

---

# Käyttäjäkeskeinen käyttöliittymäsuunnittelu web-sovellukseen: Case ATR Soft

---

Diplomityö  
Turun yliopisto  
Tulevaisuuden teknologioiden laitos  
Ohjelmistotekniikka  
2020  
Markku Tuomola

TURUN YLIOPISTO  
Tulevaisuuden teknologioiden laitos

MARKKU TUOMOLA: Käyttäjakeskeinen käyttöliittymäsuunnittelu web-sovellukseen:  
Case ATR Soft

Diplomityö, 84 s., 4 liites.  
Ohjelmistotekniikka  
Joulukuu 2020

---

Käytettävyys ja käyttöliittymät ovat olennaisia elementtejä suurimmalle osalle ohjelmia. Hyvän käytettävyyden tavoittelu on itsestään selvä asia, mutta miten se saavutetaan? Kuka on ohjelman todellinen käyttäjä, kenelle käytettävyyttä optimoidaan? Mikä on tehokkain tapa lähteä suunnittelemaan ohjelman käytettävyyttä, mitä työkaluja ja metodeja tähän on olemassa? Entä miten mitata käytettävyyttä tehokkaasti, luotettavasti sekä vertailukelpoisesti?

Diplomityön toimeksiantajana toimi ATR Soft Oy ohjelmistotalo, jonka palveluihin kuuluvat muun muassa integraatiot, ohjelmointi sekä ohjelmistoprojektien hallinta. Diplomityön aiheena oli suunnitella uusi ja mahdollisimman käyttäjäystävällinen käyttöliittymä ATR Soft:in CT Publisher -tuotteen web sovellusversioon. Suunnittelutyössä käytettiin hyväksi käyttäjakeskeisen suunnittelun periaatteita ja metodeja.

Työn alussa avataan kirjallisuusanalyysillä sitä, mitä käyttäjakeskeisellä käyttöliittymäsuunnittelulla tarkoitetaan ja miten se saavutetaan. Case-osuudessa kerättyä tietoa sovelletaan yksittäisessä suunnitteluprojektissa.

Lopputulokset vahvistivat teoriaosuudessa esitettyjen metodien tehokkuuden ja monet käyttäjakeskeisen suunnittelun työkaluista voitiin todeta hyödyllisiksi käyttöliittymäsuunnittelussa. Työn yhteenvetona voidaan sanoa varsinkin käyttäjäpersoonien, käyttöliittymän prototyyppien ja käyttäjien kanssa testaamisen olevan erittäin tehokkaita menetelmiä hyvän käytettävyyden takaamisessa.

Asiasanat: käytettävyys, käyttöliittymäsuunnittelu, käyttäjakeskeinen

UNIVERSITY OF TURKU  
Department of Future Technologies

MARKKU TUOMOLA: Käyttäjäkeskeinen käyttöliittymäsuunnittelu web-sovellukseen:  
Case ATR Soft

Master of Science Thesis, 84 p., 4 app. p.  
Software Engineering  
December 2020

---

Usability and user interfaces are essential for most programs. Therefore, good usability is an obvious goal, but how does one achieve good usability? Who is the actual user of the program for whom the usability is optimized for? What is the most efficient way to start designing usability of a program, what kind of tools and methods there are for this purpose? And how does one measure usability effectively, reliably and comparably?

The client for this Master's Thesis was the software company ATR Soft Oy, which provides various services, such as integrations, software development and software project management. The subject of this thesis was to design a new, as user-friendly user interface as possible for the web version of their product, the CT Publisher. The principles and methods of user centered design were used while doing the design work.

In the beginning of this thesis we analyze what user-centered design means and how it is done. In the case section this knowledge is applied in an the chosen design project.

The results confirmed the efficiency of the methods presented in the theoretical part of the thesis and many of the tools used in user-centered design were proven useful. As a conclusion, we note that especially user personas, user interface prototypes and usability testing with users are very efficient methods in guaranteeing good usability.

Keywords: usability, user interface design, user-centered

# Sisällys

<b>1</b>	<b>Johdanto</b>	<b>1</b>
1.1	Työn tausta . . . . .	1
1.2	Tavoitteet ja rajaus . . . . .	1
1.3	Rakenne . . . . .	2
<b>2</b>	<b>Käytettävyys</b>	<b>3</b>
2.1	Miksi hyvä käytettävyys on tärkeää . . . . .	3
2.2	Käytettävyyden määritelmä . . . . .	5
2.2.1	ISO 9241-11 standardi käytettävyydestä . . . . .	6
2.2.2	Nielsenin määritelmä käytettävyydestä . . . . .	6
2.2.3	Opittavuus . . . . .	7
2.2.4	Tehokkuus . . . . .	8
2.2.5	Muistettavuus . . . . .	8
2.2.6	Vähän virheitä . . . . .	8
2.2.7	Tyytyväisyys . . . . .	9
2.3	Käyttäjakeskeinen suunnittelu . . . . .	9
2.4	Käyttäjäpersoonat . . . . .	11
2.5	Empatiakartta . . . . .	14
2.6	Käyttäjätarinat . . . . .	15
2.7	Käyttöliittymän prototyypaus . . . . .	17

2.7.1	Paperiprototyypit . . . . .	17
2.7.2	Tietokoneella tehdyt prototyypit . . . . .	19
2.8	Prosessimallit . . . . .	19
2.8.1	Goal-Directed Design . . . . .	20
2.8.2	GUIDe . . . . .	23
2.8.3	Suunnitteluajattelu . . . . .	24
<b>3</b>	<b>Käytettävyyden arviointi</b>	<b>28</b>
3.1	Arviointimenetelmät . . . . .	29
3.1.1	Heuristinen evaluointi . . . . .	29
3.1.2	Kognitiivinen läpikäynti . . . . .	32
3.1.3	Pluralistinen läpikäynti . . . . .	33
3.1.4	System Usability Scale . . . . .	34
3.1.5	Käytettävyydestaus . . . . .	36
<b>4</b>	<b>Case: CT Publisher</b>	<b>39</b>
4.1	Tuotteen taustaa . . . . .	39
4.2	Käyttöliittymä projektin alussa . . . . .	40
<b>5</b>	<b>Case: Prosessi</b>	<b>42</b>
5.1	Suunnittelutyön aloittaminen . . . . .	42
5.2	Määrittelyvaihe . . . . .	45
5.3	Esitutkintavaihe 1: Tuotteen ymmärtäminen . . . . .	46
5.4	Esitutkintavaihe 2: Käyttäjien ymmärtäminen . . . . .	47
5.4.1	GDPR:n huomiointi . . . . .	49
5.4.2	Kyselyjen testaus . . . . .	50
5.4.3	Kyselyiden lähettäminen . . . . .	50
5.4.4	Kyselyjen tulokset . . . . .	51
5.5	Käyttäjäpersoonien määrittäminen . . . . .	51

5.6	Persoonien tavoitteet ja käyttötapaukset . . . . .	53
5.7	UI Suunnittelu ja prototyypaus . . . . .	54
5.7.1	Käyttöliittymän paperiprototyypit. . . . .	57
5.8	Paperiprototyyppien testaus . . . . .	59
5.8.1	Paperiprototyypeistä saatu palaute . . . . .	59
5.9	Siirtyminen tietokoneella tehtyihin prototyyppeihin . . . . .	60
5.9.1	Testaajat . . . . .	60
5.9.2	Wizardin muokkaaminen . . . . .	61
5.9.3	Ensimmäiset tietokoneprototyypit . . . . .	62
5.9.4	Seuraavat versiot ja niiden testaus . . . . .	64
5.9.5	Käyttöliittymän viides versio . . . . .	67
5.9.6	Käyttöliittymän kuudes versio . . . . .	70
5.9.7	Käyttöliittymän seitsemäs versio . . . . .	71
5.9.8	Käyttöliittymän kahdeksas versio . . . . .	72
5.9.9	Viimeistely käyttöliittymä . . . . .	74
5.9.10	Viimeinen testaus SUS . . . . .	75
5.10	Käyttöliittymän toteutuksen malli . . . . .	76
5.10.1	Dokumentaatio . . . . .	76
5.11	Käyttöliittymän implementointi . . . . .	77
5.12	Pilotointi ja julkaisu . . . . .	78
<b>6</b>	<b>Lopputulos ja johtopäätökset</b>	<b>79</b>
6.1	Projektista . . . . .	79
6.2	Menetelmästä . . . . .	81
6.3	Testeistä . . . . .	82
6.4	Yhteenvedo . . . . .	83
	<b>Lähdeluettelo</b>	<b>85</b>

## **Liitteet**

<b>A Tutkimuskysely</b>	<b>A-1</b>
A.0.1 Yleiskysymyksiä . . . . .	A-1
A.0.2 Kysymyksiä varaosa/huoltopalveluiden prosesseista . . . . .	A-2
<b>B Järjestelmän käytettävyysarvio</b>	<b>B-1</b>

# 1 Johdanto

## 1.1 Työn tausta

Käyttöliittymät ovat olennainen osa useimpia järjestelmiä ja usein ainoa osa, jonka käyttäjät näkevät. Hyvän käyttöliittymän ja käytettävyyden uskoisi näin ollen olevan erittäin tärkeä osa tuotesuunnittelua. Valitettavan usein kuitenkin törmää järjestelmiin, joissa käyttöliittymä on syntynyt ohjelmointityön sivussa ja näin ollen järjestelmän käyttökokemus voi pahimmillaan olla todella surkea. Kuinka hyvin järjestelmä voi palvella käyttäjiä, jos heitä ei ole huomioitu kunnolla järjestelmän suunnittelussa? Ja miten ylipäätään löydetään tuotteen käyttäjät ja saadaan heidät tuotua mukaan osaksi tuotesuunnittelua?

Inspiraatio tähän työhön syntyi halusta hyödyntää aikaisempaa mediatekniikan insinöörin tutkintoani. Toiveenani oli, että diplomityöni sisältäisi käyttöliittymäsuunnittelua tai jotain muuta graafista suunnittelua. Aihe konkretisoitui löydettyäni asiakasyrityksen, jolle sain tehdä diplomityöni juuri käyttöliittymäsuunnitteluun liittyen.

## 1.2 Tavoitteet ja rajaus

Työn tavoitteena oli suunnitella käyttöliittymä olemassa olevan työpöytäsovelluksen verkkoversioon.

Alunperin työn oli tarkoitus sisältää käyttöliittymäsuunnittelun lisäksi myös ly-



hyt osio käyttäliittymän varsinaisesta toteutuksesta ja ohjelmoinnista. Työn aikarajoitteiden vuoksi ohjelmointityön osuus jouduttiin rajaamaan pois työstä. Näin ollen tämä diplomityö keskittyy vain käyttäliittymän suunnitteluun ja siinä käytettyihin työkaluihin ja metodeihin.

Tämän työn tutkimuskysymyksenä oli selvittää, miten käyttäjakeskeistä käyttäliittymäsuunnittelua sovelletaan käytännössä ja mitkä ovat sen tuomat suurimmat hyödyt. Tätä tutkitaan case-tapauksen avulla. Työn tavoitteena on identifoida käyttäjakeskeisen suunnittelun hyödyllisimmät työkalut ja menetit ja tunnistaa mahdolliset ongelmat niiden käyttämisessä.

### 1.3 Rakenne

Työn toisen luvun alussa käydään läpi käytettävyyden tärkeyttä ja sen määritelmää. Sen jälkeen luvussa esitetään myös käyttäjakeskeisen suunnittelun periaatteet, sekä suunnittelussa usein käytettyjä metodeja ja työkaluja. Luvun lopussa käydään läpi kolme tunnettua käyttäjakeskeisen suunnittelun prosessimallia.

Kolmannessa luvussa kerrotaan käytettävyyden arvioinnista ja yleisimmistä arviointiin käytettävistä menetelmistä.

Neljäs luku koostuu case-esittelystä. Luvussa käydään läpi läpi tuotetta, jonka käyttäliittymää tässä työssä suunnitellaan ja kerrotaan lyhyesti tuotteen tarkoituksesta ja ominaisuuksista. Luvun lopussa kuvataan millaisia käyttäliittymiä tuotteella oli tätä työtä aloitettaessa.

Työn viides luku kertoo itse käyttäliittymän suunnittelutyöstä ja sen vaiheista.

Kuudes luku sisältää työstä saadut tulokset, johtopäätökset sekä yhteenvedon työn tuloksista.

## 2 Käytettävyys

Tässä kappaleessa kerrotaan siitä miksi hyvä käytettävyys on tärkeää, mistä käytettävyys koostuu sekä tutkitaan eri prosessimalleja siitä, miten saavutetaan hyvä käytettävyys.

### 2.1 Miksi hyvä käytettävyys on tärkeää

Nykyään käytettävyyttä pidetään tärkeänä osana tuotetta. Tuote, jota on helppoa, mukavaa ja tehokasta käyttää mikä antaa luonnollisesti paremman käyttökokemuksen ja sillä on tyytyväisemmät loppukäyttäjät. Hyvän käyttökokemuksen takaaminen käyttäjille on myös hyvä mahdollisuus erottautua kilpailevista tuotteista ja näin ollen tärkeä myyntivaltti. [1]

Tuotetta, jolla on huono käyttöliittymä on epämiellyttävä käyttää ja tehtävien suorittaminen sillä on aikaa vievää ja työlästä. Joskus huonon käytettävyyden omaavalla järjestelmällä ei ole edes mahdollista suorittaa kaikkia käyttäjän työnkulun vaiheita, vaan käyttäjän pitää tavoitteensa saavuttaakseen suorittaa osa työnkulusta muilla järjestelmillä. Huono käytettävyys johtaa myös usein suurempaan määrän yhteydenottoja tekniseen tukeen, mikä tarkoittaa sitä, että teknisen tuen ylläpitoon pitää käyttää enemmän rahaa.

Mahdollisia syitä huonosti toteutettuun käyttöliittymään on, että sen suunnitteluun ei ole käytetty mitään suunnitteluprosessia tai että suunnitteluprosessia ei oltu implementoitu kunnollisesti. Muita syitä ovat esimerkiksi käyttöliittymän riit-

tämätön testaus tai se, että käyttöliittymäsuunnittelussa ei loppukäyttäjiä ja heidän työnkulkuaan oltu otettu huomioon tarpeeksi. Yleinen ongelma on myös, että tuotteen käytettävyyteen ja sen parantamiseen kiinnitetään huomiota vasta projektin loppuvaiheessa, jolloin muutosten tekeminen on kallista.

Hyvän käytettävyyden takaamiseksi on myös tärkeää minimoida käyttöliittymän vaatimaa työmäärää ja näin optimoida käyttökokemusta. Hyvällä käyttöliittymäsuunnittelulla pyritään minimoimaan muun muassa käyttäjän kognitiivista taakkaa, muistin taakkaa, visuaalista taakkaa sekä fyysistä työtä [2, p. 151].

Kognitiivinen taakka tarkoittaa käyttöliittymän käyttäjälle aiheuttamaa henkistä taakkaa, eli kuinka paljon käyttäjä joutuu ajattelemaan käyttöliittymän käyttöä. Käyttäjän kognitiivista taakkaa voidaan helpottaa välttämällä tekemästä visuaalisesti turhan sotkuista käyttöliittymää. On myös tärkeää käyttää käyttöliittymäsuunnittelussa hyödyksi jo vakiintuneita mentaalisia malleja. Web-suunnittelussa tähän on olemassa niin kutsuttu "*Jakobin laki*", tunnetun käytettävyyden tutkijan Jakob Nielsenin toteamus siitä, että käyttäjät viettävät suurimman osan aikaa verkossa muilla sivuilla [3]. Tämä tarkoittaa, että omaa sivuaan ei kannata suunnitella liikaa muista sivuista poikkeavaksi, koska käyttäjillä on jo muodostettu mentaalinen malli siitä miten he olettavat sivujen toimivan. Jos sivuston käyttöliittymä poikkeaa liikaa käyttäjien mentaalisesta mallista, joutuvat he käyttämään enemmän aikaa sivuston käytön ymmärtämiseen kuin itse sivun sisältöön keskittymiseen.

Muita kognitiivisen taakan vähentämisessä huomioon otettavia asioita ovat esimerkiksi ihmisen työmuistin rajallisuus: käyttäjiltä ei saa vaatia liikaa tarpeettomia vaiheita käyttöliittymällä tehtävien tavoitteiden suorittamiseksi. Liian pitkien vaiheiden suorittaminen kuormittaa käyttäjän työmuistia, mikä saattaa johtaa siihen, että käyttäjä kokee käyttöliittymän käytön henkisesti uuvuttavaksi ja ärsyttäväksi. On siis tärkeää pitää käyttäjältä vaadittavien vaiheiden määrä pienenä. Vaiheiden suorittamisen hankaluudesta ei ole vakiintuneita käytäntöjä. Joissain tapauksissa on

suositeltavaa suunnitella käyttöliittymällä suoritettavan tehtävän vaiheet mieluummin hankaliksi, jos halutaan käyttäjien keskittyvän enemmän vaiheiden suorittamiseen.

Visuaalisen taakan helpottamiseksi on useita hyviä tapoja. Huomioon ottamisen arvoisia ovat muun muassa erilaiset gestalt-hahmolaist siitä, miten ihmiset ryhmittävät ja erottelevat eri tavoilla järjestettyjä visuaalisia elementtejä riippuen esimerkiksi niiden samankaltaisuudesta, läheisyydestä ja jatkuvuudesta. Näihin liittyy myös valkoisen tai negatiivisen tilan hyödyksi käyttäminen käyttöliittymän visuaalisessa suunnittelussa. Valkoisella tilalla tarkoitetaan tyhjää tilaa käyttöliittymän eri elementtien välillä, joka voi nimestään huolimatta olla muunkin kuin valkoisen väristä. Liian vähäinen valkoinen tila käyttöliittymässä vaikeuttaa sen eri elementteihin keskittymistä ja valkoisen tilan tehokas käyttö auttaakin käyttäjää erottamaan nopeasti käyttöliittymän tärkeimmät elementit.

Fyysisellä taakalla tarkoitetaan käyttäjän tekemää fyysistä työtä, yleisemmin tietokoneen hiiren liikuttamista ja näppäinten painalluksia. Fyysisen taakan vähentämisen ohjenuoraksi voidaan käyttää Paul Fittsin esittämää Fittsin lakia. Yksinkertaistettuna Fittsin laki tarkoittaa, että mitä pienempi käyttöliittymän nappulat ovat ja mitä pidempi matka kursorin on niihin liikuttava, sitä enemmän aikaa nappuloiden painamiseen kuluu. Fittsin lain mukaan myös näytön reunoilla ja kulmissa olevat painikkeet ovat helpommin klikattavissa, koska kursori ei voi liikkua reunojen ylitse.

## 2.2 Käytettävyyden määritelmä

Käytettävyydelle ei ole vain yhtä määritelmää. Käytettävyys ja käyttöliittymä eivät ole sama asia. Käyttöliittymällä tarkoitetaan järjestelmän näkyviä osia, kun taas käytettävyys sisältää sen lisäksi muitakin asioita, jotka eivät välttämättä ole loppukäyttäjälle niin näkyviä. Kaksi tunnetuinta määritelmää käytettävyydelle ovat ISO

9241-11 standardi sekä Jakob Nielsenin määritelmä käytettävyydestä.

### 2.2.1 ISO 9241-11 standardi käytettävyydestä

Kansainvälisen standardisoimisjärjestön ISO:n 9241-11 standardi jakaa käytettävyyden kolmeen komponenttiin seuraavasti:

- Tuloksellisuus (*engl. Effectiveness*)
- Tehokkuus (*engl. Efficiency*)
- Tyytyväisyys (*engl. Satisfaction*)

Tämän standardin mukaan käytettävyys on sitä miten tuloksellisesti ja tehokkaasti jotain tuotetta voidaan käyttää tietyssä kontekstissa, ja kuinka tyytyväisiä käyttäjät ovat tuotetta käyttäessä tuolloin.

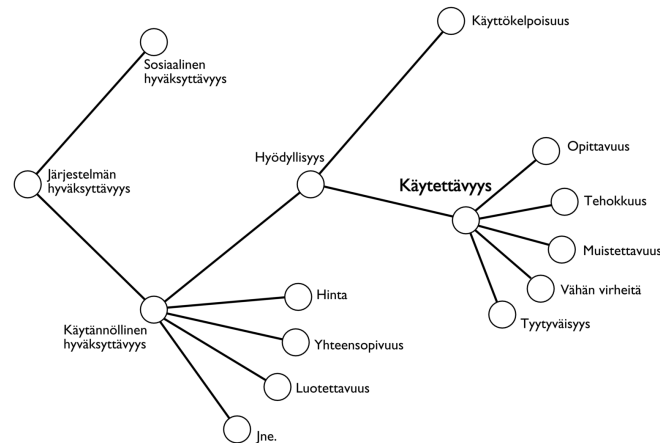
Tuloksellisuudella tarkoitetaan sitä, kuinka täydellisesti ja tarkoin käyttäjä saavuttaa tavoitteensa järjestelmää käyttäessään. Tämä tarkoittaa myös sitä, kuinka virheettömästi käyttäjältä onnistuu järjestelmän käyttö.

Tehokkuudella tarkoitetaan sitä, kuinka paljon resursseja tavoitteiden saavuttaminen kuluttaa. Tehokkuutta mitataan ajallisesti mittaamalla kuinka paljon käyttäjällä kului aikaa tehtävän tai tehtävien suorittamiseen järjestelmällä.

Tyytyväisyys taas on sitä, kuinka miellyttävää ja mukavaa käyttäjälle on käyttää järjestelmää tavoitteidensa saavuttamiseksi. Tyytyväisyys voi olla hyvin subjektiivista ja siksi hankalaa mitata.

### 2.2.2 Nielsenin määritelmä käytettävyydestä

Käytettävyytutkija ja asiantuntija Jakob Nielsen toteaa kirjassaan *Usability Engineering*, että käytettävyys ei ole pelkästään yksiulotteinen käsite käyttöliittymässä vaan käytettävyys on jaettavissa viiteen osa-alueeseen: opittavuus, tehokkuus,



Kuva 2.1: Käytettävyyden elementit (Nielsen, 1993).

muistettavuus, virheet ja tyytyväisyys [1, p.26]. Näiden osa-alueiden tärkeysjärjestys riippuu kuitenkin järjestelmän käyttätarkoituksesta ja kohdekäyttäjistä.

