseen. Tämä vaatii kuitenkin testaajalta visuaalista silmää, jotta voidaan todentaa suunnitelmien vastaavan tuotosta. Aikaisemmin organisaatiossa ei ole kovin tarkasti otettu käyttöliittymää mukaan sovelluksen hyväksyntään.

Jatkotutkimuksena opinnäytetyön pohjalta olisi mahdollista tutkia saavutettavuuden määrittäjiä. Koska uusia EU direktiivi pakottaa osaa yrityksiä sekä julkista hallintoa ottamaan huomioon erilaiset rajoitteiset henkilöt palveluiden käyttäjinä. Koska tämä on pakollinen vaatimus pitäisi yritysten ja julkishallinnon tuottaa saavutettavia mobiilisovelluksia ja palveluita. Saavutettavuudelle on monta hyötyjää. Saavutettavuuden päällimmäisenä tarkoituksena on tarjota palveluita niille, joilla on jokin rajoite. Mobiilisovelluksien käyttöliittymien suunnitteluun ja etenkin testauksessa saavutettavuuden toteaminen tulee olemaan merkittävässä roolissa.

5.1 Lähdekritiikki ja eettisyys

Tutkimuksellisen kirjoittamisen mukaisesti työssä on viitattu käytettyihin lähteisiin viittauksilla. Tekstin sisältöjen muuttaminen tai sen suoraan plagiointi ovat vilpillisiä menetelmiä. Vilpillisten menetelmien käytöstä tutkimustyössä kutsutaan harhautukseksi. (Vilka 2020, 44). Tutkimuksessa on tärkeää noudattaa hyviä eettisiä menetelmiä, joilla luodaan uskottava tutkimus. (TENK 2020, 3).

Lähdekritiikissä tarkastellaan käytettyä lähdetä ja sen relevanssia. Tiedon tulisi olla käytettävää ja tutkimuksella todennettua. Tutkittua tietoa voidaan käyttää, kunnes toinen tutkimus kumoo tiedon. (Vilka 2020, 47). Olennaisinta lähdekritiikissä on tutkia käytettäviä lähteitä ja niiden luotettavuutta. (Hirsjärvi ym. 2020, 113). Lähdekritiikissä lähteen arvioinnin tukena toimivat kirjoittajan tunnettavuus ja arvostettavuus. Käytetyn lähteen ikä on vaikuttava tekijä. Lähteen julkaisutaho ja tämän tunnettavuus. Todellisuus ja puolueettomuus lähteellä. (Hirsjärvi ym. 2020, 113-114). Tässä opinnäytetyössä on pääasiassa käytetty lähteinä tutkimusta julkaisuja sekä painettua kirjallisuutta. Lähteissä on myös viittaus valtion viraston tekemään viralliseen julkaisuun.

Lähteet

Painetut

Adipat, B. Zhang, D. Interface Design for Mobile Applications. AMCIS 2005

Ali, N., Lai, R. A method of software requirements specification and validation for global software development. Requirements Eng 22, 191-214 2017.

Cooper, A. Csizmadi, J. Reimann, R. Cronin, D. Noessel, C. Csizmadi, J. LeMoine, D. 2014. About face: the essentials of interaction design. Neljäs painos. John Wiley & Sons, Inc, Indianapolis.

Dooley, John. 2011. Software Development and Professional Practice

Galitz, W. 2007. The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques. Kolmas painos. Wiley Publishing, Inc., Indianapolis, Indiana.

Haikala, I. Mikkonen, T. 2011. Ohjelmistotuotannon käytännöt., 12 painos, Helsinki. Talentum.

Heath, F. 2020. Managing software requirements the agile way : Bridge the gap between software requirements and executable specifications to deliver successful projects.

Hevner, A., Chatterjee, S. 2010. Design Science In Information Systems. Springer

Hevner, A., March, S., Park, J. Ram, S. 2004. Design Science In Information Systems Research . MIS Quarterly 28(1).

Hirsjärvi, S. Remes, P. Sajavaara, P. 2013. Tutki ja kirjoita. 17 ., uudistettu painos. Porvoo

Leffingwell, D. Widrig, D. 1999. Managing software requirements: a unified approach. Addison-Wesley Longman Publishing Co. USA.