### 2.2.3 Opittavuus

Opittavuudella Nielsen tarkoittaa sitä, kuinka nopeasti käyttäjä oppii käyttämään järjestelmän käyttöliittymää ja näin suorittamaan haluamiaan toimintoja sillä. Opittavuus on järjestelmälle hyvin tärkeä ominaisuus, sillä ihmiset eivät yleensä uuden järjestelmän käytön aloittaessaan ala ensimmäisenä lukemaan ohjeita vaan oppivat mieluummin käyttämällä itse järjestelmää. Tässä kohtaa on hyvä tunnistaa, että erilaiset käyttäjät arvostavat erilaisia oppimiskäyriä järjestelmissä. Jos järjestelmä keskittyy noviisikäyttäjiin, on nopeasti sisällettävissä oleva käyttöjärjestelmä heille mieluisin, kun taas tehokäyttäjät ovat valmiita uhraamaan opittavuutta kunhan järjestelmässä pystytään suorittamaan halutut toiminnot tehokkaasti [1, p.28]. Nämä molemmat käyttäjäryhmät ovat kuitenkin mahdollisesti saavutettavissa yhdellä käyttöliittymällä, esimerkiksi tarjoamalla tehokäyttäjälle erilaisia näppäimien tai näppäinyhdistelmien takaa löytyviä pikatoimintoja [1, p.41]. Näiden pikatoimintojen kanssa on kuitenkin huomioitava järjestelmän virheistä palautuminen: noviisin tai satunnaiskäyttäjän epähuomiossa syöttämästä väärästä pikatoiminnosta on käyttä-

jän pystyttävä palaamaan helposti ja intuitiivisesti takaisin järjestelmän edelliseen tilaan ennen syötettä. On siis tärkeää myös kommunikoida järjestelmän tila käyttäjälle, näin käyttäjä huomaa virheelliseen syötteen ja ymmärtää yrittää palautua siitä.

#### **2.2.4 Tehokkuus**

Tehokkuudella tarkoitetaan sitä, kuinka tehokkaasti käyttäjä pystyy suorittamaan järjestelmällä haluamansa toiminnon kun kyseisen käyttäjän järjestelmän käytön opimisvaihe on ohitse ja käyttäjän voidaan katsoa olevan järjestelmän kokenut käyttäjä. Kokemusta on kuitenkin hankala määrittää formaalisesti, yleisesti käyttäjän voidaan katsoa olevan kokenut kun hän on käyttänyt järjestelmää riittävän pitkän aikaa [1, p.30].

#### **2.2.5 Muistettavuus**

Järjestelmän muistettavuus on tärkeä ominaisuus satunnaiskäyttäjille, eli niille jotka käyttävät järjestelmää vain silloin tällöin. He ovat jo kertaalleen opetelleet käyttämään järjestelmää ja sen sijaan että heidän tarvitsisi aina opetella sen käyttö uudestaan on tärkeää että järjestelmän heille tärkeimmät toiminnot ovat helposti muistettavissa kun he palaavat myöhemmin käyttämään järjestelmää [1, p.31].

#### **2.2.6 Vähän virheitä**

Käyttäjien on pystyttävä käyttämään järjestelmää niin että heille aiheutuu mahdollisimman vähän virheitä ja että niistä palautuminen on mahdollisimman helppoa. On myös tärkeää, että käyttäjille kerrotaan virheestä aina virheen tapahtuessa ja myös sellaisella kielellä mitä käyttäjät itse ymmärtävät. Olisi myös hyvä kertoa käyttäjälle tarkasti mikä virheen aiheutti ja ehdottaa miten virheestä voidaan palautua.

Käyttäjille virheilmoituksia näytettäessä on myös huomioitava kohteliaisuus käyttäjää kohtaan ja se, että käyttäjää ei saa syyllistää tai saada tuntemaan itseään typeräksi. [4]

### 2.2.7 Tyytyväisyys

Tyytyväisyydellä kuvataan sitä, kuinka miellyttävää järjestelmää on käyttää. Tyytyväisyys on subjektiivista ja kenties vaikeimmiten mitattavissa. Suurimmassa osaa käytettävyytutkimuksissa käyttäjien tyytyväisyyden mittaamiseksi käytetään erilaisia kyselyjä käyttäjille. Tyytyväisyyden voidaan ajatella olevan tärkeää esimerkiksi kodin viihdejärjestelmissä, koska viihdekäytössä mukavuutta arvostetaan tehokkuutta enemmän. [1, p. 34]

## 2.3 Käyttäjakeskeinen suunnittelu

Käyttäjakeskeisen suunnittelun keskeinen ajatus on loppukäyttäjän mukaan ottaminen ja huomioiminen tuotteen käyttöliittymäsuunnittelussa. Siinä on tavoitteena tuotteen helppokäyttöisyys sekä käyttäjän tarpeiden tyydyttäminen. Käytettävyyttä suunniteltaessa käyttäjakeskeisesti on huomioitava, mihin tuotetta käytetään, millainen henkilö on tyypillinen tuotteen loppukäyttäjä ja millaisessa ympäristössä hän tuotetta käyttää.

Kuten Alan Cooper toteaa [5, p. 22], syntyy käyttöliittymä usein niin, että kehittäjät tekevät itselleen mieluisia ratkaisuja suunnitellessaan ja toteuttaessaan käyttöliittymiä. Sen sijaan että keskitytään käytettävyyteen ja loppukäyttäjään, valittavan usein ratkaisut valitaan sen mukaan kuinka helppo ne on kehittäjän itse toteuttaa. Näin syntyvä käyttöliittymä on usein mieluisa käyttää vain sen kehittäjälle itselleen. Tästä syystä on tärkeää jo suunnitteluprojektin alussa muodostaa kuva tuotteen käyttäjistä ja suunniteltava tuotteen käyttöliittymä heitä varten.



Nämä loppukäyttäjät löydetään määrittelemällä tuotteelle kohderyhmä ja rajaamalla sieltä tuotteen tärkeimmät käyttäjät, he joille tuote pääasiallisesti kohdennetaan. Loppukäyttäjistä tietojen keräämiseen on tarjolla monia eri tapoja. Uutta tuotetta suunniteltaessa voidaan tuotteen tulevia loppukäyttäjiä haastatella ja tarkkailla. Tämä auttaa hahmottamaan, miten tuote ratkaisee käyttäjien jonkin ongelman ja miksi he ylipäättään käyttäisivät tuotetta. Jos tuote on jo olemassa ja käytössä, voi sen nykyisiltä käyttäjiltä kerätä tietoa ja palautetta sähköpostikyselyillä tai mahdollisesti tuotteen verkkosivuilla olevalla kyselyllä. Näiden käytössä on kuitenkin tärkeää ottaa huomioon yleinen tietosuoja-asetus GDPR.

Nielsen [1] esittää käyttäjakeskeisen suunnittelun koostuvan seuraavista vaiheista:

- Tunnista käyttäjät ja hanki tietoa heistä.
- Tunnista ja analysoi olemassa olevia sekä kilpailijoiden tuotteita.
- Aseta käytettävyydelle tavoitteet: opittavuus, helppokäyttöisyys, tehokkuus.
- Rinnakkainen suunnittelu, useampia suunnitelmia vaihtoehtoisista ratkaisuista.
- Käyttäjien ottaminen mukaan suunnitteluun.
- Käyttöliittymän yhtenevyyden koordinointi, jos sen osia kehitetään erillisesti.
- Käyttöliittymäsuunnittelun suuntaviivojen ja heuristiikkojen käyttö.
- Prototyyppien käyttö.
- Käyttöliittymän arviointi käytettävyydestä avulla.
- Iteratiivinen suunnittelu.

## 2.4 Käyttäjäpersoonat

Alan Cooperin yleistämä konsepti käyttöliittymäsuunnittelussa, jonka hän toi esille kirjassaan *Inmates Are Running the Asylum* [5, p. 123]. Persoonat ovat fiktionaalisia representatiivisia hahmoja käyttäjistä, jotka kuitenkin pohjautuvat oikeaan dataan. Persoonien avulla voidaan kiteyttää käyttäjistä kerättyä dataa. Persoonien avulla suunnittelijoiden ja kehittäjien on helpompi ymmärtää, minkälaisia tuotteen käyttäjät ovat ja ottaa heidät paremmin huomioon.

Dataa persoonia varten kerätään tutkimalla kehitettävän tuotteen kohdekäyttäjiä ja potentiaalisia käyttäjiä. Tähän löytyy useita erilaisia käyttäjätutkimuksen menetelmiä, kuten haastattelu, käyttäjien tarkkailu ja kyselyt. Puuttuvia tietoja voidaan täydentää sekundaarisilla tutkimuksilla, esimerkiksi haastatteleamalla sidosryhmiä tai keräämällä dataa sosiaalisesta mediasta. [2, p. 97]

Tuotteelle löytyy yleensä useita erilaisia käyttäjiä, joista jokaisesta olisi syytä luoda oma käyttäjäpersoonansa. Persoonat priorisoidaan niiden merkittävyyden mukaan: [2, p. 104]

- Primaarinen persoona
- Sekundaariset persoonat
- Täydentävät persoonat
- Asiakaspersoonat
- Palveltavat persoonat
- Negatiiviset persoonat

**Primaarinen persoona** on se, ketä varten käyttöliittymää pääasiassa suunnitellaan eli se muodostetaan tuotteen ensisijaista käyttäjistä. Tuotteen käyttöliittymällä on vain yksi primaarinen persoona. Jos primaarisia persoonia tunnustetaan

useampia, on heille jokaiselle oltava oma käyttöliittymänsä, sillä heidän tarpeensa ovat tuolloin hyvin erilaisia.[2, p. 104].

**Sekundaaristen persoonien** tavoitteiden ja tarpeiden täyttämiseen riittää yleensä primaarisen persoonan käyttöliittymä pienillä muutoksilla. Näin ollen heidän tarpeensa tulee suurimmaksi osaksi täytettyä samalla kun keskitytään suunnittelemaan primaariselle persoonalle. On tärkeää, että sekundaariselle persoonalle suunniteltaessa ei rikota käytettävyyttä primaariselle persoonalle. Kaikissa tapauksissa ei välttämättä ole sekundaarisia persoonia ja primaarinen persoona riittää kaikkiin käyttötapauksiin. Jos sekundaarisia persoonia muodostuu liikaa, voi se olla merkki siitä, että tuote yrittää toteuttaa liian monen asiakassegmentin tarpeita kerralla ja tuotteen kohdemarkkinoita on tarkennettava.

**Täydentävät persoonat** ovat tuotteen käyttäjiä, jotka eivät ole primaarisia eivätkä sekundaarisia persoonia, mutta joiden tarpeet nämä persoonat jo täyttävät. Näin ollen heitä ei tarvitse erikseen huomioida suunnittelussa. Täydentäviä persoonia käytetäänkin yleensä pääasiassa demonstroimaan sidosryhmille, että näitä käyttäjiä ei ole jätetty huomioimatta kehityksessä. [2, p. 105]

Persoonia voidaan myös luoda muistakin kuin tuotteen loppukäyttäjistä. Joissain tapauksissa, esimerkiksi B2B-ympäristöissä tuotteen ostaja on eri henkilö kuin sen käyttäjä. Näistä ostajista muodostettuja **asiakaspersoonia** käytetään pääasiassa apuna tuotteen markkinoinnissa, eikä lainkaan käyttöliittymäsuunnittelussa. [2, p. 84]

**Palveltavat persoonat** eivät ole tuotteen käyttäjiä, vaan asiakkaita joihin tuotteen käyttö vaikuttaa suoraan. Näitä voi kohdella sekundaarisina persoonina. Esimerkkinä palveltavasta persoonasta on kaupan asiakas, joka ei käytä kassakonetta, mutta jonka ostokokemukseen kassakoneen käyttöliittymän potentiaaliset käytettävyysoingelmat vaikuttavat.

**Negatiiviset persoonat** eivät myöskään ole tuotteen käyttäjiä, vaan täyden-

tävien persoonien tapaan niitä voidaan käyttää kommunikoimaan muille kehittäjille ja sidosryhmille minkälaisille persoonille tuotetta ei kehitetä.

Mitä tarkempia käyttäjäpersoonista tehdään, sitä tehokkaammin niitä voidaan käyttää käyttöliittymäsuunnittelussa [5, p. 128]. Tämä johtuu siitä, että tarkkaa persoonaa varten on helpompaa suunnitella käyttöliittymää. Varsin usein persoonan määritelmän ollessa puutteellinen, suunnitellaan käyttöliittymää Cooperin mukaan liian helposti niin sanotulle elastiselle käyttäjälle, joka toimii järjestelmän suunnittelijan olettamalla tavalla. Tämä sotii Cooperin ajatusta vastaan siitä, että käyttöliittymän pitäisi toimia käyttäjien olettamalla tavalla. Persoonan elastisuuden vähentämiseksi suositellaan tälle määritettävä hänen taitonsa, motivaationsa sekä tavoitteensa [5, p. 128]. Cooperin mukaan on myös lisäksi suositeltavaa nimetä käyttäjäpersoonat, sillä tämä auttaa kehittäjätiimiä paremmin visualisoimaan persoonaa ja helpottaa käyttöliittymän suunnittelussa tälle persoonalle. Valokuvan liittäminen persoonaan on toinen suositeltu tapa helpottamaan visualisointia ja saamaan persoonan tuntuvan enemmän oikealta ihmiseltä. Persoonien tukena voi käyttää useampia valokuvia, esimerkiksi kuvia tyypillisestä ympäristöstä missä persoona käyttäisi tuotetta. Riskinä on kuitenkin käyttää liiallinen määrä fiktiivisiä elementtejä, mikä vähentää persoonan uskottavuutta [2, p. 103]

Jos resurssit eivät riitä persoonien luomiseen tarkkailemalla tai haastatteleamalla käyttäjiä, on myös mahdollisuus turvautua niin sanottuihin väliaikasiin persooniin (*provisional personas*) [2, p. 86]. Toisin kuin oikeat käyttäjäpersoonat, väliaikaiset persoonat eivät pohjautu oikeaan käyttäjistä kerättyyn dataa, vaan suurimmaksi osaksi suunnittelijoiden oletuksiin käyttäjistä ja heidän tavoitteistaan. Nimensä mukaisesti väliaikaiset persoonat on syytä korvata mahdollisimman pian oikeaan dataan perustuvilla persoonilla.

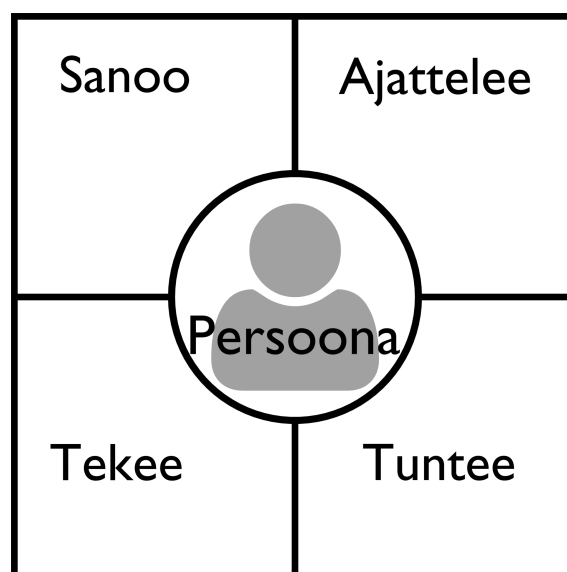
Persoonia voidaan käyttää hyödyksi myös käyttöliittymän testauksessa, sillä vaikka käyttäjäpersoonien käyttö testauksessa ei korvaa testausta oikeilla käyttäjillä, on

persoonista kuitenkin apua kun halutaan testata ja saada nopeasti palautetta käyttäjäliittymästä. Persoonien avulla myös testajat pystyvät paremmin samaistumaan käyttäjiin. [2, p. 79]

## 2.5 Empatiakartta

Empatiakartat toimivat työkaluna loppukäyttäjien ymmärtämisessä ja auttavat rakentamaan empatiaa käyttäjiä kohtaan kehitysprosessin alkuvaiheessa. Empatiakartat jaetaan yleisesti neljään lohkokon kuvaamaan sitä mitä käyttäjä sanoo, ajattelee, tuntee ja tekee. Näiden lohkojen keskellä on itse käyttäjäpersoonana. Dataa empatiakarttoja varten voidaan kerätä esimerkiksi käyttämällä haastatteluja, kyselyjä ja tarkkailemalla käyttäjiä.

Empatiakartat ovat näin ollen myös tehokas työväline datan visualisoinnissa ja sen keskittämässä yhteen helposti ymmärrettävään kuvaajaan. Ne ovat myös nopeasti tehtäviä, olettaen että käyttäjistä on tarpeeksi dataa saatavilla. Empatiakartan pohjana voidaan käyttää yhtä tiettyä henkilöä, tai ne voivat pohjautua useasta eri henkilöstä kerättyyn ja yhdistettyyn dataan. [6]



Kuva 2.2: Empatiakartta.

*Sanoo*-lohkoon kerätään lainauksia siitä, mitä käyttäjät ovat sanoneet haastatteluissa tai esimerkiksi jossain testaustilanteessa.

*Ajattelee*-lohkossa kuvataan käyttäjän ajatuksia silloin kun tämä käyttää tuotetta.

*Tuntee*-lohkoon taas sijoitetaan käyttäjän tuntemukset, kuten käyttäjien toiveet ja huolet tämän käyttäessä tuotetta. *Tuntee*- ja *ajattelee*-lohkojen kanssa voi joskus olla hankala päättää, kumpaan lohkoon jokin asia kuuluu. Tällöin on tärkeää muistaa, että empatiakarttojen päätarkoitus on datan esiin tuonti, eikä kannata takertua liikaa siihen onko jokin informaatio oikeassa lohkossa.

*Tekee*-lohkoon kuvataan mitä käyttäjä fyysisesti tekee ja miten hän käyttäytyy.

Joissain empatiakarttojen versiossa käytetään näiden neljän lohkon lisäksi *näkee*, *kuulee* ja muita lohkoja, sekä näiden yhdistelmiä. On suunnittelijan oma tapauskohtainen valinta, mitkä lohkot hän näkee parhaaksi sisällyttää empatiakarttaan. On tärkeää ymmärtää, mitkä lohkot auttaisivat parhaiten empatian rakentamisessa käyttäjiä kohtaan missäkin käyttötapauksessa.

## 2.6 Käyttäjätarinat

Käyttäjätarinat ovat kuvaus siitä, miten joku käyttäjä käyttäisi järjestelmää tyypillisessä käyttötapauksessa. Käyttäjätarinat poikkeavat käyttötapauksista, skenaarioista ja kuvakäsikirjoituksista keskittymällä enemmän käyttäjään ja tämän tavoitteisiin. Käyttäjätarinoiden avulla käytettävyyden suunnittelu pysyy käyttäjäkeskeisempänä ja näin tuotteeseen lisätään vain niitä ominaisuuksia joita käyttäjät tarvitsevat. Näin vältetään tarpeettomien ominaisuuksien lisäämiseltä. Käyttäjätarinat auttavat myös suunnittelijoita kommunikoimaan paremmin keskenään ja eri sidosryhmien kanssa. Käyttäjätarinat kirjoitetaan käyttäjän perspektiivistä. [7]

Hyvä käyttäjätarina koostuu seuraavista komponenteista [8]:

- Käyttäjä
- Käyttäjän tavoite ja motivaatio
- Tausta
- Juoni
- Oivalentaminen
- Speaktaakkeli

Käyttäjä on luonnollisesti käyttäjätarinan keskiössä. Mitä helpompi käyttäjään on samaistua, sitä uskottavampi käyttäjätarina on. Käyttäjään samaistumiseen voi käyttää apuna esimerkiksi persoonia ja empatiakarttoja.

Käyttäjän tavoite ja motivaatio on myös hyvä tuoda esille käyttäjätarinassa. Tämä auttaa käyttäjän päätöksenteon ymmärtämisessä.

Taustalla tarkoitetaan tarinan kontekstia eli missä ja milloin käyttötilanne tapahtuu. Minkälainen ympäristö käyttäjällä todennäköisesti on tuotetta käyttäessä?

Juonen lisääminen auttaa tekemään tarinasta muistettavamman, mikä helpottaa suunnittelijoita samaistumaan käyttäjiin ja käyttötilanteeseen.

Oivalentaminen on tarinan ahaa-hetki, sen avulla kommunikoidaan aikaisemmin tuntematon käyttäjän kipupiste, joka löydettiin tutkimalla käyttäjiä.

Speaktaakelilla tarkoitetaan käyttäjätarinan eloon tuomista. Tähän voi käyttää apuna esimerkiksi kuvia, videoita tai musiikkia. Näiden avulla on tarkoitus tehdä käyttäjätarinasta helpommin ymmärrettävä ja muistettava.

Yksittäisistä käyttäjätarinoista ei kannata tehdä liian laajoja, vaan ne on syytä pilkkoa pienemmiksi tarinoiksi. Pienet tarinat ovat myös helpommin priorisoitavissa ja suunnittelu voidaan näin aloittaa tärkeimmistä tarinoista. Lyhyissä käyttäjätarinoissa keskitytään vastaamaan kysymyksiin kuka, mitä ja miksi yhdellä lauseella

[7]. Nämä tarinat ovat useimmiten muotoa "X-käyttäjänä, haluan Y jotta Z". X kuvaa käyttäjän roolia käyttötilanteessa, Y kertoo minkä toiminnon käyttäjä haluaa suorittaa saavuttaakseen tavoitteensa Z. Tässä muodossa lyhyetkin käyttäjätarinat auttavat pitämään suunnittelijoiden mielessä käyttäjän ja hänen tavoitteensa.

## 2.7 Käyttöliittymän prototyypaus

Ennen kuin käyttöliittymää lähdetään toteuttamaan ohjelmoimalla, voidaan sen ongelmakohtia paljastaa käyttämällä eritasoisia prototyyppejä. Kuten ohjelmoinnissa yleensäkin, käytettävyyden ongelmat on helpompia ja halvempia korjata kehityksen alkuvaiheessa. Carolyn Snyder ehdottaa kirjassaan *Paper Prototyping* [9, p. 260] neljää ulottuvuutta prototyypeille: leveys (horisontaaliset prototyypit), syvyys (vertikaaliset prototyypit), ulkonäkö sekä interaktiivisuus. Horisontaalit prototyypit ovat yleisiä käyttöliittymäsuunnittelussa, näille on ominaista se että ne demonstroivat hyvin käyttöliittymää ja sovelluksen ulkonäköä, mutta jättävät back-endin toteuttamatta. Jos prototyyppi demonstroi back-endiä, sen sanotaan olevan vertikaali prototyyppi.

Prototyyppien käytön etuna on niiden helppous ja halpuus verrattuna käyttöliittymän ohjelmointiin. Niiden avulla on helpompi saada aikaan keskustelua suunnittelijoiden, kehittäjien ja sidosryhmien välillä kuin esimerkiksi kirjallisilla dokumenteilla. Käyttämällä prototyyppejä ohjelmoinnin sijaan varmistetaan myös se, ettei käyttöliittymän ohjelmointikieli tai kehittäjän ohjelmointitaidot sanele käyttöliittymän ulkonäköä.

### 2.7.1 Paperiprototyypit

Prototyypit voivat yksinkertaisimmillaan olla paperille tehtyjä piirroksia. Paperiprototyyppien etuna on niiden nopeus: ne ovat nopeita hahmotella ja rakentaa, ja



näin ollen mahdollistavat nopeita iteraatioita. Niiden yksinkertaisuus ja tuottamisen nopeus auttaa myös siinä, että suunnittelijat eivät kiinny niihin niin helposti ja näin ollen ovat valmiimpia hylkäämään ne, sekä kokeilemaan uusia ratkaisuja käytettävyyden ongelmiin. Useiden paperiprototyypin käytöstä on myös etuna se, ne auttavat suunnittelijaa reagoimaan palautteeseen vähemmän negatiivisesti. Testihenkilöiden ja muiden on myös helpompi antaa palautetta käyttöliittymistä, kun heillä on jotain mihin verrata niitä. Usein myös käy niin että lopulta paras käyttöliittymä muodostuu yhdistelemällä parhaita piirteitä useista prototyypeistä. [10]

Optimaalista määrää käyttöliittymien prototyypeille ei ole, tärkeintä on saada prototyypit näytettäväksi muille testattavaksi ja palautteen saamiseksi. Tähän riittää jo kaksikin erilaista prototyyppiä. Liikaa aikaa paperiprototyypin luomiseen ei kannata käyttää, sillä on tärkeää muistaa niiden päätarkoitus nopeana prototyypaustyökaluna ja välttää kiintymistä liikaa tiettyyn malliin.

Paperiprototyyppejä ei yleensä tehdä 1:1 skaalassa lopulliseen toteutukseen verrattaessa, vaan käytettävyyden elementit saattavat olla ylisuuria, jotta haluttu toiminnallisuus voidaan esitellä selkeämmin. Värejä paperiprototyypin kanssa kannattaa käyttää hillitysti vain joidenkin elementtien korostamisessa; on muistettava että nopeilla prototyypeillä halutaan hiota toiminnallisuutta, eikä keskittyä järjestelmän ulkonäköön.

Paperiprototyypin ongelmana on juuri niiden yksinkertaisuus ja tarkkuuden puute. Käyttöliittymissä tarkkuutta on kahdenlaista: horisontaalista ja vertikaalista. Horisontaalisesti tarkka käyttöliittymän prototyyppi visualisoi ja emuloi mahdollisimman tarkasti käyttöliittymän front-endiä. Vertikaalisesti tarkalla prototyypillä pyritään emuloimaan back-endiä [1, p .93]. Paperiprototyypit ovat tarkkoja ainoastaan horisontaalisesti. Niiden avulla on myös hankala demonstroida joitain käyttöliittymän yksityiskohtia, toimintoja ja käyttäjälle annettavaa palautetta, kuten esimerkiksi raahaa ja pudota toimintoja tai vaikka kursorin hover-efektejä. Paperi-

prototyyppien testaaminen ei myöskään ole niin helppoa. Siitä kerrotaan luvussa 3 enemmän. [11]

### 2.7.2 Tietokoneella tehdyt prototyypit

Tietokoneella voidaan myös luoda prototyypejä ilman, että niiden luomiseen tarvitsee ohjelmointia. Näillä tietokoneprototyypeillä on mahdollista toteuttaa interaktiivisuutta paremmin verrattuna edellä mainittuihin paperiprototyyppeihin. Tietokoneprototyyppien avulla voidaan myös esittää ja testata paremmin käyttöliittymän elementtien sijoittelua ruudulla, ja käyttöliittymän antamaa palautetta käyttäjälle.

Tietokoneprototyyppien taso riippuu täysin siitä, mitä ohjelmaa niiden luomisessa käytetään. Yksinkertaisimmillaan prototyypejä voidaan luoda esitysgraafikkaohjelmistoilla, kuten Microsoft PowerPoint. Tietokoneprototyyppien luomiseen on kuitenkin nykyään saatavilla useita juuri siihen tarkoitukseen tarkoitettuja ohjelmia, kuten esimerkiksi Adobe XD, InVision, Figma sekä Sketch. Useimmista tietokoneprototyyppien luomiseen tarkoitetuista ohjelmista löytyy myös työkaluja niiden jakamisen ja testauksen helpottamiseksi.

Tietokoneprototyyppien etu paperisiin verrattuna on, että ne vastaavat enemmän lopullista järjestelmää. Toisin kuin paperiprototyyppien kanssa, niillä voi testata miten elementit kannattaa sijoittaa ruudulla, elementtien kokoa, fontteja sekä värejä. Mutta kuten paperiprototyyppien kanssa, ei tietokoneprototyypeillä yleensä ole juurikaan vertikaalista syvyyttä.

## 2.8 Prosessimallit

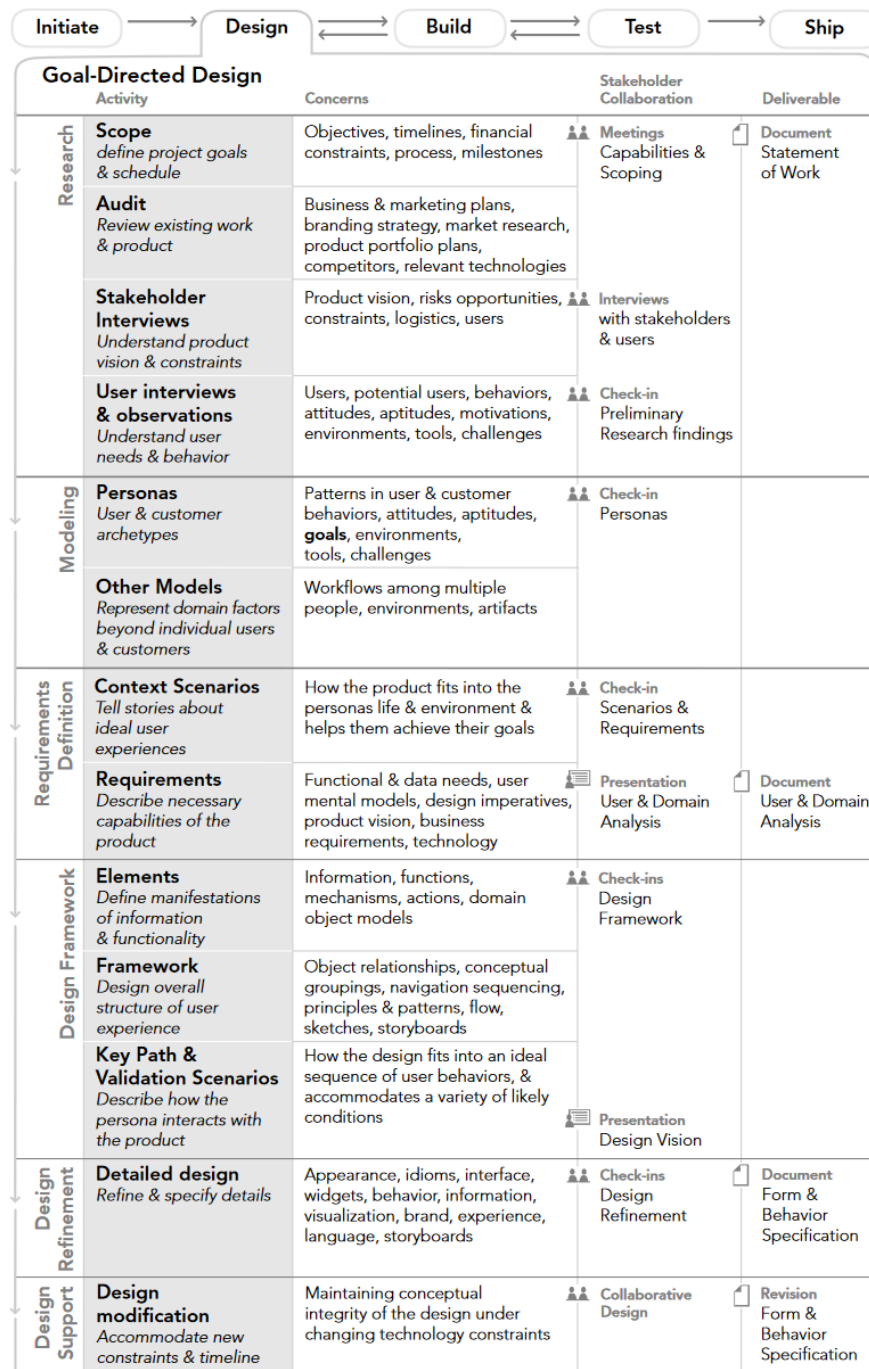
Käyttäjälähtöiseen käyttöliittymäsuunnitteluun on olemassa useita valmiita prosessimalleja. Tässä kappaleessa kerrotaan yleistasolla kolmesta eri prosessimallista: Goal-Directed Design, GUIDe sekä suunnitteluajattelu.

### 2.8.1 Goal-Directed Design

Goal-Directed Design (GDD), vapaasti suomennettuna tavoiteohjautunut suunnittelu, on Alan Cooperin kehittämä käyttäjäkeskeinen prosessimalli, jossa suunnittelu edeltää toteutusta. Tavoiteohjautuneessa suunnittelussa käyttäjien tarpeet ja tavoitteet ovat ohjelmistokehitysprosessin keskiössä. Prosessissa tuodaan tuotteen loppukäyttäjät mukaan sen kehitysprosessiin ja pyritään näin tekemään tuotteen käytöstä loppukäyttäjälle mieluisampi kokemus.

Tavoiteohjautuneessa suunnittelussa suunnittelu tapahtuu ennen toteutusta. Suunnittellessaan GDD-prosessia Cooper halusi poiketa vanhoista ajattelumalleista, missä ohjelmointi joko edelsi suunnittelua tai ohjelmointi tapahtui samanaikaisesti suunnittelun kanssa. Tavoiteohjautunut suunnitteluprosessi tarjoaa kehittäjille vaihtoehdoisen tavan toiminnallisuuden toteuttamiseen, Cooperin mukaan käyttäjien ja käyttäjäpersoonien tavoitteiden huomointi käyttöliittymän suunnittelussa on huomattavasti parempi tapa ratkaista suunnitteluongelmia [5, p. 152]. Tavoiteohjautuneessa suunnittelussa on myös tärkeää suunnittelijan ja toteuttajan erotus toisistaan. Ohjelmoija taipuu liian helposti pistämään etusijalle toteutuksen helppouden: suunnitellaan järjestelmä helpoksi toteuttaa sen sijaan että suunniteltaisiin käyttäjän ehdoilla.

Prosessimalli itsessään koostuu ylätasolla kuudesta eri vaiheesta: tutkiminen (*engl. research*), mallintaminen (*engl. modeling*), vaatimusten määrittely (*engl. requirements definition*), kehyksen suunnittelu (*engl. design framework*), hienosäätö (*engl. design refinement*) ja kehityksen tuki (*engl. design support*) [2, p. 24].



Kuva 2.3: GDD prosessimalli tiivistettynä (Cooper, Reimann, Cronin, 2007).

Tutkimisvaiheessa määritellään projektin laajuus sekä aikataulu, tutkitaan vastaavia olemassa olevia tuotteita, haastatellaan sidosryhmiä sekä tutkitaan ja haastatellaan käyttäjiä. Käyttäjiä tarkkailemalla saadaan dataa käyttötavoista, joilla suunnitellaan käyttöä.

niteltavaa tuotetta voitaisiin käyttää. Käyttötavoista voidaan tunnistaa käyttäjien tavoitteet sekä motivaatio. Tätä dataa käytetään hyödyksi prosessin seuraavassa vaiheessa.

Mallintamisvaiheessa käyttäjistä kerätystä datasta muodostetaan tuotteen käyttäjäpersoonat. Käyttäjäpersoonat ovat keskeinen osa GDD-prosessimallia, joista kerrottiin aikaisemmin tarkemmin luvussa 2.4. GDD tekee erotuksen käyttäjän tavoitteiden (*engl. Goals*) ja tehtävien (*engl. Tasks*) välillä. Käyttäjät pyrkivät saavuttamaan tavoitteensa tehtävien avulla, tavoite pysyy aina samana mutta sen saavuttamiseen tarvittavat tehtävät voivat muuttua [5, p. 150]. On siis tärkeämpää kehittäjille keskittyä käyttäjien tavoitteisiin, siihen mitä heidän on tarkoitus järjestelmää käyttämällä saavuttaa, eikä siihen mitä tehtäviä he suorittavat tavoitteen saavuttamiseksi. Tavoite on käyttäjän lopullinen päämäärä järjestelmää käytettäessä, ja tavoitteen saavuttaakseen käyttäjä suorittaa tehtäviä järjestelmällä [5, p. 150]. Cooper varoittaa kehittäjien keskittyvän liian helposti tehtäviin ja unohtavan käyttäjän lopullisen tavoitteen, mikä puolestaan johtaa turhauttavaan käyttökokemukseen.

Vaatimusten määrittelyvaiheessa tuodaan yhteen persoonat ja tuote, jota suunnitellaan. Tässä vaiheessa luodaan käyttöskenaarioita, joissa persoonat käyttäisivät tuotetta.

Tavoiteohjautuneessa suunnittelussa on myös tärkeä ottaa huomioon muidenkin kuin käyttäjien tavoitteet. Muita tavoitteita, jotka suunnittelijan on huomioitava ovat asiakkaiden tavoitteet, yritysten ja organisaatioiden tavoitteet sekä tekniset tavoitteet. Nämä tavoitteet on tiedostettava ja pidettävä mielessä, mutta niiden ei saa antaa sanella suunnittelun suuntaa, käyttäjän tavoitteet on priorisoitava.

Asiakkailla tarkoitetaan niitä, jotka ostavat tuotteen mutteivat ole sen käyttäjiä. Nämä voivat olla esimerkiksi kuluttaja-asiakkaiden puolella vanhempia, jotka ovat huolissaan tuotteen turvallisuudesta. Yritysassiakkaat ovat yleensä yritysten IT-vastaavia, jotka ovat kiinnostuneita tuotteen tietoturvasta, ylläpidosta ja integraa-

tion helppoudesta.

### 2.8.2 GUIDe

GUIDe (*Goals–User Interface Design – Implementation*) on Sari A. Laakson ja Karri-Pekka Laakson käyttöliittymäsuunnittelua varten kehittämä prosessimalli [12]. GUIDe on hyvin samankaltainen edellä mainitun Goal Directed Design -prosessimallin kanssa, molemmissa käyttöliittymäsuunnittelu tapahtuu varhaisessa suunnitteluvaiheessa projektin alussa ennen kuin käyttöliittymää lähdetään toteuttamaan. Molempien prosessimallien keskiössä on myös loppukäyttäjien huomiointi prosessin alusta asti. GUIDe:ssa käyttöliittymä suunnitellaan käyttötapausten pohjalta. Tämän etuna on se, että järjestelmää voidaan suoraan testata samoja käyttötapauksia vasten. [12]

Kantavana ajatuksena on käyttäjäsuunnittelussa samaa menettelyä kuin testauksessa ja järjestelmän todellisessa käytössä: käyttöliittymä suunnitellaan käyttötilanne asetelmien pohjalta.

GUIDe koostuu kolmesta eri vaiheesta [12]:

- Vaatimusten kartoittaminen
- Käyttöliittymäsuunnittelu simulointipohjaisesti
- Käyttöliittymän toteuttaminen

GUIDe prosessin ensimmäinen vaihe ennen varsinaista käyttöliittymäsuunnittelua on käyttäjien tavoitteiden määrittely (Goals), joista voidaan johtaa järjestelmän vaatimukset. Tavoitteiden määrittely tehdään käyttäjiä haastatteleamalla ja tarkkailemalla heidän nykyistä työnkulkuaan, selvittämällä mikä on käyttäjien oikea tavoite, mitä he pyrkivät saavuttamaan.

Prosessin toisessa vaiheessa käyttöliittymäsuunnittelija alkaa suunnittelemaan käyttöliittymää käyttötapauksia simuloimalla (User Interface Design). Toista vai-

hetta iteroidaan kunnes kaikki järjestelmän käyttötapaukset on saatu sisällytettyä käyttöliittymään ja ne on testattu toimiviksi.

Toisen vaiheen tuloksena on dokumentaatio ja näyttökuvia suunnitellusta käyttöliittymästä.

Suunnitteluprosessin viimeinen vaihe on järjestelmän toteuttaminen (Implementation). Tähän vaiheeseen sisältyy itse toteutuksen suunnittelu, ohjelmointi ja sen testaus.

### 2.8.3 Suunnitteluajattelu

Suunnitteluajattelu *engl. Design thinking* on GDD:n ja GUIDen tapaan käyttäjälähtöinen prosessimalli, mitä voidaan käyttää käyttöliittymien suunnittelussa.

Suunnitteluajattelulla on pitkä historia. Menetelmän juuret ovat John E. Arnoldin vuonna 1959 julkaistussa kirjassa (*engl. Creative Engineering*) [13], mutta menetelmää ei kutsuttu vielä tuolloin suunnitteluajatteluksi. Suunnitteluajattelu -termi esiintyi ensimmäisen kerran Peter Rowen vuonna 1987 julkaistussa *Design Thinking* kirjassa, joka käsitteli suunnitteluajattelun hyödyntämistä rakennusarkkitehtuurissa [14].

Suunnitteluajattelu on nykyään laajalti käytössä ja siitä on useita eri variaatioita. Prosessi kuvataan kuitenkin useimmiten koostuvan viidestä vaiheesta [15]:

1. Ymmärtäminen, (*engl. Emphasize*)
2. Määrittely, (*engl. Define*)
3. Ideointi, (*engl. Ideate*)
4. Prototyypaus, (*engl. Prototype*)
5. Testaus, (*engl. Test*)

Prosessin ymmärtämisympäristössä tutkitaan ja yritetään ymmärtää käyttäjiä, suunnittelijoiden on pystyttävä asettumaan käyttäjän asemaan. Määrittelyvaiheessa käyttäjistä kerätty data yhdistetään ja siitä määritetään ongelma, jota pyritään ratkomaan ideointivaiheessa.

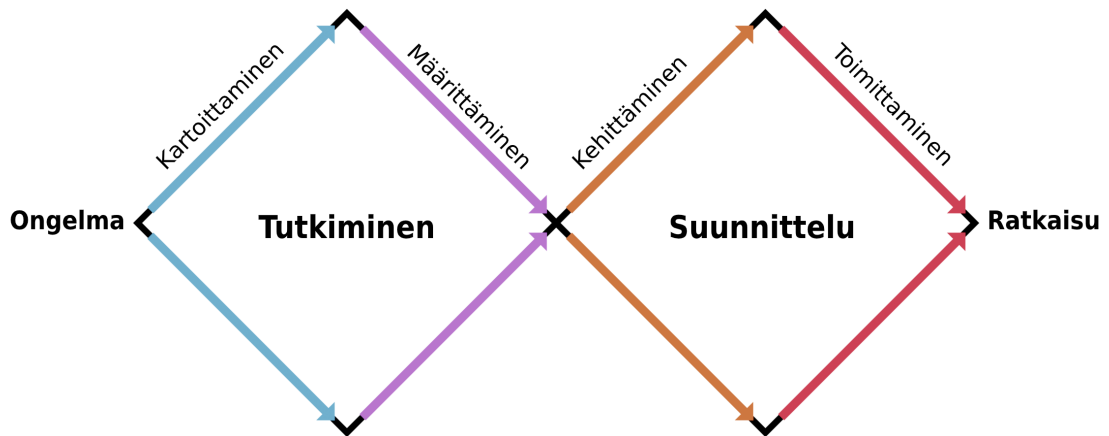
Ideointivaiheessa on tärkeää pyrkiä löytämään mahdollisimman monta ideaa ongelman ratkomiseksi ja onkin hyvin tärkeää, että tässä vaiheessa ideoita ei vielä kritisoida, jotta pystytään löytämään mahdollisimman luovia ratkaisuja. Vasta prototyypausvaiheessa ideat käydään läpi ja katsotaan, mikä niistä on eniten toteuttamisen arvoinen.

Testausvaiheessa palataan käyttäjiin ja kerätään heiltä palautetta prototyypistä. Palautteen avulla voidaan arvioida, mikä on seuraava toimenpide suunnittelussa. Jos ratkaisu sai paljon negatiivista palautetta, voidaan olettaa että käyttäjien tarpeet olivat arvioitu väärin ja on tarvetta palata takaisin ymmärtämisympäristöön. Jos palaute oli suurimmaksi osaksi positiivista, voidaan palata ideointivaiheeseen pohtimaan ratkaisuja löydettyihin käytettävyyden ongelmiin. Tätä ideointi-prototyypaus-testausketjua iteroidaan, kunnes kaikki käytettävyyden ongelmat on korjattu.[15]

Tämän jälkeen voidaan siirtyä itse käyttöliittymän toteutukseen. Tämä toteutusvaihe (*engl. Implement*) sisällytetään suunnitteluajattelun joissain variaatioissa kuudenneksi ja viimeiseksi vaiheeksi [16].

Yksi tunnetuimmista variaatioista on British Design Councilin vuonna 2005 kehittämä suunnitteluprosessi; kaksoistimantti (*engl. Double Diamond*). Kaksoistimantti menetelmässä vuorotellaan divergenttia ja konvergenttia ajattelua prosessin eri vaiheissa [17].





Kuva 2.4: Kaksoistimantti prosessimallin vaiheet.

Kaksoistimantti voidaan jakaa kahteen päävaiheeseen; tutkimisvaihe ja suunnitteluvaihe. Mallin neljä sisempää vaihetta ovat [17]:

1. Kartoittaminen, (*engl. Discover*)
2. Määrittäminen, (*engl. Define*)
3. Kehittäminen, (*engl. Develop*)
4. Toimittaminen, (*engl. Deliver*)

Ensimmäisen kahden vaiheen on tarkoitus selvittää, että rakennetaan varmasti oikeaa asiaa. Kahden jälkimmäisen vaiheen taas varmistaa, että asia rakennetaan oikein. Kartoittamis- ja kehittämisvaiheissa suunnittelijoiden ajattelun on oltava divergenttiä, heidän pitää pystyä luomaan useita ideoita ja vaihtoehtoisia ratkaisuja. Määrittämis- ja toimittamisvaiheissa edellytetään konvergenttiä ajattelua, suunnittelijoiden on pystyttävä keskittymään yhteen ideaan tai ratkaisuun.

Kartoittamisvaihe alkaa projektin käynnistyessä. Prosessin kartoittamisvaiheessa haetaan laajasti inspiraatiota suunnittelutyöhön. Tähän vaiheeseen kuuluu markkinoiden tutkiminen ja käyttäjien tarpeiden löytäminen. On myös tärkeää huomioida

kerätyn tiedon hallinta ja varmistaa, että tietoa jaetaan suunnittelijoiden välillä. Tiedon keräämiseen ja esittämiseen voidaan käyttää apuna aikaisemmin esiteltyjä metodeja ja työkaluja, kuten empatiakartat ja persoonat. [17]

Määrittämisvaiheessa käydään läpi sekä yhdistetään ja tulkitaan ensimmäisessä vaiheessa kerättyä informaatiota. Tarkoituksena on heittää pois tarpeeton informaatio ja asiat mitä ei projektiin varatuilla resursseilla kyetä saavuttamaan. Vaiheen tuloksena määritetään se alue tai ongelma, minkä kehittämiseen keskitytään.