March, S. Smith, G. 1995. Design and natural science research on information technology. Decision Support Systems 15(4), 251 - 266.

Marcirio, S.C., Majdenbaum, A., 2020. Distributed Software Development In Agile Projects: A Model For The Promotion Of Social Interactions. Revista de Gestão e Projetos, 11(1), pp. 17-35.

McWherter, J. Gowell, S. 2012. Professional Mobile Application Development, John Wiley & Sons

Measey, P. 2015. Agile Foundations : Principles, practices and frameworks. BCS Learning & Development Limited, Swindon.

Mendoza, A. 2013. Mobile User Experience : Patterns to Make Sense of It All, Elsevier Science & Technology, San Francisco

Min Tun, Phy. 2014. Choosing a Mobile Application Development Approach.

Neugent, W. 1982. Acceptance criteria for computer security. In Proceedings of the June 7-10, 1982, national computer conference (AFIPS '82). Association for Computing Machinery, New York, NY, USA, 441-448.

Niazi, M. 2002. Improving the requirements engineering process through the application of a key process areas approach. AWRE 20202.

Pandey, D., Suman, U., Ramani, A.K.. (2010). An Effective Requirement Engineering Process Model for Software Development and Requirements Management. 287 - 291.

Peppers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science research methodology for information systems research. Journal of Management Information Systems, 24(3), 45-77.

Runeson, P., Andersson, C., Thelin, T., Andrews, A. & Berling, T. 2006, "What Do We Know about Defect Detection Methods?", IEEE Software, vol. 23, no. 3, pp. 82-90.

Vilkka, H, 2020. Akateemisen lukemisen ja kirjoittamisen opas. PS-Kustannus. Jyväskylä.

Wasserman, Anthony. 2010. Software engineering issues for mobile application development. Proceedings of the FSE/SDP Workshop on the Future of Software Engineering Research

Xiao, Bin. 2020. A Requirement Engineering Approach to User-Centered Design. Journal of Physics: Conference Series

Sähköiset

<https://www.scaledagileframework.com/features-and-capabilities/>, lainattu 5.10

JUHTA, Julkisen hallinnon tietohallinnon neuvottelukunta 2018. ICT-palvelujen kehittäminen: Vaatimusmäärittely, lainattu 8.8

loukkausepäilyjen käsitteleminen Suomessa.

https://www.tenk.fi/sites/tenk.fi/files/htk_ohje_verkko14112012.pdf, lainattu 10.8

<https://www.scaledagileframework.com/epic/>, lainattu 5.10

<https://www.scaledagileframework.com/story/>, lainattu 5.10

<https://atomicdesign.bradfrost.com/chapter-2/>, lainattu 13.10

Kuviot

Kuva 1: Leffingwell & Widrig pyramidi.....	9
Kuva 2: Leffingwell & Widrigin malli (Leffingwell & Widrig, 20)	10
Kuva 3: Juhta vaatimusten määrittelyn vaiheet (JUHTA, 11)	11
Kuva 4: Vaatimusmäärittelyn prosessimalli (Pandey & Suman & Ramani 288)	12
Kuva 5: Vaatimusmäärittelyn elinkaari.....	14
Kuva 6: Vaikutus kartan tarkoitus	15
Kuva 7: Käyttäjälähtöisen suunnittelun prosessi (Xiao, 3).....	18
Kuva 8: Viitekehys mobiili käyttöliittymän suunnitteluun (Adipat & Zhang, 2282)	19
Kuva 9: Informaatio systeemi tutkimuksen viitekehys	21
Kuva 10: Tutkimus viitekehys käyttäen Hevnerin mallia.....	23
Kuva 11: Havainnollistava kuva käyttöliittymän määrittämisestä.....	28
Kuva 12: Atomic design prosessi (https://atomicdesign.bradfrost.com/images/content/atomic-design-process.png).....	30

Taulukot

Taulukko 1: March ja Smithin tutkimuskehys	21
Taulukko 2: Match & Smith viitekehysten käyttäminen tutkimuksessa.....	22
Taulukko 3: Hevnerin seitsemän kohdan ohjeistuksen soveltaminen opinnäytetyössä.	24
Taulukko 4: Virhe kirjausten jakautuminen.....	27
Taulukko 5: Suunnittelu tieteellisen tutkimuksen tarkistuslista.....	34