Kehittämisvaihe alkaa ideoinnilla. Tarkoitus on löytää mahdollisimman monta erilaista ratkaisua määritettyyn ongelmaan. On jälleen tärkeää olla kritisoiden ideota tässä vaiheessa. Kun on kerätty tarpeeksi ideota, voidaan siirtyä niiden evaluointiin. Tavoitteena on valita paras idea ongelman ratkaisuun. Tärkeitä asioita ratkaisun valinnassa on sen toteutettavuus sekä se, kuinka hyvin se ratkaisee alkuperäisen ongelman. [18]

Lopulta kaksoistimantti prosessimallin viimeisessä vaiheessa voidaan ryhtyä toteuttamaan valittua ratkaisua. Toimittamisvaihe koostuu ratkaisun prototyypaamisesta ja rakentamisesta, minkä jälkeen ratkaisulle suoritetaan vielä lopulliset testit [18]. Testien tulosten perusteella voidaan joko jälleen palata iteroimaan prosessin aikaisempia vaiheita ja pyrkiä etsimään parempi ratkaisu, tai jos tulokset olivat hyviä, voidaan tuote hyväksyä ja julkaista. Julkaisun jälkeen tähän vaiheeseen kuuluu myös projektin evaluointi.

# 3 Käytettävyyden arviointi

Käytettävyyttä arviointiin on useita eri menetelmiä. Nielsenin mukaan käytettävyyttä voidaan arvioida ja mitata eri metriikoilla [19]. Mitattaessa voidaan arvioida käyttöliittymällä suoritettua tehtävää:

- onnistumisprosenttia
- suorittamiseen kulunutta aikaa
- virheiden määrää
- käyttäjän tyytyväisyyttä

Mitattaessa on tärkeää käyttää tarpeeksi suurta testikäyttäjäryhmää. Siitä mikä on sopiva määrä, on useampia eri mielipiteitä ja tutkimuksia, mutta Nielsen painottaa että jo viidellä testikäyttäjällä löydetään merkittävä osa testattavan järjestelmän käytettävyysongelmista [20].

Käytettävyyden arviointi voidaan aloittaa käyttöliittymäsuunnittelun alkuvaiheessa jo ensimmäisten paperiprototyyppien kanssa [11]. Aikaisen aloittamisen hyötyä on palautteen saaminen käytettävyydestä ilman että kehittäjien tarvitsee käyttää aikaa käyttöliittymän toteuttamiseen.

Paperiprototyyppien testauksessa haasteena on, se että paperit eivät sisällä toiminnallisuutta. Jos niillä halutaan mitata muutakin kuin käyttöliittymän ulkoasua, testaustilanteessa suunnittelijan on simuloitava järjestelmän ja käyttäjän vuorovaikutusta.

## 3.1 Arviointimenetelmät

Käytettävyyttä voidaan arvioida useilla eri menetelmillä. Arviointimenetelmät poikkeavat toisistaan niiden vaatimien resurssien, kuten ajan ja testaaajien mukaan. Seuraavissa alaluvuissa kerrotaan yleisimmin käytetyistä käytettävyyden arviointimenetelmistä pääpiirteittäin.

### 3.1.1 Heuristinen evaluointi

Heuristisessa evaluoinnissa pieni määrä arvioijia etsii ja kommentoi käyttöliittymän ongelmia, käyttäen apuna jotain sovittua listaa käytettävyyden säännöistä ja periaatteista. Näitä listoja kutsutaan heuristiikoiksi. Menetelmää käytettäessä on tärkeää, että arvioijat ovat käytettävyyteen perehtyneitä asiantuntijoita, eivätkä esimerkiksi tavallisia käyttäjiä. Heuristinen testaus löytää yleensä sellaisia ongelmia, joita ei käyttäjätesteillä löydettäisi, koska käyttäjät eivät ole käytettävyyden asiantuntijoita.

Arvioijien on osattava perustella löytämänsä ongelmat. Tätä varten heille on tärkeää antaa arvioinnin tueksi heuristinen lista. Nykyään heuristisessa evaluoinnissa tunnetuin ja käytetyin heuristinen lista on Jakob Nielsenin ja Rolf Molichin kehittämät kymmenen heuristiikkaa [21]:

- **Järjestelmän tilan selkeys:** järjestelmän pitää antaa palautetta käyttäjälle siitä, missä tilassa järjestelmä on.
- **Järjestelmän ja oikean maailman vastaavuus:** järjestelmän tulee kommunikoida käyttäjän kanssa käyttäen käyttäjälle tuttuja termejä ja konsepteja.
- **Käyttäjän kontrolli ja vapaus,** käyttäjille pitää jättää selkeästi merkitty tapa palautua virheistä.

- **Johdonmukaisuus ja standardit**, käyttäjien ei pitäisi joutua pohtimaan sanojen, tilanteiden tai toimintojen merkityksiä.
- **Virheiden estäminen**, on parempi estää virheitä kuin näyttää käyttäjille virheviestejä.
- **Tunnistaminen muistamisen sijaan**, käyttäjien muistia ei tulisi kuormittaa ja järjestelmän käytön ohjeiden tulisi olla näkyvissä tai tarvittaessa helposti saatavilla.
- **Käytön joustavuus ja tehokkuus**, järjestelmän tulisi sopia myös tehokäyttäjille ja sallia näiden suorittaa haluamiaan toimintoja nopeilla tavoilla.
- **Estetiikka ja minimalistinen suunnittelu**, käyttäjälle tarpeettoman tiedon näyttämistä pitäisi välttää.
- **Auta käyttäjiä tunnistamaan, diagnosoimaan ja palautumaan virheistä**, virheviestien tulisi olla selkokieliä, kertoa käyttäjälle tarkalleen mikä virhe on ja ehdottaa siihen ratkaisua.
- **Opastus ja ohjeet**, vaikka on tärkeää että käyttäjät pystyvät käyttämään järjestelmää ilman ohjeita, on näiden oltava helposti saatavilla tarvittaessa. Ohjeiden tulisi olla helposti etsittävät, keskittyä käyttäjän tehtäviin, listata konkreettisia askeleita eikä olla liian pitkiä.

Jokainen asiantuntija käy yksinään läpi järjestelmää heuristinen lista apunaan ja merkitsee ylös sieltä löytämänsä käytettävyyden ongelmat. Sen jälkeen asiantuntijoiden on arvioitava ja luokiteltava löydettyjen ongelmien vakavuus. Löydettyjen virheiden vakavuus on kolmen tekijän yhdistelmä: *yleisyys*, eli kuinka usein virhe esiintyy; *vaikutus*, kuinka helposti käyttäjien onnistuu palautua virheestä ja *sitkeys*, voivatko käyttäjät välttää virhettä kerran siitä palaututtuaan vai vaivaako virhe

käyttäjiä toistuvasti. Arvioijat luokittelevat löytämänsä käytettävyyden ongelmat niiden vakavuuden mukaan seuraavalla asteikolla [22]:

- 0 = Ei käytettävyysongelma
- 1 = Kosmeettinen ongelma, korjataan jos on ylimääräistä aikaa
- 2 = Lievä käytettävyysongelma, korjaaminen ei ole prioriteetti
- 3 = Vakava käytettävyysongelma, tärkeä korjata mahdollisimman pian
- 4 = Katastrofaalinen ongelma, pakko korjata ennen kuin tuote voidaan julkaista

Kun jokainen arvioija on tehnyt luokittelun omalta osaltaan, voi heuristisen evaluoinnin järjestäjä listata löydetty ongelmat ja laskea keskiarvot eri ongelmien vakavuusluokituksille. Tässä kohtaa voidaan myös kommunikoida asiantuntijoiden kanssa, jos jokin ongelma on listattu epäselvästi. Seuraavaksi testin järjestäjä kerää kaikkien asiantuntijoiden ongelmalistaukset, käy nämä läpi ja laskee keskiarvot ongelmille. Löydetty pahimmat käytettävyyden ongelmat on korjattava ennen seuraavan evaluoinnin suorittamista.

Vaikka heuristinen evaluointi on tehokas tapa löytää käytettävyyden ongelmia, on sen soveltamisessa tärkeä huomioida sen ongelmat. Ollakseen hyödyllinen, evaluoinnin suorittajien on oltava osaavia käytettävyyden asiantuntijoita. Asiantuntijoiden käyttö on myös kallista, mutta on pidettävä mielessä, että löytämättä jääneiden ongelmien korjaaminen myöhemmin käy myös kalliiksi. Evaluointi ei myöskään aina onnistu paljastamaan kaikkia ongelmia, eikä näin ollen korvaa käyttäjättestausta, joka suoritetaan todellisilla loppukäyttäjillä. Heuristinen evaluointi ei myöskään anna tietoa siitä, miten korjata löydetty ongelmat.

### 3.1.2 Kognitiivinen läpikäynti

Kognitiivinen läpikäynti kehitettiin 1990-luvulla menetelmäksi löytää käytettävyyden ongelmia. Kuten heuristisessa evaluoinnissa, kognitiivisen läpikäynnin suorittamiseen ei tarvitse loppukäyttäjiä, vaan tuotteen käyttöliittymää ja käytettävyyttä voidaan testata ilman oikeaa loppukäyttäjää. Kognitiivisessa läpikäynnissä arvioinnin painopiste on yksittäisissä tehtävissä ja käyttäjäpoluissa, toisin kuin heuristisessa evaluoinnissa, jolla arvioidaan järjestelmän käytettävyyttä kokonaisuutena. Kognitiivista läpikäyntiä voi käyttää pelkkien prototyyppien avulla. [23]

Menetelmän tavoitteena on selvittää, kuinka hyvin ja helposti järjestelmän käytön oppii käyttämällä sitä. Kognitiivisen läpikäynnin takana on ajatus siitä, että käyttäjät haluavat oppia käyttämään järjestelmää suorittamalla tehtäviä sillä, sen sijaan että he oppisivat järjestelmän käyttöä lukemalla manuaalia.

Kognitiivisessa läpikäynnissä testaaaja asettautuu loppukäyttäjän rooliin ja simuloi realistista käyttötilannetta suorittamalla järjestelmällä loppukäyttäjille tyypillisiä tehtäviä. Testaaaja koittaa näin selvittää ovatko käyttäjien tavoitteet ja tehtävät saavutettavissa järjestelmän avulla, ja osaako käyttäjä valita oikeat toiminnot niiden saavuttamiseksi. [23]

Testaaajan on jokaisen tehtävän kohdalla kysyttävä itseltään seuraavat kysymykset [24]:

1. Yrittääkö käyttäjä saavuttaa oikean tavoitteen?
2. Huomaako käyttäjä, että oikea toiminto on saatavilla?
3. Osaako käyttäjä yhdistää kyseisen toiminnon tavoitteeseensa?
4. Jos käyttäjä suorittaa oikean toiminnon, saako hän palautetta siitä, että hän etenee kohti tavoitettaan?

Jos testaaaja vastaa johonkin näistä kieltävästi on hän löytänyt käytettävyysongelman, joka tulee korjata.

Kognitiivista läpikäyntiä varten pitää ensin määrittää keitä todelliset käyttäjät ovat ja mitä tehtäviä he suorittavat järjestelmällä. Tehtävän ympärille on suositeltavaa kirjoittaa uskottava ja realistinen skenaario, jotta testaaajien on helpompi samaistua käyttäjään. Järjestelmän testaukseen sopivia tehtäviä valitaan esimerkiksi sen perusteella kuinka usein käyttäjät todennäköisesti suorittavat kyseisiä tehtäviä: yleisimmät tehtävät kannattaa priorisoida testauksessa. On myös hyvä yrittää valita tehtäviä, jotka kattavat järjestelmää mahdollisimman laajasti, jotta sen kaikki osat tulevat testattua.

Menetelmän etuna ovat sen matalat kustannukset ja suorittamisen helppous verrattuna käytettävyytestaukseen. Kognitiivista läpikäyntiä voi käyttää käyttöliittymän suunnitteluprosessin alkuvaiheista asti ja testaaajaksi käy esimerkiksi yksi kehittäjistä. On kuitenkin tärkeää, että testaaaja osaa asettautua käyttäjän rooliin. Tämän tueksi on hyvä olla luotuna käyttäjäpersoonaa sekä käyttöskenaarioita jotka auttavat testaaajan samaistumisessa käyttäjään. Joskus on myös eduksi käyttää testaajana täysin ulkopuolista henkilöä, jotta saadaan mielipiteitä myös henkilöiltä, joille järjestelmä ei ole tuttu.

### 3.1.3 Pluralistinen läpikäynti

Pluralistinen läpikäynti on käytettävyytestauksen ja kognitiivisen läpikäynnin yhdistelmä. Siinä tuodaan yhteen edustajia niin käyttäjistä, kehittäjistä kuin suunnittelijoista keskustelemaan ryhmässä testattavan järjestelmän käytettävyysongelmista ja niiden korjausehdotuksista. Pluralistisesta läpikäynnistä on eniten hyötyä suunnitteluprosessin alkuvaiheessa, sillä silloin kehittäjät ja suunnittelijat saavat välitöntä palautetta suoraan käyttäjiltä käyttöliittymäsuunnitteluprosessin alussa.

Menetelmän viisi määrittelevää piirrettä ovat [25]:

1. Samaa läpikäyntisessioon osallistuu käyttäjiä, kehittäjiä sekä käytettävyyden asiantuntijoita.



2. Testattavaa järjestelmää edustetaan paperiversiolla.
3. Kaikki osallistujat asettautuvat käyttäjän rooliin.
4. Osallistujat kirjoittavat ylös mitä toimintoja he suorittaisivat annettujen tehtäviä tekemiseksi.
5. Osallistujat keskustelevat yhdessä ratkaisuisista joihin he päätyivät. Keskustelun aloittaa läpikäynnin järjestäjä. Järjestäjän jälkeen käyttäjät kertovat omat ratkaisunsa ja vasta sen jälkeen kehittäjät sekä asiantuntijat saavat ilmaista mielipiteensä.

Pluralistisen läpikäynnin järjestämiseen riittää pelkkä paperiprototyyppi järjestelmästä, joten sen käyttö käytettävyyden arvioinnissa ja suunnittelussa on hyvin kustannustehokasta. Menetelmä paljastaa hyvin käytettävyyden ongelmat ja antaa myös näihin korjausehdotuksia. Käyttäjien mukanaolo läpikäynnissä myös varmistaa, että järjestelmää rakennetaan heitä varten.

### 3.1.4 System Usability Scale

SUS-kysely [26] rakentuu kymmenestä kysymyksestä, joissa jokaisessa on viisi vastausvaihtoehtoa. Testaajat vastaavat kyselyyn käytettävyydestin lopuksi. Vastausvaihtoehdot ilmaistaan viisiportaisella Likert-asteikolla, jossa vastaajat valitsevat seuraavista vaihtoehdoista heidän mielipidettään eniten vastaavan:

1. Täysin eri mieltä
2. Eri mieltä
3. Ei mielipidettä
4. Samaa mieltä

## 5. Täysin samaa mieltä

SUS:ssä kysytyt kymmenen kysymystä ovat vapaasti suomennettuina seuraavat:

1. Käyttäisin tätä järjestelmää usein.
2. Järjestelmä oli mielestäni tarpeettoman monimutkainen.
3. Järjestelmää oli mielestäni helppo käyttää.
4. Luulen, että tarvitsen teknisen henkilön tukea jotta voisin käyttää tätä järjestelmää.
5. Järjestelmän eri osat toimivat mielestäni hyvin yhteen.
6. Mielestäni järjestelmässä oli liian paljon eri lailla toimivia asioita.
7. Luulen, että useimmat oppivat käyttämään tätä järjestelmää hyvin nopeasti.
8. Järjestelmää oli mielestäni hyvin hankala käyttää.
9. Tunsin itseni hyvin itsevarmaksi käyttäessäni järjestelmää.
10. Minun piti opetella paljon asioita, ennen kuin pystyin käyttämään tätä järjestelmää.

Kysymysten järjestystä ei pidä vaihtaa, sillä pisteet lasketaan eri tavalla eri kysymyksille. Parittomissa kysymyksissä annetuista vastauksista vähennetään yksi piste, parillisissa kysymyksissä taas pisteet lasketaan vähentämällä käyttäjän antamat pisteet viidestä. Tämä muuntaa kaikki pistearvot välille 0-4; mitä korkeampi niin sitä parempi käytettävyys. Lopullinen käytettävyyden arvosana saadaan laskemalla kaikkien vastausten pisteet yhteen ja kertomalla saatu tulos 2,5:llä.

Lopullinen pisteytys on 0-100 välillä. Mitä korkeammat pisteet järjestelmä tai tuote SUS-kyselyssä saa, niin sitä parempi käytettävyys sillä voidaan ajatella olevan. Vaikka SUS ei itsessään kerro mitä pisteytys ilmaisee, sen laajan käytön ansiosta muodostunutta dataa tutkimalla voidaan muodostaa pisterajat käytettävyydelle. Tutkimusten perusteella SUS:n keskiarvo on 68 pistettä. Tästä voidaan arvioida, että

sen alapuolelle putoavat järjestelmät ovat käytettävyydeltään keskiarvoa huonompia ja yli 68 pistettä saaneet taas ovat käytettävyydeltään keskiarvoa parempia. [27]

SUS Pisteet	Arvosana	Arvosanan kuvaus
> 80,3	A	Loistava
68 - 80,3	B	Hyvä
68	C	OK
51 - 67	D	Huono
< 51	F	Surkea

Kuva 3.1: SUS pisteytys [27].

SUS:in käytössä on useita etuja. Se on nopea ja helppo järjestää käytettävyydentestauksen ohessa ja myös tarvittaessa helposti skaalattavissa. Sen avulla on myös helppo muuntaa abstrakti käytettävyyden konsepti numeroarvoksi, jonka avulla on helpompi kommunikoida järjestelmän käytettävyyden laatua sidosryhmille. Sitä voi käyttää myös projektin alussa prototyyppien kanssa, mutta tällöin saatua käytettävyyden arvosanaa ei voi pitää luotettavana

SUS:in ongelmana on se, että se ei yksinään paljasta mikä testattavan järjestelmän käytettävyydessä oli testaaajien mielestä pielessä.

Tutkimusten mukaan SUS on luotettava työkalu käytettävyyden mittaukseen riippumatta testaaajien määrästä; sitä voi käyttää vaikka vain kahdella testaaajalla. Suuremmalla testaaajien määrällä saadaan kuitenkin luotettavampia tuloksia. [27]

### 3.1.5 Käytettävyydentestaus

Ennen käytettävyydentestin järjestämistä on tärkeää suunnitella sessio ja kirjoittaa ylös protokolla testausta varten. Testin suunnittelussa on tärkeää vastata seuraaviin kysymyksiin [1, p.170 ]:

1. Testin tavoite, mitä testillä halutaan mitata?
2. Milloin ja missä testi tapahtuu?

3. Kuinka kauan jokainen testaussessio kestää?
4. Minkälaista tietokonetukea tarvitaan?
5. Mitä ohjelmia pitää olla valmiina testiä varten?
6. Missä tilassa järjestelmän on oltava testin alussa?
7. Mitkä ovat realistisia vasteaikoja järjestelmälle?
8. Ketkä toimivat testin järjestäjinä?
9. Ketkä ovat testaaajia ja miten heidät tavoitetaan?
10. Kuinka monta testaaajaa tarvitaan?
11. Mitä tehtäviä testaaajat suorittavat?
12. Mitkä ovat tehtävien onnistumisen kriteerit?
13. Mitä mahdollisia apuja testaaajille tarjotaan?
14. Kuinka paljon järjestäjät saavat tarjota apua testaaajille?
15. Mitä dataa kerätään ja miten sitä analysoidaan?
16. Mitkä ovat hyvän käyttöliittymän kriteerit?

Käytettävyydestestauksen perimmäisenä ideana on selvittää, kuinka hyvin testattava järjestelmä vastaa loppukäyttäjien tarpeita. Käytettävyydestestaus suoritetaan oikeilla loppukäyttäjillä, tai henkilöillä jotka vastaavat näitä mahdollisimman hyvin järjestelmän käytön kannalta tärkeissä piirteissä.

Käytettävyydestestin järjestäjä tarkkailee testaaajaa tämän suorittaessa tehtäviä. Tarkkailijoita voi olla useampikin, mutta tarkkailtavia aina vain yksi. Käytettävyydestesti myös yleensä taltioidaan videolle, tai vähintäänkin testaustilanteen audio nauhoitetaan. Tallenteiden uudelleenkatseilu mahdollistaa sellaisten virheiden havainnoinnin, jotka jäivät testaustilanteen tarkkailijalta huomaamatta.

Käytettävyydestä yksistään ei takaa täydellistä käytettävyyttä; joskus saattaa olla resurssillisesti tehokkaampaa suorittaa kognitiivisia läpikäyntejä tai heuristisia analyysejä [28, p.26 ]. Tämä pätee varsinkin projektin alkuvaiheessa ja prototyyppien testauksessa. Toisaalta, Rubin ehdottaa myöskin käytettävyydestästä käytettäväksi jo alkuvaiheessa webbisovellusten kehittämisessä ja tämän avulla välttämään suurimpia virheitä käyttöliittymän peruselementtien suunnittelussa jo heti projektin alkuvaiheessa [28, p.29 ]. Rubin myös suosittelee kysymään mahdollisia parannusehdotuksia testikäyttäjältä, esimerkiksi miten tämä parantaisi käyttöliittymän alueita ja mitkä elementit aiheuttivat hänelle ongelmia käyttöliittymää käyttäessä. On kuitenkin huomioitava, että testikäyttäjä ei ole käytettävyyden asiantuntija, eikä näin ollen hänen mielipiteitään ja parannusehdotuksiaan voi pitää tiukkoina ohjenuorina siitä, mitä pitäisi noudattaa.

# 4 Case: CT Publisher

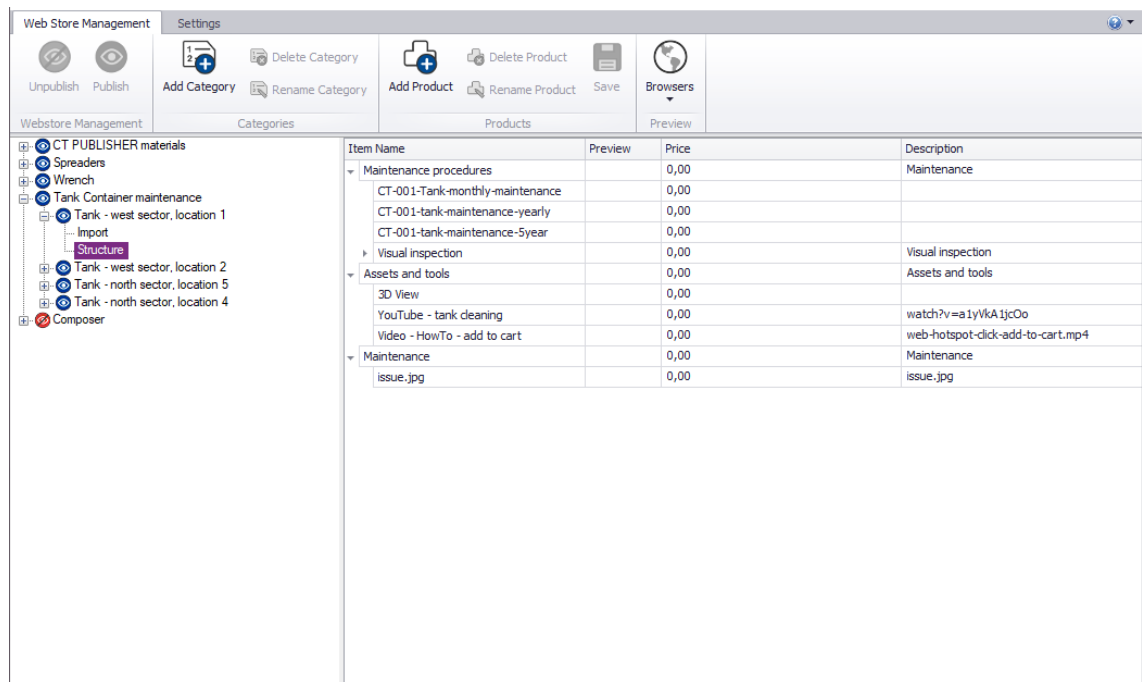
## 4.1 Tuotteen taustaa

Tuote, jonka käyttöliittymää tässä työssä suunnitellaan, on työn toimeksiantajan ATR Soft:in kehitteillä oleva CT Publisher tuotteen web sovellusversio.

CT Publisher -tuotteen päätarkoitus on toimia alustana monimutkaisten teknisten dokumenttien julkaisuun eri käyttäjäryhmien tarkoituksiin. Tuotetta voi käyttää yksinomaan asiakasyrityksen sisäiseen dokumentaation levitykseen tai myös varaosien myyntialustana, josta asiakkaat voivat selata tuotteita hierarkisesti ja räjäytyskuvien avulla löytää helposti tarvitsemansa tuotteen. Räjäytyskuvien tarkoitus on selkeyttää asiakkaalle mahdollisesti monimutkaisen tuotteen varaosan tilausta ja näin vähentää tarvetta väärän osan tilaamisesta johtuville yhteydenotoille ja palautuksille. Alustalla on myös mahdollisuus julkaista manuaaleja ja muita vastaavia dokumentteja asiakkaiden käyttöön.

CT Publisherin tarkoitus on myös vähentää käyttäjien tarvetta tietää haluamiensa varaosien tuotenumeroita ja syöttää niitä järjestelmään. Tuotteella on myös tavoitteena vähentää asiakkaan tarvetta soittaa varaosamyyntiin, mahdollistaen varaosien myynnin kellon ympäri. Tavoitteena on, että alustalla olevat tiedot olisivat aina ajan tasalla.

Tuotetta käyttävän yrityksen huoltoteknikoille alusta hyödyttäisi myös viallisen komponentin varaosan saatavuuden selvittämistä: tavoitteena olisi, että teknikko



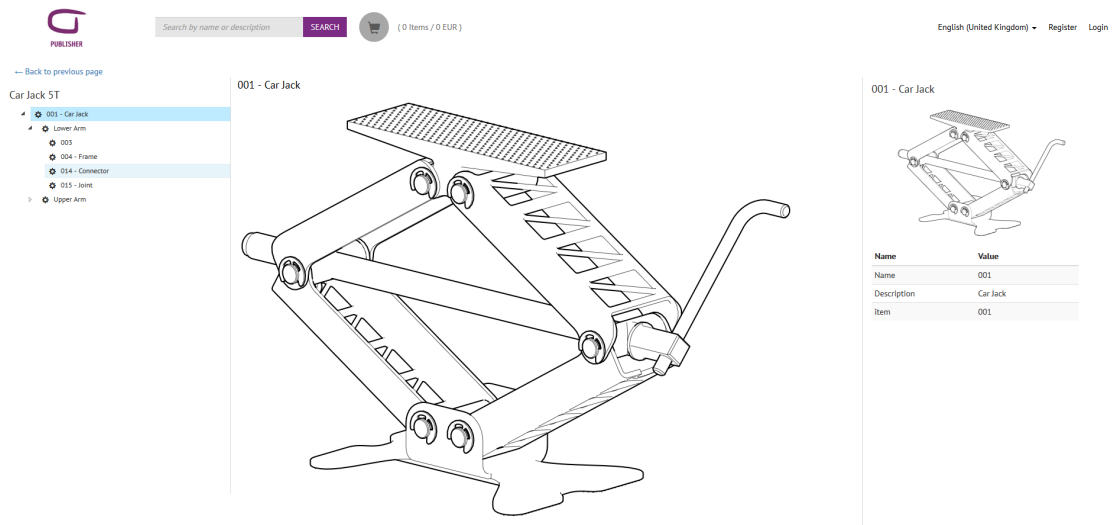
Kuva 4.1: CT Publisher editorin käyttöliittymä.

pystyisi huoltokäynnillään mobiililaitteella huoltokohteessa varmistamaan tarpeellisten varaosien varastotilanteen ja että hänellä olisi myös nopea pääsy asiakkaan laitteen dokumentaatioon.

## 4.2 Käyttöliittymä projektin alussa

CT Publisherin editorille ja selaimessa toimivalle verkkokauppakomponentille oli jo ennen tämän työn aloittamista luotu käyttöliittymät. ATR Softilla oli kuitenkin tarkoitus saada myös CT Publisher editorin toiminnallisuus tuotua myös selaimen kautta käytettäväksi. Tuotteen web sovelluksen käyttöliittymän pitäisi siis mahdollistaa tuotteiden luominen, kuvien ja muun teknisen dokumentaation lisääminen sekä tarkastelu web-alustan kautta mahdollisimman käyttäjäystävällisellä tavalla.

Itse editorin käyttöliittymän tilanne tätä diplomityötä aloitettaessa oli se, että käyttöliittymä oli pääosin syntynyt tuotetta käyttävien asiakkaiden pyytämistä



Kuva 4.2: CT Publisherin webshopin käyttöliittymä.

toiminnallisuuksista ja tarpeista syntyneestä backlogista, jota ohjelmistokehittäjät olivat purkaneet. Alunperin tuote oli erään asiakkaan käytössä dokumentinhallinnan ja julkaisun työkaluna, mutta asiakasyrityksessä huomattiin tarve yhdistää näihin toimintoihin aikaisemmin niistä erillään ollut varaosakauppa. Tämän ratkaisun hyötyinä on, että taataan sisäisten ja ulkoisten dokumenttien yhtenäisyys ja vältetään versionhallinnan tarvetta kahden järjestelmän välillä.

Editorin selainversion käyttöliittymän toteuttamisesta tehtiin päätös aloittaa puhtaalta pöydältä, eikä kierrättää editorin käyttöliittymäratkaisuja. Ajatuksena oli testata millainen käyttöliittymä tuotteelle syntyisi käyttämällä käyttäjäläheistä käyttöliittymäsuunnittelua käyttäjäkokemuksen luomiseen ja mitä hyötyjä tällaisella kehitysprosessilla saataisiin aikaan.



# 5 Case: Prosessi

Tässä kappaleessa käydään läpi, miten CT Publisherin web editorin käyttöliittymää suunniteltiin käyttäen käyttäjälähtöisen suunnittelun metodeja.

## 5.1 Suunnittelutyön aloittaminen

Työ alkoi hahmottelemalla ensin työssä käytettävä suunnitteluprosessi. Suunnitteluprosessin mallina käytettiin pääosin Goal-Directed Design prosessimallia. Ennen työn aloittamista prosessimalli esitettiin ja hyväksytettiin tuotteen omistajalla. Prosessimallille ei suunniteltu aikataulua päivälleen, sillä aikaisempaa kokemusta tämänkaltaisen suunnitteluprojektin läpiviennistä ei ollut. Alussa rajattiin kuitenkin, että suurin osa projektin ajasta tullaan käyttämään sen suunnitteluvaiheessa. Päätettiin myös, että jos aikataulusta pitää joustaa niin jousto tulisi tapahtumaan projektin loppupäästä, eli käyttöliittymän ohjelmointivaiheesta.

Prosessi muotoiltiin rakenteeltaan seuraavanlaiseksi:

- **Määrittelyvaihe**
  - Määrittele mitä ongelmaa ratkaistaan.
  - Määrittele työn ja tavoitteiden laajuus.
  - Määrittele tarvittavat resurssit (aika, henkilöt, roolit)
  - Luo prosessisuunnitelma ja työnkuvaus mitä seurata.

- Määrittele mitä metodeja ja työkaluja käytetään.
- **Esitutkintavaihe 1: Tuotteen ymmärtäminen**
  - Selvennä CT Publisherin tarkoitus ja toiminnot.
  - Määritä tuotteen arvot, millä liikealoilla sitä voitaisiin käyttää
- **Esitutkintavaihe 2: Käyttäjän ymmärtäminen**
  - Tutki olemassaolevia asiakkaita ja heidän käyttötapauksiaan.
  - Tutki kilpailijoita ja heidän asiakkaitaan.
  - Haastattele tuotteen omistajaa ja muita työntekijöitä olemassa olevista käyttäjistä sekä potentiaalisista asiakassegmenteistä
  - Tunnista kohdekäyttäjät ja kerää heistä dataa
- **Ymmärtämisvaihe 1: Persoonat**
  - Muodosta tyypillisten käyttäjien persoonat kerätystä datasta.
  - Tunnista ja priorisoi tärkeimmät käyttäjäpersoonat.
  - Luo kaaviot persoonille
- **Ymmärtämisvaihe 2: Tavoitteet ja käyttötapaukset**
  - Luo persoonille tavoiteohjautuneet käyttötapaukset ja käyttäjäpolut
  - Kirjoita skenaariot käyttötapausten ympärille
  - Listaa jokaiselle skenariolle sen sisältämät yksittäiset tehtävät
- **Ideointivaihe: Käyttöliittymäsuunnittelu ja prototyypaus**
  - Järjestetään työpaja, missä osallistujat kukin edustavat valittua käyttäjäpersoonaa ja hahmotellaan yhdessä skenaarioiden näkymiä.

- Työpajan jälkeen suunnittelu aloitetaan implementoimalla tärkeimmästä yksittäisestä käyttötapauksesta.
- Luodaan aluksi paperiprototyyppejä, niiden jälkeen siirrytään tietokoneprototyyppeihin.
- Tutki lisää yleisiä ohjeita käyttöliittymäsuunnittelussa, tutki kilpailijoiden käyttöliittymäratkaisuja.

- **Testausvaihe**

- Testaa käyttöliittymää käyttäjillä tai ilman. Käytettävyydestä testaa- jien kanssa, kognitiivinen läpikäynti jos testaa- jia ei ole käytettävissä. Tut- ki heuristisen evaluoinnin mahdollisuutta.
- Testaa käyttöliittymää aina kun uusi toiminnallisuus implementoitu.
- Useampi pieni testi yhden suuren sijaan.
- Iteroi suunnittelua ja testausta, käyttöliittymän monimutkaistuessa sitä mukaa kun käyttötapauksia implementoidaan.
- Valmiin määritelmä: SUS testi.

- **Suunnittelu: käyttöliittymän konsepti ja määrittely**

- Dokumentoi käyttöliittymä: miltä se näyttää ja miten se käyttäytyy
- Luo opas, jonka avulla käyttöliittymän implementointi onnistuu ilman, että kehittäjien tarvitsee miettiä käytettävyyden

- **Implementointi**

- Käyttöliittymän implementointi dokumentaation mukaisesti.

- **Pilotointi ja julkaisu: jatkotestaus**

- Jos implementoinnissa joudutaan tekemään kompromissejä käyttöliittymän mallin ja teknisten rajoitusten välillä, muutokset käyttöliittymässä on syytä testata.

- **Evaluointi**

- Prosessin ja tulosten arviointi.

## 5.2 Määrittelyvaihe

Ongelmaa lähdettiin hahmottamaan aluksi määrittelyvaiheessa, jossa määriteltiin tarkemmin tuotteen omistajan kanssa työn laajuutta ja rajattiin sitä, mikä on varsinainen ongelma joka tämän työn avulla pyrittiin ratkaisemaan. Tämän määrittelyn jälkeen tutkittiin ja vertailtiin erilaisia käyttöliittymäsuunnittelun prosesseja ja miten hyvin ne soveltuisivat tähän ongelmaan. Sopivimmaksi suunnitteluprosessiksi valittiin Goal Directed Design, joka soviteltiin tähän käyttötapaukseen. Sen lisäksi suunnitteluprosessiin otettiin käyttöön GUIDe-prosessimallista iteroivien silmukoiden käyttö prototyypausvaiheessa.

Sen jälkeen luotiin sovelletulle prosessille ja projektille etenemissuunnitelma, mitä oli tarkoitus käyttää mallina käyttöliittymän suunnittelussa ja toteutuksessa. Tämä etenemissuunnitelma toimi myös apuna tuotteen omistajalle ja muille projektin osakkaille, sillä etenemissuunnitelma auttoi kommunikoimaan heille prosessin eri vaiheet ja aikataulut, helpottaen näin ollen projektin seuraamista. Etenemissuunnitelma toi myös esille projektissa tarvittavia resursseja, sekä selkeytti eri henkilöiden rooleja kehityksessä. Etenemissuunnitelma luotiin yrityksen Confluence-ympäristöön, missä se oli kaikille helposti saatavilla.

### 5.3 Esitutkintavaihe 1: Tuotteen ymmärtäminen

Esitutkintavaihe aloitettiin tutkimalla itse tuotetta, sen visiota ja ominaisuuksia. Tähän vaiheeseen sisällytettiin myös alustavaa käyttöliittymäsuunnittelun ja hyvän käyttäjäkokemuksen luomisen julkaisujen ja metodien tutkimusta.

Tuotteen kohdemarkkinoita ei oltu loppuun asti määritetty projektin alussa. Tämä tieto on olennainen tieto kun suunnitellaan mahdollisimman käyttäjäystävällistä käyttöliittymää. Sitä ei oikeastaan voi tehdä tietämättä varsinaista loppukäyttäjää. Tuotteen kohdeyleisön tarkempi määrittely oli suoritettava seuraavaksi. Loppukäyttäjiä oltiin koitettu hahmotella aikaisemmin projektissa tuolloin mukana olleiden kesäharjoittelijoiden avulla, mutta aikataulujen ja työn vaativuuden takia selvitystyössä ei oltu päästy juurikaan alkua pitemmälle.

Kohdemarkkinoiden määrittely toteutettiin haastattelemalla tuotteen omistajaa. Tältä kerättiin tietoa erilaisista yrityksistä jotka jo käyttävät CT Publisheria ja minkälaiset henkilöt käyttävät tuotetta nyt. Lisää tietoa saatiin haastattelemalla tuotteen markkinointivastaavaa ja selvittämällä hänen näkemyksiään tuotteen parhaista markkina-alueista. Tämän jälkeen perehdyttiin tuotteen kilpailijoihin ja tutkittiin syvemmin kilpailijoiden vastaavia tuotteita. Listaa kilpailijoista oli luotu jo aiemmin markkinoinnin toimesta, käymällä tätä listää läpi selvitettiin kilpailijoiden tuotteiden tärkeimmät ominaisuudet. Aikaa kilpailijoiden tuotteiden asentamiseen ja niihin tarkempaan perehtymiseen ei ollut riittävästi, joten tuotteita tutkittiin pääasiallisesti analysoimalla niiden markkinointimateriaaleja ja tutoriaalivideoita.

Kilpailijoiden hinnoitteluja selvitettiin myös mahdollisuuksien mukaan, ja vertailtiin mihin CT Publisher sijoittuu vertailussa ominaisuuksiltaan ja hinnoittelultaan. Tämä auttaa ymmärtämään paremmin tuotetta ja sen käyttömahdollisuuksia, sekä toimii myös samassa yrityksessä markkinoinnin tukena. Hintatietoja ei ollut saatavilla suurimmasta osasta kilpailevista tuotteista.

Tutkimalla kilpailevien yritysten asiakastapauksia löydettiin tuotteelle kolme eri-

laista käyttötapausta:

- Tuotetta voitaisiin käyttää varaosamyynnin alustana yritykselle, joiden asiakaskunta on tee-se-itse -henkistä ja haluaa korjata tuotteensa itse. Tällaiset asiakkaat voisivat alustaa käyttämällä helposti löytää ja tilata tarvitsemansa varaosat. Alusta tarjoaisi heille myös käyttöohjeet korjausta varten
- Toinen käyttökohde olisivat yritykset, jotka tarjoavat huoltopalveluja asiakkailleen. Huoltomiehet voisivat käyttää alustaa katalogina, joka sisältää manuaaleja ja muita tietoja yrityksen tuotteista. Interaktiivinen katalogi joko räjäytyskuvilla tai 3D-malleilla selkeyttäisi huoltoeknikoiden työtä ja vähentäisi virheiden määrää heidän korjaustöissään.
- Kolmas potentiaalinen käyttökohde olisi alustan käyttö yrityksen sisäiseen dokumentaatioon. Kohdemarkkinat olisivat pääosin suunnittelutoimistoja, joita alusta helpottaisi tuotetiedon jakamisessa ja parantaisi kollaboratiivista suunnittelua.

Näistä käyttökohteista keskusteltiin tuotteen omistajan ja markkinoinnin kanssa: he näkivät potentiaalisimmaksi markkinaksi alustan käytön huoltoteknikoiden käyttöön B2B-markkinoilla.

## 5.4 Esitutkintavaihe 2: Käyttäjien ymmärtäminen

Esitutkintavaiheen toisena osuutena oli tuotteen käyttäjien ymmärtäminen. Haasteena projektin tässä vaiheessa olivat projektin suunniteluvaiheessa heikosti määritetyt kohdemarkkinat ja tuotteen loppukäyttäjät. Puutteellisten määrittelyjen vuoksi käyttöliittymän suunnitteluprosessi venyi tässä kohtaa, koska oli tarve käyttää suunniteltua enemmän aikaa loppukäyttäjien löytämiseen ja tutkimiseen, jotta heidän tavoitteitaan voitaisiin ymmärtää ja hyödyntää GDD-prosessin mukaisella ta-

valla. Toisena haasteena olivat käytettävissä olevat ajalliset- ja henkilöstöresurssit. Aikaa kului prosessin suorittamisen lisäksi sen syvempään taustoihin ja eri vaiheisiin perehtymiseen.

Tässä vaiheessa tutustuttiin tarkemmin tuotteen potentiaalsiin käyttäjiin, tutkien ja vertaillen samankaltaisia asiakastapauksia CT Publisherin ja sen kilpailijoiden välillä. Tarkoitus oli yrittää ymmärtää asiakkaita ja tuotteen käyttäjiä; miksi he valitsivat juuri tietyn tuotteen eivätkö jotain muuta vaihtoehtoista. Vertaillen samantyyppisiä asiakastapauksia löydettiin asiakkaiden pääsyyksi jonkin alustan valintaan sen käyttöönoton ja käytön helppous, toisin sanoen kuinka hyvin alusta integroituu asiakkaalla jo käytössä olevien järjestelmien kanssa ja kuinka helppo sitä on käyttää.

Näistä asiakastapauksista kartoitettiin tuotteen erilaiset käyttäjäryhmät ja heidän tavoitteensa. Tämän jälkeen oli kerättävä lisää tietoa löydetyistä käyttäjistä ja selvitettävä heidän tavoitteensa.

Tiedon keräys suunniteltiin suoritettavan käyttäjille lähetetyllä kyselyllä. Kyselyillä kerättyä dataa voitaisiin tarvittaessa täydentää haastatteluilla. Kyselyn etuna nähtiin mahdollisuus kerätä tietoa laajalti pienillä resursseilla, mutta riskinä oli ettei siihen saada tarpeeksi vastauksia. Tuotteen omistajan ehdotuksesta kyselyt suunniteltiin lähetettävän viikon alkupuolella, sillä hänen kokemustensa mukaan kyselyihin saataisiin näin enemmän vastauksia.

Haastattelujen käytön ajateltiin olevan monella tavoin parempia ja tehokkaampia käyttäjien tarpeiden selvittämiseksi. Haastattelijalla on näissä mahdollisuus kysyä jatkokysymyksiä käyttäjiltä ja myös adaptoitua tilanteen mukaan esittämään erilaisia kysymyksiä. Ongelmana haastatteluissa ovat niiden vaatimat resurssit, usein ei ole riittävästi aikaa tai henkilöstöresursseja tehdä useita haastatteluja. Toinen suuri riski haastattelujen käytössä on myös se, että haastattelijan suositellaan olevan kokenut jotta saataisiin mahdollisimman laadukasta dataa. Näistä syistä haastattelujen käyttöä pidettiin vaihtoehtoisena suunnitelmana kyselyiden avulla kerätyn

datan täydentämiseen.

Kyselyt suunniteltiin ATR Soft:illa työskentelevän suunnittelijan ohjauksessa. Oli tärkeää saada kerättyä lisää tietoa eri loppukäyttäjistä ja varsinkin siitä käyttäjäryhmästä, joka tulee olemaan tuotteen todellinen käyttäjä ja vastuussa jälkimarkkinoinnista. Näitä selvitettiin tekemällä tutkimusta eri yrityksistä, jotka tarjoavat jälkimarkkinointipalveluja. Näitä yrityksiä löydettiin muun muassa haastattelemalla ATR Softin tuotemarkkinointivastaavaa ja asiakasrajapinnassa toimivia henkilöitä, heiltä saatiin tärkeää tietoa mahdollisista kohdeyrityksistä sekä yhteistietoja potentiaalisista loppukäyttäjistä.

Kyselyt suunniteltiin ja kohdennettiin eri yrityksille, joiden toimenkuvaan kuuluu koneiden valmistaminen ja jotka tarjoavat myös varaosamyyntiä tai sekä huoltopalveluita. Yritykselle lähetetyillä sähköpostikyselyillä oli tarkoitus tavoittaa yritysten sisältä ne henkilöt, jotka edustaisivat mahdollisimman hyvin tuotteen kohdemarkkinoita ja loppukäyttäjiä. Kyselyt lähetettiin opiskelijasähköpostin kautta, sillä sen ajateltiin madaltavan kynnystä vastaamiseen. Yrityksen sähköpostin käytössä todettiin riskiksi se, että kyselyn vastaanottajat voivat helpommin luulla sitoutuvansa johonkin vastaamalla kyselyyn, minkä ajateltiin mahdollisesti vähentävän saatujen vastausten määrää.

#### 5.4.1 GDPR:n huomiointi

Kyselyitä varten piti ottaa huomioon myös EU:n yleinen tietosuoja-asetus, GDPR (General Data Protection Regulation), sillä kerätty lista potentiaalisista kyselyn vastaanottajista sisältää yksilöiviä henkilötietoja kuten nimiä, sähköpostiosoitteita ja puhelinnumeroita. Markkinoinnin toiveena oli kyselyitä suunniteltaessa, että kyselyyn sisällytettäisiin mukaan kysymys luvasta markkinoinnille ottaa jatkossa yhteyttä. Tämän lisäksi kyselyssä oli myös tarkoitus pyytää vastaajalta suostumusta testaukseen osallistumisesta. Molemmat kysymykset päätettiin jättää kuitenkin



pois. Syynä oli tavoite välttää yksilöivien henkilötietojen keräystä, koska niiden ajateltiin mahdollisesti vähentävän intoa kyselyyn vastaamiseen. Näiden kysymysten poistaminen myös lyhensi kyselyn pituutta. Tutkimusten mukaan kyselyn pituus vaikuttaa halukkuuteen vastata kyselyyn: mitä lyhyempi kysely, sitä suuremmalla todennäköisyydellä siihen vastataan [29, p 266].

### 5.4.2 Kyselyjen testaus

Ennen lähettämistä kyselyt testattiin yrityksen sisäisesti viidellä vapaaehtoisilla koe-käyttäjällä. Testien tarkoitus oli pääasiallisesti selvittää, onko itse kyselyn kieliasu helposti ymmärrettävissä ja kuinka kauan sen suorittaminen keskimäärin kestää. Testauksessa kyselyn suorittamiseen kului keskimäärin noin viisi minuuttia, jota pidettiin sopivana kestona. Palautetta tuli myös itse kysymyksistä, mutta nämä jätettiin suurimmaksi osaksi huomiotta, koska testiyleisöön ei kuulunut jälkimarkkinoinnin asiantuntijoita, joille testi oli suunnattu. Kyselyn rakennetta ja kieliasua muokattiin testeistä saadun palautteen perusteella helpommin ymmärrettävämmäksi.

### 5.4.3 Kyselyiden lähettäminen

Itse sähköpostit lähetettiin kahdessa erässä suunnitellusti käyttämällä opiskelija-sähköpostia. Ensimmäisessä erässä vastaanottavat henkilöt tiedettiin tehdyn esitutkimuksen perusteella, toisessa sähköpostit lähetettiin yritysten geneerisiin info@sähköpostiosoitteisiin. Korkeaa vastausprosenttia info@-osoitteista ei odotettu, mutta koska käytettiin samaa kyselyä, arvioitiin olevan kannattavaa lähettää kysely näihin osoitteisiin toivoen että kysely ohjattaisiin relevantille vastaajalle yrityksen sisällä.

#### 5.4.4 Kyselyjen tulokset

Kyselyjen lähettämistä sähköpostilla pidettiin suurimmaksi osaksi onnistuneena, onnistuimme saavuttamaan noin 24% vastausasteen. Tämä datamäärä riittää auttamaan käyttäjäpersoonien muodostamisessa. Vastauksista saatu data oli kuitenkin osittain puutteellista, sillä eräisiin vastauksiin kaivattiin tarkennusta sekä jatkokysymyksiä. Kyselyn formaatti ei kuitenkaan tätä sallinut. Ongelmana oli myös muodostaa empatiakarttoja käyttäjäpersoonille, sillä saadut vastaukset eivät juurikaan olleet avuksi niiden muodostamisessa. Kyselyn avulla saatiin kuitenkin validoitua aikaisemmin kerättyä dataa ja luotua dataan pohjautuvia käyttäjäpersoonia.

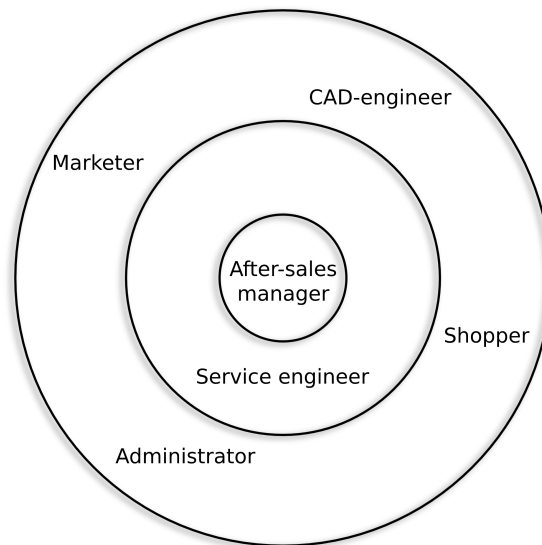
### 5.5 Käyttäjäpersoonien määrittäminen

Prosessin aikaisemmista vaiheista kerätystä datasta luotiin käyttäjäpersoonat vastaamaan tuotteen lopullisia käyttäjiä.

Tässä projektissa huomiottiin, että tuotteen käyttäjäpersoonana ei ole sama kuin ostajapersoonana. Koska tuotteen ensisijaiset kohdemarkkinat ovat B2B-yritykset, henkilö joka tekee ostopäätöksen alustasta on hyvin todennäköisesti eri henkilö kuin tuotteen todellinen loppukäyttäjä. Ostajapersoonat jätettiin siis muodostamatta, koska ne eivät ole relevantteja käyttöliittymäkehityksen kannalta eikä niitä varten näin ollen oltu myöskään kerätty dataa.

Tutkimusten ja kyselyn avulla selvitetystä käyttäjäpersoonista valittiin ne käyttäjäryhmät, joita priorisoitaisiin ja joille käyttäjäkokemusta ensisijaisesti suunniteltiin. Tärkeimmiksi persooniksi nousivat tässä esille huoltoteknikot ja jälkimarkkinointivastaavat. Muita löydettyjä tuotteen käyttäjäpersoonia olivat markkinoijat, CAD-insinöörit, järjestelmänvalvojat ja varaosien ostajat. Kyselyistä saatuja tuloksia ja kilpailijoiden tuotteiden käyttötapauksista saatuja tietoja yhdistämällä luotiin käyttäjäpersoonat näistä kahdesta. Huoltoteknikoista persoonan luominen onnis-

tui helpommin saatavilla olevan datan määrän ansiosta, jälkimarkkinointivastaavan persoonan määrittelyssä oli hankaluuksia varsinkin empatiakartan muodostamisessa. Jälkimarkkinointivastaavan persoonaa täydennettiin jälkepäin haastattelemalla ja keräämällä dataa tuotteen omistajalta, jolla oli tietoa tästä käyttäjäkunnasta.



Kuva 5.1: Kartta CT Publisherin käyttäjäpersoonista.

Persoonien luomisessa käytettiin pohjana ATR Softin käyttämän Confluencen tarjoamaa käyttäjäpersoonapohjaa ja persoonat luotiin yrityksen Confluence-ympäristöön. Tämän oli tarkoitus auttaa välittämään helpommin tietoa persoonista tuotteen omistajalle sekä muille kehittäjille. Valmiista Confluence-pohjaa muokattiin tälle projektille paremmin sopivaksi lisäämällä kenttiä sekä poistamalla vähemmän relevantteja kenttiä. Persoonien tietokentäksi lisättiin IT-taidot kenttä, sillä sen katsottiin olevan tarpeellinen ja merkittävä tieto käyttöliittymää suunniteltaessa persoonia varten. Valmiista kentistä poistettiin persoonien tulotasot sekä kuvaukset asuin-ympäristöstä, sillä näitä varten ei oltu kerätty dataa eikä niiden ajateltu olevan kovin merkityksellisiä tässä tapauksessa.

Molemmille persoonille luotiin myös luvussa 2.5 kuvatut empatiakartat, joiden

lohkot täytettiin persoonista kerätyllä datalla. Sen lisäksi persoonille kirjoitettiin kuvaukset heidän motivaatiostaan tuotteen käytössä sekä kipupisteistä työtehtävissään. Näiden tarkoituksena oli tehdä persoonista enemmän realistisia ja helpommin samaistuttavia.

## 5.6 Persoonien tavoitteet ja käyttötapaaukset

Käyttäjäpersoonille määritettiin mitkä ovat heidän tavoitteensa CT Publisheria käytettäessä ja millaisia työtehtäviä he tuotteella suorittaisivat. Näiden päämäärien ympärille kirjoitettiin skenaariot, jotta testaaajat voisivat paremmin samaistua käyttäjään. Esimerkiksi jälkimarkkinointivastaavan käyttötapaauksia olivat muun muassa tuotteiden lisääminen järjestelmään ja tuotetietojen editointi. Huoltoteknikon käyttöskenaarioihin kuuluivat asiakaskohtaisten tuotetietojen etsiminen järjestelmästä, sekä myös näiden tietojen lisäys.

Kirjoitetut skenaariot käytiin läpi tuotteen omistajan kanssa ja ne todettiin todenmukaisiksi käyttäjäskenaarioiksi, jotka vastaavat jo olemassa olevien asiakkaiden käyttötapaauksia. Suunnittelua varten käyttöpersoonat ja skenaariot piti myös priorisoida, jotta voitaisiin aloittaa suunnittelu tärkeämmälle käyttäjälle ja näin tuottaa heille paras käyttökokemus. Tuotteen omistajan kanssa päätettiin priorisoida jälkimarkkinointivastaavan käyttäjäpersoona, sillä se nähtiin tuotteen tärkeimmäksi käyttäjäksi.

**Skenaario 1:**

Janne's company has decided to improve their after sales service by sharing product data on a single platform. Janne needs to add all the product data to the new platform info about 30 different machines their company manufactures:

- 6 excavators
- 4 stationary crushers
- 4 mobile crushers
- 10 drills
- 6 rock dusters

The company ERP system has the needed documents, such as datasheets, drawings,pictures etc. but the company doesn't use Composer, so Janne has to input product data manually.

Kuva 5.2: Esimerkki kirjoitetuista käyttöskenaarioista.

Jälkimarkkinointivastaavan käyttötapauksissa oletettiin, että tuotteen on jo hankittu ja implementoitu järjestelmänvalvojan tai muun vastaavan puolesta. Oletuksena oli myös, että järjestelmänvalvoja on luonut tunnukset jälkimarkkinointivastaavaa varten, eikä käyttöskenaariota käyttäjätilin luomisesta tarvinnut luoda. Nämä molemmat oletukset järjestelmänvalvojan roolista tuotteen käyttöönotossa pohjautuivat asiakashaastatteluista ja tuotteen omistajaa haastatteleamalla saatuun dataan.

## 5.7 UI Suunnittelu ja prototyypaus

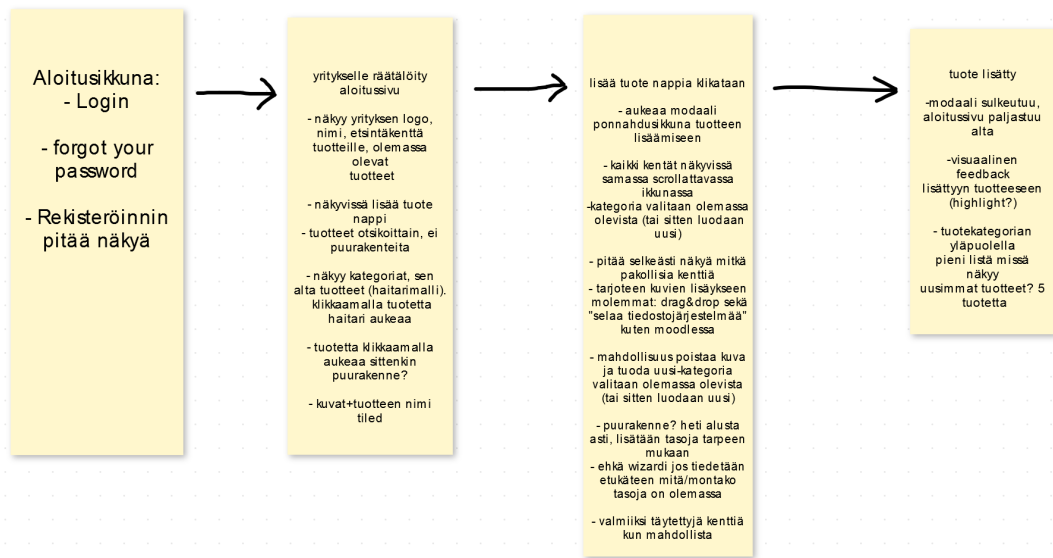
Persoonien ja niiden skenaarioiden muodostamisen jälkeen aloitettiin työpajojen suunnittelu. Niissä oli tarkoituksena, että osallistujat edustavat käyttäjäpersoonia ja simuloivat eri käyttötapauksia prosessimallin mukaisesti. Työpajojen tarkoitus käyttöliittymäsuunnittelun kannalta oli tunnistaa, mitä olennaisia vaiheita skenaarioiden suorittamiseen kuuluu käyttäjän kannalta ja luoda näistä käyttäjäpolku skenaariolle.

Työpajoja varten tarvittiin osallistujia, joihin oli tarjolla kaksi vaihtoehtoa: etsi-

mällä yrityksen sisältä halukkaita osallistumaan suunnitteluun tai värväämällä suunnittelusta kiinnostuneita Turun Yliopiston opiskelijoita. CT Publisher -projektin parissa jo työskenteleviä ei haluttu käyttää työpajoissa, sillä haluttiin mahdollisimman tuore visio tuotteen käyttöliittymän suunnitteluun ja ajateltiin, että on parempi käyttää kokonaan projektin ulkopuolisia tähän tarkoitukseen.

Alunperin työpajaan oli tarkoitus käyttää Turun yliopiston ryhmätyötiloja ja järjestää työpaja kasvotusten parhaan mahdollisen kommunikoinnin mahdollistamiseksi osallistujien välillä. Vallinneen COVID-19 epidemian takia julistetun poikkeus-tilan vuoksi yliopiston tilojen käyttö ei kuitenkaan ollut ajankohtana mahdollista, joten työpaja jouduttiin järjestämään etänä Skype-puhelun avulla. Työpajaan osallistui kolme henkilöä. Ryhmä haluttiin pitää pienenä, jotta sitä olisi helpompi ohjata etäyhteyden ylitse ja jotta kaikki saisivat varmasti esitettyä ideansa.

Ennen työpajan aloitusta osallistujille oli lähetetty kyseisessä työpajassa simuloitavan persoonan kuvaus, jotta he voisivat etukäteen tutustua siihen ja näin ollen helpommin asettautua persoonan rooliin. Työpajan alussa käytiin yhdessä läpi ensin työpajan tarkoitus ja sen tavoite, sekä esiteltiin käyttäjäskenaario persoonaa varten. Sen jälkeen pajaan osallistujien kanssa aloitettiin hahmottelemaan käyttäjäpolkua käyttämällä selaimella toimivaa muistilappu-piirtotaulu sovellusta, johon jokainen lisäsi omia ajatuksiaan ranskalaisin viivoin.



Kuva 5.3: Kuvakaappaus ensimmäisestä työpajasta.

Ensimmäisen työpajan jälkeen haluttiin kuitenkin kerätä lisää erilaisia ideoita käyttöliittymän toteuttamiseksi, joten päätettiin järjestää toinen työpaja uusien osallistujien kanssa. Tällä kertaa osallistujille lähetettiin tarkempi kuvaus itse työpajan sisällöstä ennen tapaamista, jotta he näin pystyisivät jo etukäteen pohtimaan yksin erilaisia käyttöliittymäratkaisuja ja työpajassa voitaisiin keskittyä näiden ratkaisujen läpikäyntiin sekä ideoimaan näyttösisältöjä niiden pohjalta. Tämän lisäksi skenaarioita muokattiin yksityiskohtaisemmaksi, jottei niitä tarvitsisi avata osallistujille enempää suullisesti työpajan aikana. Myös tämä työpaja jouduttiin järjestämään etänä.

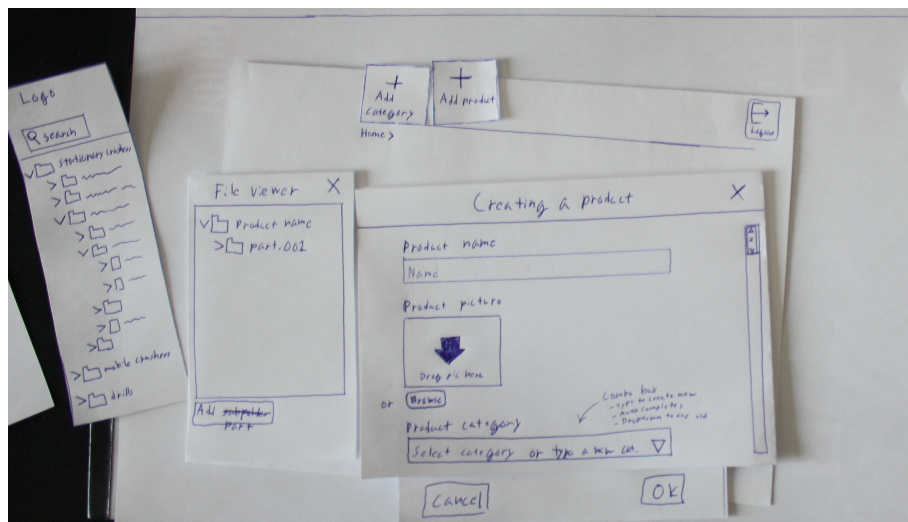
Jälkimmäisessä työpajassa huomattiin, että osallistujat olivat pohtimalla käyttöliittymää etukäteen muodostaneet itselleen vahvan mielipiteen siitä, millaiselta käyttöliittymän kuuluisi näyttää ja toimia. Osallistujilta saatiin näin enemmän omia mielipiteitä, mutta heidän huomattiin puolustavan omia mielipiteitään hyvinkin vahvasti ja kompromissien löytäminen eri osallistujien käyttöliittymäratkaisujen välillä oli haastavaa. Näin saatiin kuitenkin enemmän dialogia osallistujien välillä, mikä johti

puolestaan siihen että työpajan aikana muodostui enemmän erilaisia ideoita käyttöliittymän toteuttamiseksi. Työpajan vetäjän rooli toisessa työpajassa oli enemmän ohjaava verrattuna ensimmäiseen työpajaan.

### 5.7.1 Käyttöliittymän paperiprototyypit.

Työpajoista saatujen ideoiden ja käyttäjäpolkujen pohjalta lähdettiin muodostamaan käyttöliittymän runkoa paperille aloittaen ensimmäisestä ja tärkeimmäksi arvioidusta käyttötapauksesta, eli siitä kun jälkimarkkinointivastaava haluaa lisätä tuotteen ja sen varaosat järjestelmään. Suurimmaksi ongelmaksi tuotteen lisäysprosessissa oltiin jo työpajoissa huomattu tuotteen osien hierarkian luominen. Tähän ei löydetty yksimielisesti parasta ratkaisua ja ajatuksena oli tehdä prototyyppejä useista mahdollisista eri ratkaisuista ja testauksen avulla löytää niistä toimivin.

Aikaa paperiprototyypin luomiseen kului keskimäärin yhdestä kahteen tuntia per luotu prototyyppi. Sitä edelsi useamman tunnin tutkimustyö ja perehtyminen erilaisiin käyttöliittymäratkaisuihin ja mahdollisuuksiin.

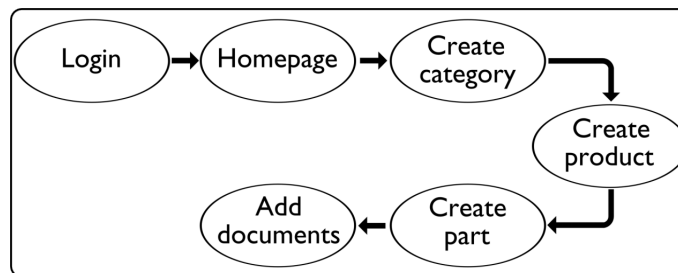


Kuva 5.4: Kuva paperiprototyypistä.

Yhtenä ratkaisuna suunniteltiin paperilla käyttäjälle vaiheittaista ohjattua luo-



mista, (*engl. wizard*), jossa käyttäjän tuotteen lisäämisprosessia ohjaisi monivaiheinen wizard-ikkuna. Jo prototyypausvaiheen alkupuolella kuitenkin huomattiin tämän mallin soveltuvan huonosti tähän käyttötapaukseen. Ongelmaksi huomattiin hierarkian muodostaminen ohjatulla luomisella. Käyttöliittymän wizard-prototyyppi oltiin jaettu kuuteen vaiheeseen: sisäänkirjautuminen, päänäkyvä, kategorian luominen, tuotteen luominen, varaosan luominen ja dokumenttien lisääminen.



Kuva 5.5: Wizardin vaiheiden hahmotelma

Wizardin ongelmaksi kuitenkin huomattiin jo paperiprototyyppien suunnittelu- vaiheessa, ettei se auta juurikaan käyttäjää tuotehierarkian luomisessa. Wizard olisi tehokas työkalu yhden tuotehierarkian haaran luomisessa, mutta jos sillä koitettaisiin luoda tuotetta, jolla oli useampia varaosia ja niihin liittyviä dokumentteja, ongelmaksi muodostuisi toisten haarojen luominen yhden wizardin avulla.

Toisella paperiprototyyppillä haluttiin verrata, miten hyvin käyttöliittymä toimisi jos siitä jätettäisiin pois kategorioiden, tuotteiden ja varaosien välinen erittely, ja kaikki olisivatkin kansioita. Tämän idean pohjana oli aikaisemmin käyttäjäpersoonia varten kerätystä datasta saatu huomio siitä, että tuotteen tyyppinen loppukäyttäjä on tottunut käyttämään erilaisia tiedostojen jakamiseen tarkoitettuja pilvipalveluita, eikä näin ollen vierasta tiedostojärjestelmien käyttöä.

## 5.8 Paperiprototyyppien testaus

Käyttöliittymän ensimmäisten paperisten prototyyppien valmistuttua oli alunperin tarkoitus antaa se testihenkilöiden käytettäväksi. Epidemian aiheuttamasta poikkeustilasta johtuen näitä ensimmäisiä testejä jouduttiin muokkaamaan. Käyttöliittymän ensimmäistä paperiprototyyppiä testattiin demonstroimalla sen käyttöä testihenkilölle videopuhelun ylitse. Tässä suunnittelija itse esitti kuinka prototyypillä luodaan uusi tuote ja videota seuraava testihenkilö esitti kommentteja, kysymyksiä sekä parannusehdotuksia. Paperiprototyyppi myös skannattiin ja siitä koostettiin storyboard-muotoinen vaiheittainen pdf-dokumentti, joka lähetettiin tuotteen omistajan kommentoitavaksi. Tavoitteena oli näin kerätä palautetta käyttöliittymästä myös häneltä, sekä pitää hänet selvillä käyttöliittymäsuunnitteluprosessin kulusta.

### 5.8.1 Paperiprototyypeistä saatu palaute

Paperiprototyyppejä testattiin viiden testikäyttäjän avulla. Testeistä saatu palaute jätti paljon toivomisen varaa. Testikäyttäjien oli hankala päästä järjestelmän käyttöön sisälle katsomalla pelkkiä kuvia, vaikka valtaosa testaaajista oli ollut mukana työpajoissa suunnittelemassa käyttöliittymiä. Suurin osa testaaajilta saadusta palautteesta keskittyi käyttöliittymän elementtien sijoitteluun, eikä niinkään testausskenaarion kannalta relevantin tiedon näkyvyyteen. Palaute oli kuitenkin suurimmaksi osaksi positiivista, eikä suuria käytettävyysongelmia havaittu näillä testeillä, joten prosessin mukaan käyttöliittymäsuunnittelussa edettiin tietokoneella tehtäviin prototyyppeihin.

## 5.9 Siirtyminen tietokoneella tehtyihin prototyypppeihin

Käyttöliittymien prototyypit tehtiin tietokoneella käyttäen Adobe Xd -ohjelmaa, jolla saatiin luotua käyttöliittymän visuaalinen ulkoasu, sekä simuloitua osaa sen toiminnallisuuksista. Vaikka tähän tarkoitukseen on tarjolla muitakin ohjelmia, Adoben tuote valittiin koska se oli entuudestaan tuttu, eikä näin ollen ohjelman käytön opetteluun kuluisi ylimääräistä aikaa. Xd:llä oli myös helppoa jakaa prototyyppejä verkon yli. Ohjelmassa oli kuitenkin joitain puutteita, mitkä ilmenivät vasta myöhemmissä vaiheissa siirryttäessä korkeatasoisempiin prototyypppeihin. Prototyyppien kanssa ei esimerkiksi voinut käyttää hiiren oikeaa painiketta, tuplaklikkaus ei ollut mahdollista eikä elementtejä voinut raahata paikasta toiseen.

Kehittäminen päätettiin aloittaa yleisimmästä käyttötapauksesta, eli uuden tuotteen lisäämisestä järjestelmään. Käyttöliittymästä kehitettiin alussa kaksi hyvin erilaista prototyyppiä samanaikaisesti, molemmat pohjautuen aikaisempiin paperiprototyypppeihin.

### 5.9.1 Testaajat

Testaajat valittiin niin, että he vastaisivat mahdollisimman paljon primaarista käyttäjäpersoonaa ominaisuuksiltaan. Oikeita käyttäjiä, eli jälkimarkkinointivastaavina toimivia henkilöitä ei testeihin saatu rekrytoitua heidän kiireidensä ja poikkeustilan aiheuttamien ongelmien vuoksi.

Testaajaksi löytyi lopulta sopiva määrä yliopisto-opiskelijoita, jotka vastasivat primaaria jälkimarkkinointivastaavaa mahdollisimman hyvin ja joiden aikataulu jousti tarpeeksi paljon jotta he pystyivät osallistumaan testeihin. Testaajat eivät saaneet testeihin osallistumisesta kompensatiota. Testausryhmä pysyi pääosin samana koko suunnitteluprosessin ajan, suurin syy tähän oli se, että jälkimarkkinointivastaa-

van persoonaa mahdollisimman hyvin edustavia henkilöitä ei ollut saatavilla. Myös skenaarioihin perehdyttäminen sekä järjestelmän tarkoituksen selittäminen uusille testaaajille vei paljon aikaa, ja tätä haluttiin välttää mahdollisimman paljon ajan säästämiseksi.

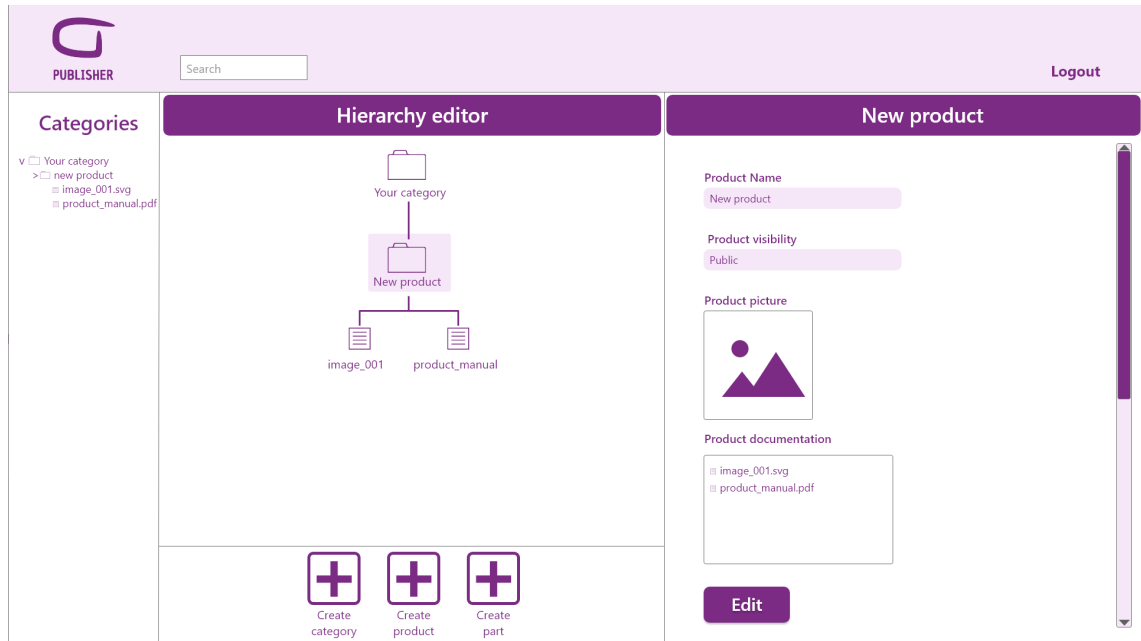
Testaajien lisäksi palautetta pyydettiin jatkuvasti myös tuotteen omistajalta sekä ATR Soft:illa työskentelevältä kokeneemmalta graafiselta suunnittelijalta. Heidän kanssa ei suoritettu käytettävyydestä, vaan heille toimitettiin aina uusimmat prototyypit, joista he saivat antaa vapaasti palautetta.

### 5.9.2 Wizardin muokkaaminen

Koska käyttöliittymäsuunnittelussa haluttiin ensisijaisesti panostaa uusiin käyttäjiin, paperiversioiden testauksen perusteella voitiin todeta wizardin käytön selkeyttävän ja helpottavan tuotteen lisäämistä katalogiin. Alkuperäisistä paperiprototyypeistä poiketen wizardia päätettiin yksinkertaista niin, että kokonaista tuotepuuta ei wizardilla luotaisi vaan wizardilla lisättäisiin yksittäisiä tuotteita ja jokaisella tuotetyypillä olisi omanlaisensa wizard.

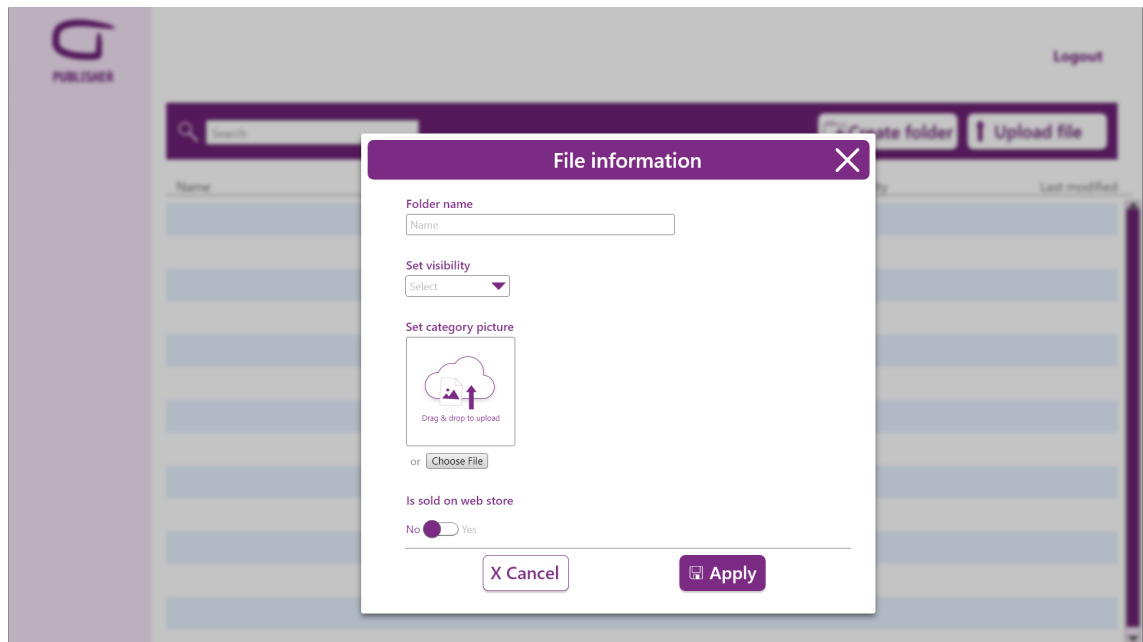
Vaikka wizardin käyttö selkeyttääkin luomisprosessia uudelle käyttäjälle, järjestelmän toistuva käyttö vaatii tehokkaampaa tapaa syöttää dataa. Wizardin yleisinä ongelmina voidaankin pitää niiden hitautta: käyttäjät joutuvat toistuvasti vaihtamaan hiiren ja näppäimistön välillä sekä siirtymään usein näkymästä toiseen, mikä lisää kognitiivista taakkaa [30] ja tehokäyttäjille pitää tarjota vaihtoehtoinen tapa syöttää tuotetietoja wizardin lisäksi. Näiden haasteiden pohdintaa päätettiin lykätä, koska alustava tarkoitus oli kuitenkin keskittyä uusiin käyttäjiin, ei niinkään tehokäyttäjiin.

### 5.9.3 Ensimmäiset tietokoneprototyypit



Kuva 5.6: Käyttöliittymän ensimmäinen tietokoneprototyyppi.

Wizardin sijaan ensimmäisessä versiossa tuotteen lisääminen tapahtui erillisessä ikkunassa, joka aukesi painamalla jotain *Create*-painikkeista. Ikkuna oli vieritettävissä jos täytettävät syöttökentät eivät mahtuneet yhteen näkymään. Jos tuotetta ei oltu luomassa, oli ikkuna joko tyhjä tai näytti *Hierarchy editor*-ikkunassa valittuna olleen tuotteen tietoja.



Kuva 5.7: Modaali wizard-ikkuna vaihtoehdoisesta käyttöliittymästä.

Toinen versio poikkesi ensimmäisestä usealla tavalla. Kahden ikkunan sijaan tässä versiossa käyttäjälle näytettiin vain yksi pääikkuna, eikä sivupalkissakaan ollut navigointia toisin kuin toisessa versiossa. Tarkoitus oli pitää näkymä mahdollisimman yksinkertaisena ja selkeänä. Myös wizard-ikkuna tuotteiden lisäämiseen oli omassa modaalisisessä ponnahdusikkunassaan. Pääikkunassa tuotteet näytettiin listanäkymässä ja listattuja tuotteita klikkaamalla käyttäjät pääsivät syvemmälle tuotehierarkian puurakenteessa.

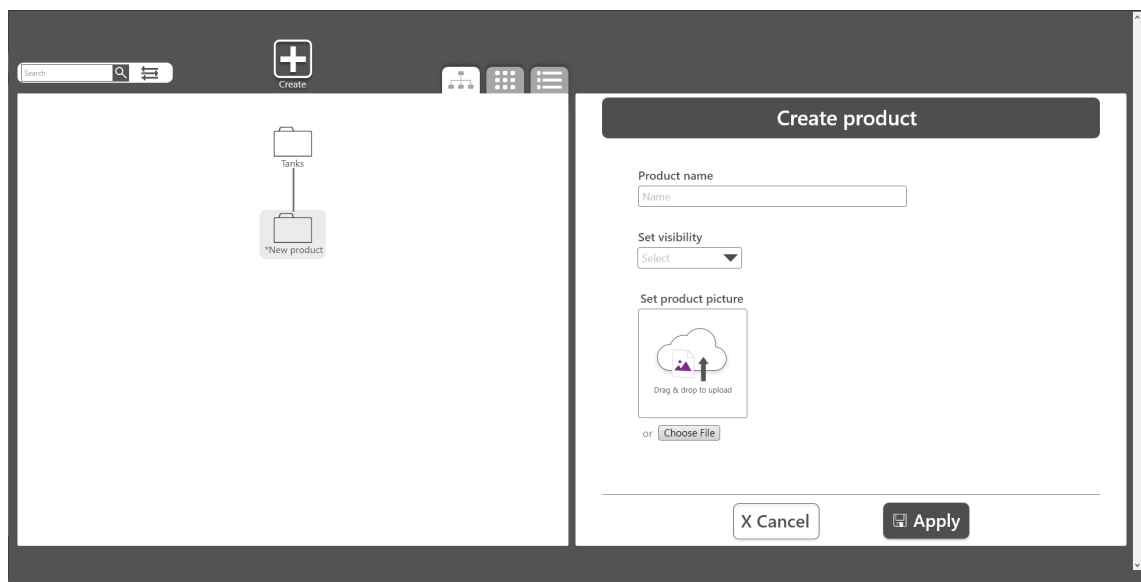
Xd-prototyypin ensimmäiset käytettävyytestit suoritettiin online-puheluiden avulla. Testaajille oli lähetetty linkit molempiin testattaviin prototyyppihin, joita he testasivat ja samalla kommentoivat niiden käyttöä. Testien jälkeen kerättiin vielä lisää palautetta ja kysyttiin tarkentavia kysymyksiä. Testaajia pyydettiin myös kommentoimaan vapaasti käyttöliittymiä ja kertomaan ajatuksiaan. Tietokoneprototyypin testeistä saatiin paperiprototyyppihin verrattuna tarkempaa palautetta järjestelmän käytettävyysoongelmista. Testaajat pitivät ensimmäisessä versiossa

suurimpana ongelmana ikkunoiden suuren määrän aiheuttamaa sekavuutta. Toisessa versiossa taas testaajat eivät olleet selvillä, millä tuotehierarkian tasolla he milloinkin olivat. Testaajat kuitenkin suosivat modaalisen ponnahdusikkunan käyttöä tuotetta lisätessä, sillä se auttoi heitä ymmärtämään ohjelman tilan paremmin.

#### 5.9.4 Seuraavat versiot ja niiden testaus

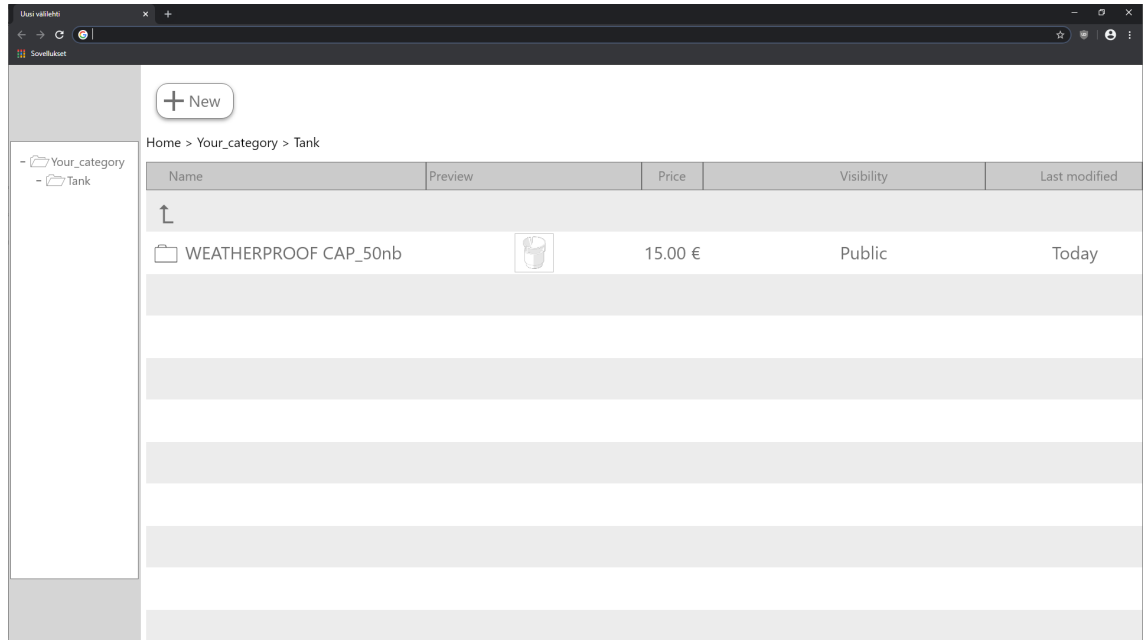
Käyttöliittymäsuunnittelua jatkettiin tekemällä molemmista prototyypeistä vielä uudet versiot, joihin löydetyt käytettävyyden ongelmat korjattiin. Ensimmäisessä versiossa testaajien mielestä sekavaa moniosaista käyttöliittymää yksinkertaistettiin poistamalla sivupalkki. Sen aikaisemmin tarjoamat navigointimahdollisuudet siirrettiin kansionäkymän yläpuolelle. Käyttäjille tarjottiin kolme eri vaihtoehtoista näkymää hierarkian näyttämiseen.

Editointipainike poistettiin kummastakin, sillä sen ei katsottu kuuluvan tähän käyttötapaukseen, koska tarkoitus oli keskittyä ensimmäiseksi uusien tuotteiden lisäämisen käyttötapaukseen.



Kuva 5.8: Ensimmäisen version paranneltu prototyyppi.

Toiseen versioon taas lisättiin sivupalkki sekä murupolku, jotka indikoivat käyttäjille tuotehierarkiasta ja millä tasolla hierarkiaa he ovat.



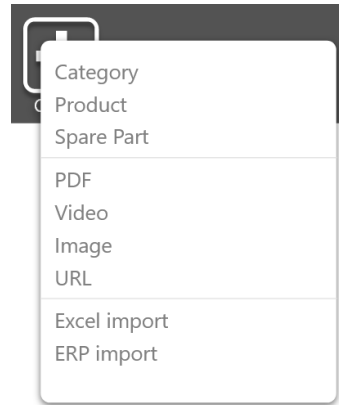
Kuva 5.9: Toinen versio parannuksineen.

Värimaailmaa oli myös yksinkertaistettu molemmissa versioissa, sillä ensimmäisen testin jälkeen todettiin myös, että ensimmäisissä prototyypeissä oltiin suunnittelussa keskitytty tarpeettoman paljon käyttöliittymän graafiseen ulkonäköön. Tämän ajateltiin vaikuttavan testaaajien keskittymiseen, vieden liikaa heidän huomioon käyttöliittymän visuaaliseen ulkonäköön kun heidän pitäisi suunnittelun tässä vaiheessa keskittyä sen sijaan enemmänkin käyttöliittymän toimintoihin ja loogisuuteen. Näin ollen käyttöliittymiä muokattaessa testeistä saatujen palautteen mukaan, muokattiin myös samalla niiden ulkoasua yksinkertaisemmaksi, jotta testaaajat voisivat paremmin keskittyä käyttöliittymän olennaisiin toimintoihin seuraavissa käytettävyystesteissä.

Sen lisäksi myös erilliset painikkeet tuotteiden lisäämiseen poistettiin molemmista ja niiden tilalle tehtiin yksittäinen *Create*-painike, jota klikkaamalla käyttä-



jät saavat esiin pudotusvalikon, missä listataan tuotteet mitkä käyttäjä voi luoda. Klikkaamalla jotain listan tuotetta viereiseen ikkunaan aukeaa sen tuotteen wizard, jossa käyttäjä voi luoda kyseisen tuotteen järjestelmään. Sillon, kun käyttäjä ei ole lisäämässä tuotetta, ikkuna näyttäisi silloin valitun tuotteen tiedot.



Kuva 5.10: Pudotusvalikko.

Ensimmäisestä testikierroksen perusteella tehtyjen muokkausten jälkeen molempia prototyyppisiä testattiin toistamiseen kolmen hengen testiryhmällä. Ensimmäisestä testistä poiketen tällä kertaa testajat jakoivat ruutunsa videopuhelun ylitse samalla kun he testasivat prototyyppisiä. Tämä antoi mahdollisuuden tarkkailla testajien kursorin liikkeitä sekä kuuntelemalla heidän palautettaan testisessioden aikana saatiin parempi kuva käytettävyyden ongelmista verrattuna aikaisempaan käytettävyydestiin. Videopuhelun jälkeen keskustelua testajien kanssa jatkettiin ja heiltä saatiin enemmän mielipiteitä ja ideoita, kuin mitä he itse testin aikana kommentoivat.

Toisen version testeissä huomattiin, että painikkeet kansionäkymän navigoinnin hierarkian näytön vaihtamiseksi jäivät testajilta helposti joko kokonaan huomaamatta tai he ymmärsivät painikkeiden tarkoituksen väärin. Testaajilla oli myös hankaluuksia löytää, mistä tuotteiden alle pystyi lisäämään uusia alatuotteita. Osa testajista yritti luoda alatuotetta editoimalla jo luotua tuotetta, eikä käyttämällä

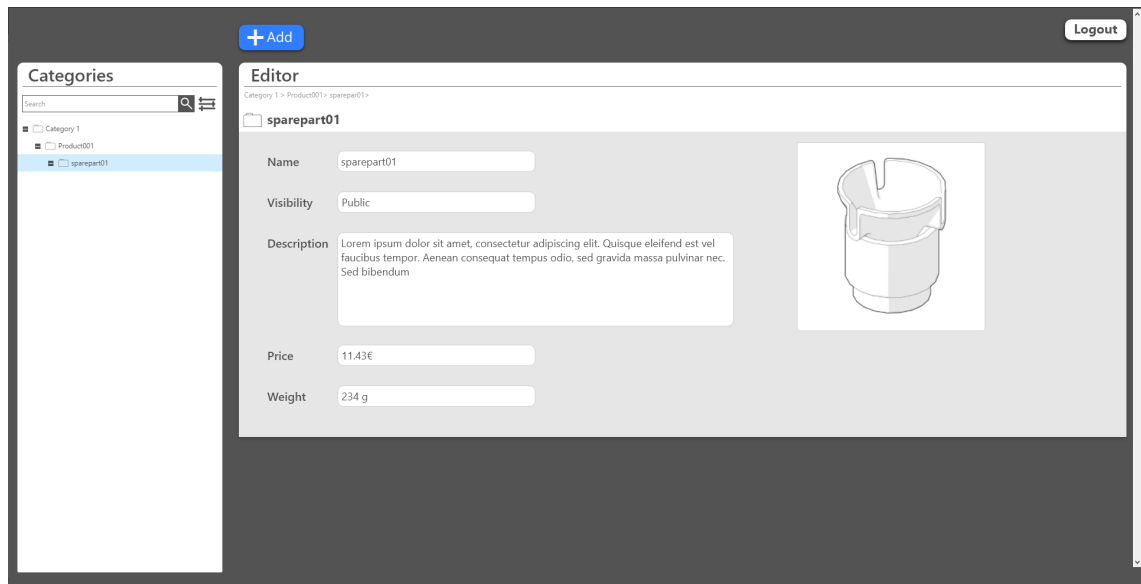
*Add*-painiketta tuotteiden lisäämiseen.

### 5.9.5 Käyttöliittymän viides versio

Tässä vaiheessa tehtiin päätös jatkaa vain yhdellä käyttöliittymän versiolla. Edellisistä testauksista saadun palautteen perusteella voitiin todeta, että käyttöliittymä toimii paremmin kun siinä on selkeästi vain yksi pääikkuna, missä tuotteiden tiedot näytetään. Tuotehierarkia näytettäisiin pienemmässä, selkeästi erillään olevassa *Categories*-ikkunassa, josta käyttäjät pystyivät hiirellä valitsemaan minkä tuotteen tiedot he halusivat nähdä. Tuotteet näytettiin ikkunassa laajennettava luettelo (engl. *Expandable list*) tyyliisesti, jossa käyttäjät pystyivät tuotteita klikkaamalla näyttää niiden alla olevat muut tuotteet.

Niin kuin toisessa aikaisemmista versioista, myös tässä tuotteiden lisääminen tapahtui *Add*-painikkeen takaa löytyvän pudotusvalikon kautta avattavalla modaalilla wizard-ponnahdusikkunalla, sillä testaajat olivat pitäneet sitä selkeimpänä tapana tapana uuden tuotteen lisäämiseen. Tuotteen lisääminen jonkin toisen tuotteen alle onnistui valitsemalla ensin tuote, minkä alle lisäys haluttiin tehdä ja sen jälkeen painamalla *Add*-painiketta.

Murupolku oltiin aikaisemmissa testeissä todettu hyväksi tavaksi indikoida käyttäjän sijaintia tuotehierarkiassa, joten se päätettiin sisällyttää myös tähän versioon.



Kuva 5.11: Käyttöliittymän viides prototyyppi.

Modaalista wizard-ikkunasta tehtiin tässä versiossa monivaiheinen, jossa käyttäjä sai etensä aina yhden syöttökentän jokaisessa wizardin vaiheessa. Tätä oltiin testattu aikaisemmin paperiversioiden kanssa, mutta silloin sen toimivuudesta ei ollut varmoja paperiversioiden testausvaikeuksien vuoksi. Wizardissa tarjottiin mahdollisuus liikkua eteen ja taaksepäin *Next*- ja *Back*-painikkeilla, joilla oli tarkoitus antaa käyttäjille parempi kontrolli wizardia käytettäessä.

The image shows a wizard titled "Create image" with a close button (X) in the top right corner. Below the title is a progress indicator with three steps: "Upload File" (green circle), "Input Information" (blue circle with 'i'), and "Summary" (grey circle). The "Input Information" step is highlighted with a blue line. Below the progress indicator is a form with the following fields:

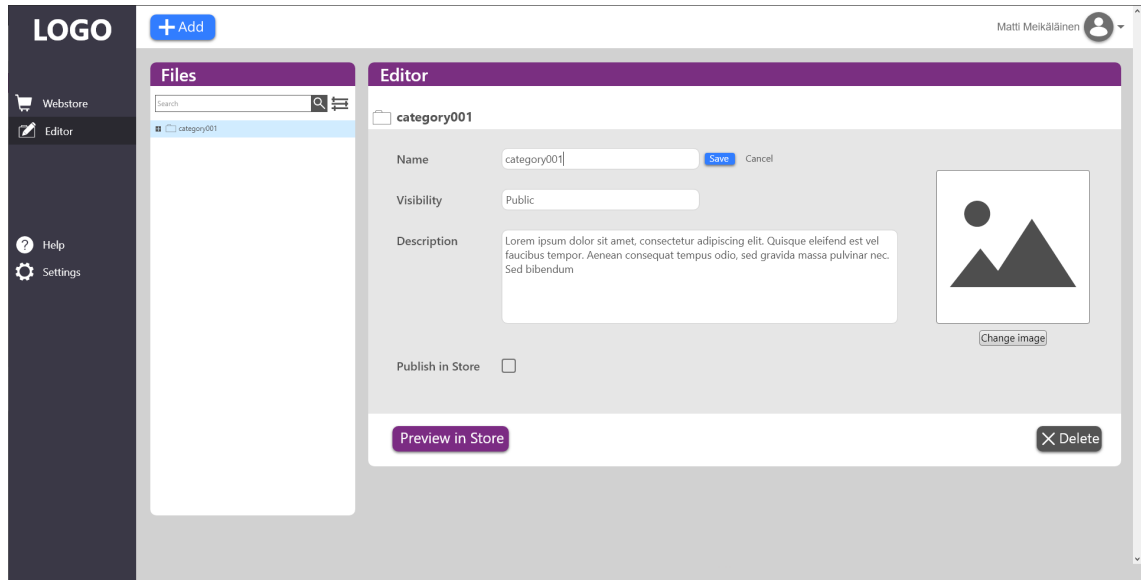
- Name**: A text input field with the placeholder text "Name".
- Set visibility**: A dropdown menu with the placeholder text "Select".
- Description**: A text area with the placeholder text "...".

At the bottom of the form are three buttons: "X Cancel", "Previous", and "Next".

Kuva 5.12: Monivaiheinen wizard.

Käyttöliittymän viidennen version testauksissa lähes kaikki testaajat pitivät monivaiheista wizardia turhan hitaana ja työläänä tuotteiden lisäykseen. Tämä korostui varsinkin kun testaajat joutuivat luomaan useampia tuotteita järjestelmään tuotehierarkian rakentamiseksi.

### 5.9.6 Käyttöliittymän kuudes versio



Kuva 5.13: Kuudes versio käyttöliittymästä.

Kuudennessa versiossa tuotiin takaisin aikaisemmin poistettu sivupalkki. Kansiohierarkian sijaan sivupalkissa mahdollistettiin verkkokaupan ja editorin välillä liikkuminen, sekä tarjottiin käyttäjille *Help*- ja *Settings*-painikkeet. *Files*-ikkunan sisällön näyttämiseen kokeiltiin myös muita tapoja kuin aikaisemmin käytetty laajennettava lista: ruudukkonäkymä (engl. *Grid view*) sekä luettelonäkymä (engl. *List view*). Näiden näkymien ongelmana oli kuitenkin se, että ne eivät toimineet hyvin verrattuna aikasempaan näyttötapaan tuotehierarkian kasvettua laajemmaksi. Laajennettavan listan tarjoama mahdollisuus piilottaa ja näyttää tuotehierarkian haaroja toimi paremmin näissä tapauksissa, joten sen käyttöä päätettiin jatkaa.

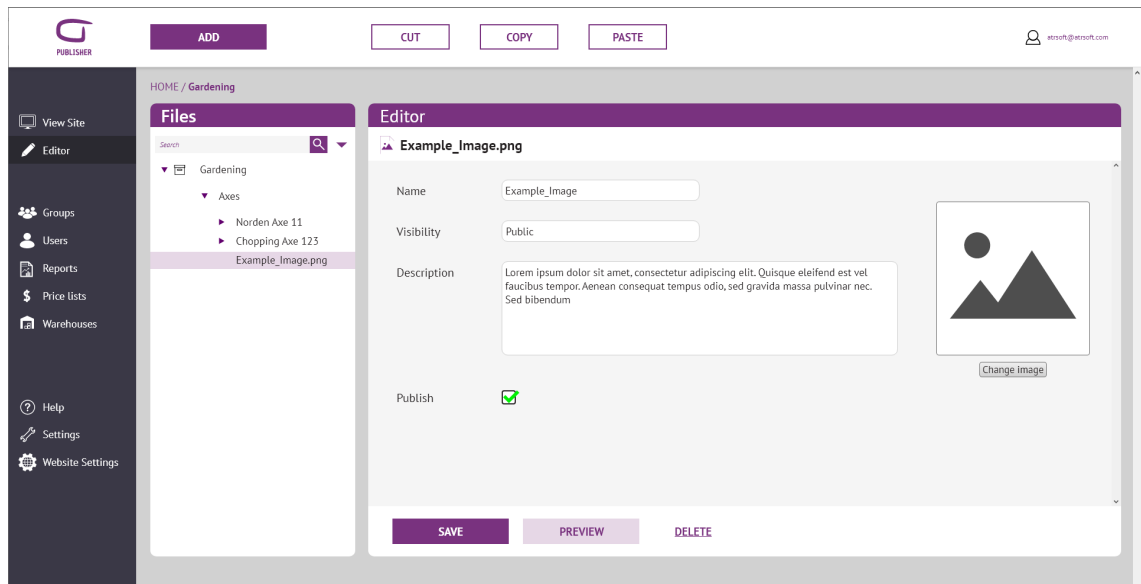
Aikaisemman version testauksesta saadun palautteen mukaan käyttöliittymä mahdollisti tässä kohtaa riittävän hyvin uusien tuotteiden lisäämisen järjestelmään, joten voitiin siirtyä seuraavaan käyttötapaukseen eli tuotetietojen editointiin järjestelmässä. Eksplisiittisen editointitilan sijaan käyttöliittymälle pohdittiin mahdollisuut-

ta sallia käyttäjän muokata tuotetietoja *edit in place* -tavalla. Käyttäjän klikattua jotain tuotteen ominaisuuden syötekenttää hiirellä ilmestyisi kyseisen kentän viereen painikkeet editoinnin tallentamiseksi ja peruuttamiseksi.

Kuudennessa versiossa mahdollistettiin myös siirtyminen verkkokaupan puolelle käyttämällä *Preview in Store*-painiketta ja näin antaa käyttäjien tarkastaa, miltä tuote näyttää siellä ennen julkaisua.

### 5.9.7 Käyttöliittymän seitsemäs versio

Käyttöliittymän seitsemäs versio ei poikennut paljoa edeltäjästään. Painikkeita ja värimaailmaa muutettiin vastaamaan tuotteen graafista ohjeistusta. Yläpalkkia muutettiin vastaamaan enemmän tuotteen graafista linjausta ja lisäämällä CT Publisher -tuotteen logo. Yläpalkkiin lisättiin myös leikkaa, kopioi ja liitä -painikkeet, joiden tarkoitus oli parantaa käyttäjien editoinnin käyttökokemusta, mutta näille ei tehty vielä mitään oikeaa toiminnallisuutta prototyyppiin. Sivupalkkiin lisättiin myös enemmän linkkejä toimintoihin, jotka olivat jo käytössä CT Publisher editorissa ja jotka haluttiin saada implementoitua tulevaisuudessa myös selainversioon. Näiden toimintojen suunnittelu ja toteutus ei kuitenkaan tulisi sisältymään tähän projektiin, mutta ne oli tarpeen ottaa huomioon käyttöliittymässä tulevaisuutta varten.



Kuva 5.14: Seitsemäs versio käyttöliittymästä.

Editoinnin käyttöä muutettiin myös hieman: jokaisen täytettävän kentän vieressä olevien *Cancel* ja *Save* -painikkeiden sijaan tuotetietojen muokkaukset tallennettaisiin erikseen *Editor*-ikkunan alaosaan olevan *SAVE*-painikkeen kautta.

### 5.9.8 Käyttöliittymän kahdeksas versio

Kahdeksannessa versiossa ei tapahtunut suuria muutoksia edelliseen prototyyppiin verrattuna. Yläpalkista poistettiin testaa- ja arveluttaneet leikkaa-, kopioi- ja liitä-painikkeet siirtämällä nämä toiminnallisuudet Files-ikkunaan. Toiminnot löytyivät joko painamalla kolmen vaakapisteen ikonia jokaisen Files-ikkunassa luettoloidun tiedoston perässä, tai oikean hiirenpainikkeen takaa. Prototyyppien luomiseen käytetty Adobe Xd -ohjelma ei kuitenkaan tukenut oikealla painikkeella avattavia toimintoja, joten ei ollut mahdollista testata kuinka helposti käyttäjät löytäisivät nämä toiminnot.

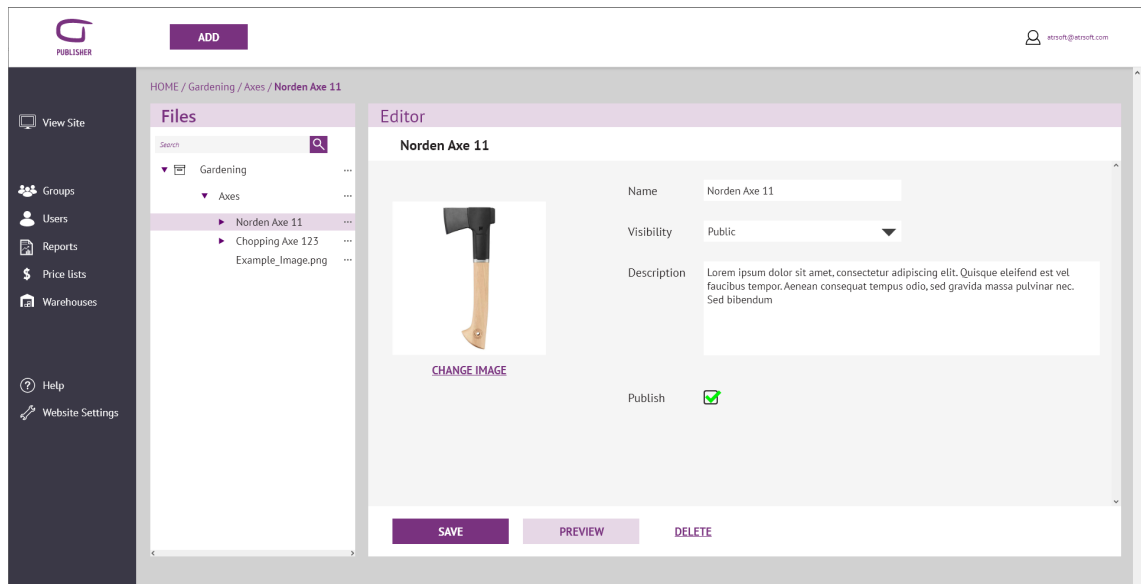
Myös sivupalkissa tapahtui pieniä muutoksia tässä versiossa. Tarpeettomia painikkeita poistettiin, sekä editointi-tilasta esikatselutilaan siirtyminen siirrettiin yh-

den saman napin taakse, jonka tehtävä riippuu siitä, missä tilassa ohjelma on. Editointitilassa painike näyttää *View Site* -tekstiä ja vie käyttäjän esikatselutilaan. Esikatselutilaan siirtyessä painikkeen teksti vaihtuu *Editor* -tekstiksi ja käyttäjä pääsee tällöin sitä klikkaamalla takaisin editointitilaan.

Tässä versiossa myös käyttöliittymän ulkoasua muutettiin enemmän tuotteen virallista graafista ohjeistusta vastaavaksi: kulmien pyöristykset poistettiin *Files* ja *Editor* -ikkunoista, kuin myös kaikista wizardeista. Tuotteen kuva siirrettiin myös *Editor*-ikkunassa tekstikenttien vasemmalle puolelle, jotta asetelma olisi yhtenevä sen kanssa miten kuva ja kentät on aseteltu tuotteiden lisäämisen wizardeissa.

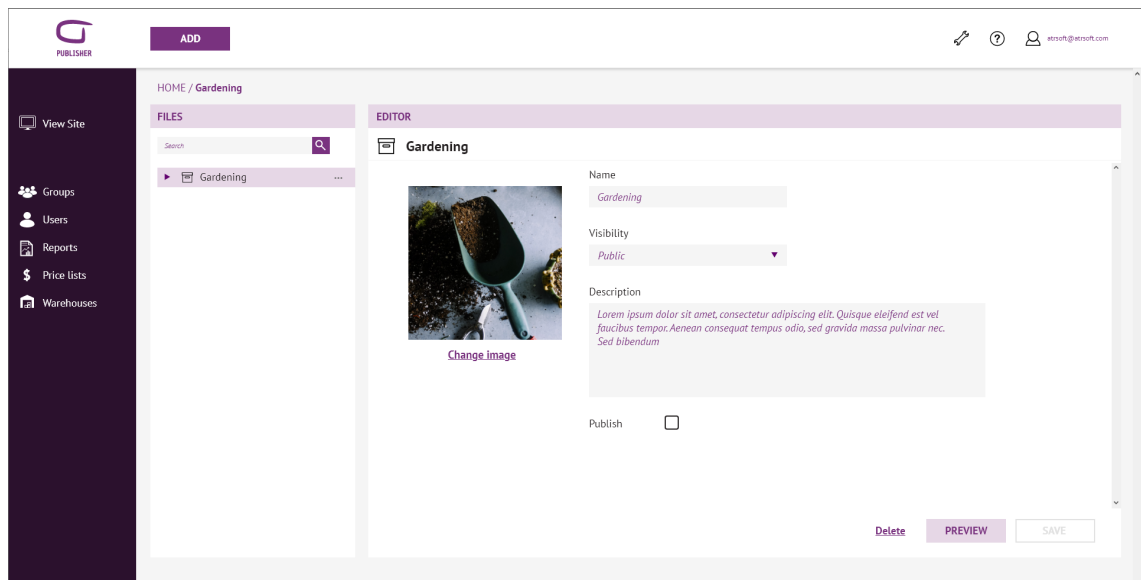
Kehitysprosessin loppupuolella, kun käyttöliittymän toiminnallisuus alkoi olla lähellä lopullista muotoaan, pystyttiin keskittymään enemmän graafiseen ulkoasuun. Graafisessa suunnittelussa oli mukana yrityksessä työskentelevä kokeneempi graafinen suunnittelija, jonka vastuualueisiin kuului muun muassa tuotteiden brändäys ja graafisista linjauksista päättäminen. Keskustelemalla graafisen suunnittelijan kanssa ja noudattamalla hänen tekemäänsä tuotteen graafista ohjetta päivitettiin CT Publisherin web-editorin käyttöliittymä vastaamaan ulkonäöltään ohjeistuksia. Käytännössä tämä tarkoitti käyttöliittymän nappien ulkoasun päivittämistä niin kooltaan, väritykseltään kuin sijainniltaan. Myös typografia piti päivittää vastaamaan tuotteelle määritettyä fonttia.





Kuva 5.15: Kahdeksas versio käyttöliittymästä.

### 5.9.9 Viimeistelty käyttöliittymä



Kuva 5.16: Käyttöliittymän ulkoasu suunnittelutyön lopussa.

Suunnittelutyön lopussa tehtiin enää pieniä korjauksia käyttöliittymään ennen implementointivaiheen aloittamista. Värimaailmaa muutettiin enemmän vastaamaan tuotteen graafista ohjeistusta. *Editor*-ikkunan alapainikkeet siirrettiin oikealle samasta syystä.

Syöttökenttien nimet *Editor*-ikkunassa siirrettiin kenttien vasemmalta puolelta kenttien yläpuolelle. Tämä tehtiin osana tuotteen tyyli vaatimuksia, mutta siirrolle löytyy myös muita perusteluja. Tutkimusten mukaan kenttien nimien ollessa suoraan kentän yläpuolella ihmissilmän on helpompi nähdä kentän sisältö ja nimi yhdellä silmäyksellä [31]. Nimien ollessa kentän vieressä tämä ei onnistu, varsinkin jos kenttien nimet ovat hyvin pitkät.

Tätä työtä tehdessä graafinen ohjeistus ei ollut kuitenkaan täysin valmis, joten ohjeistusta graafisten elementtien kanssa noudatettiin vain niin paljon kuin oli sillä hetkellä mahdollista. Ohjeistuksen päivittyessä myöhemmin lopulliseen muotoon tulisi siis myös käyttöliittymää samalla päivittää noudattamaan ja vastaamaan ohjeistusta.

### 5.9.10 Viimeinen testaus SUS

Tämän diplomityön kannalta viimeinen testaussessio suoritettiin aikaisemmista poiketen käyttämällä avuksi System Usability Scalea-työkalua arvioinnin tukena. Vaikka SUS ei anna todenmukaista kuvaa järjestelmän käytettävyydestä kun testauksen kohteena on pelkkä prototyyppi, haluttiin prototyypin käytettävyyttä kuitenkin mitata sen avulla, jotta saataisiin parempi käsitys siitä kuinka hyvä käytettävyys työn tuloksena saadulla prototyypillä on.

Arviointia varten käännettiin testaaajille SUS-arviointilomake suomenkieliseksi, koska haluttiin, että kaikki äidinkielenään suomea puhuvat testaaajat varmasti ymmärtävät lomakkeen kysymykset. Yksi arvioijista ei ollut käyttänyt mitään prototyyppiä ennen, eikä tiennyt mitään järjestelmästä. Muut arvioijat olivat olleet mu-

kana yhdessä tai useammassa testaussessiossa aikaisemmin.

#	Kysymys	A	B	C	D
1	Käyttäisin tätä järjestelmää usein.	4	4	4	3
2	Järjestelmä oli mielestäni tarpeettoman monimutkainen.	2	1	2	2
3	Järjestelmää oli mielestäni helppo käyttää.	5	4	4	4
4	Luulen, että tarvitsen teknisen henkilön tukea jotta voisin käyttää tätä järjestelmää.	2	3	1	1
5	Järjestelmän eri osat toimivat mielestäni hyvin yhteen.	5	4	4	3
6	Mielestäni järjestelmässä oli liian paljon eri lailla toimivia asioita.	2	2	2	2
7	Luulen, että useimmat oppivat käyttämään tätä järjestelmää hyvin nopeasti.	5	4	5	4
8	Järjestelmää oli mielestäni hyvin hankala käyttää.	1	1	1	1
9	Tunsin itseni hyvin itsevarmaksi käyttäessäni järjestelmää.	2	4	4	4
10	Minun piti opetella paljon asioita, ennen kuin pystyin käyttämään tätä järjestelmää.	1	2	2	1
	Vastaajan yhteenlasketut SUS pisteet:	82,5	77,5	82,5	77,5
	Pisteiden keskiarvo:	80			
	Arvosana	B			

Kuva 5.17: SUS testin tulokset.

Järjestelmä sai SUS-testauksessa 80 pistettä ja sen käytettävyyden arvosanaksi laskettiin B. Järjestelmän prototyypin käytettävyyttä voidaan siis pitää hyvänä, mutta parantamisen varaa kuitenkin on. Testaajien antamat pisteet eivät poikenneet toisistaan merkittävällä tavoin, eikä testin avulla näin ollen pystytty havaitsemaan erityisempiä käytettävyyden ongelmia.

## 5.10 Käyttöliittymän toteutuksen malli

Kun riittävä käytettävyyden taso oli todettu testaamisella, valmistettiin mockupista/prototyypistä virallinen dokumentaatio, mitä käytettiin ohjeena varsinaisessa toteutuksessa. Näin projektin ohjelmointityötä tekevät jäsenet voivat keskittyä käyttöliittymän toteuttamiseen ja dokumentaation seuraamiseen, eikä heidän tarvinnut tehdä päätöksiä ja ratkaisuja käytettävyyden suhteen.

### 5.10.1 Dokumentaatio

Käyttöliittymäsuunnittelutyön lähetessä loppuaan diplomityöosuutta koskien, sovitettiin tuotteen omistajan kanssa miten suunnittelutyö dokumentoidaan. Dokumen-

tointia varten haluttiin, että muilla kehitystiimin jäsenillä on helppoa ja nopeaa nähdä käyttöliittymän eri osa-alueet ja mitä toiminnallisuutta kuhunkin osa-alueeseen liittyy. Ensimmäinen idea oli kirjoitettu dokumentti, joka tallennettaisiin yrityksen verkkoon muiden saataville. Tätä ratkaisua pidettiin kuitenkin liian kömpelönä, sillä muutosten tapahtuessa dokumentti pitäisi päivittää ja sen jälkeen varmistaa, että kaikilla on varmasti uusin versio käytössään.

Parempana ratkaisuna pidettiin luoda yrityksen organisaatiowikiin, eli Confluenceen käyttöliittymän dokumentoinnille oma sivunsa. Tälle sivulle tallennettiin kuvakaappauksia käyttöliittymästä, sekä myös videoita prototyyppien käytöstä, jotta toiminnallisuuksia voitaisiin kommunikoida selkeämmin. Confluence-sivujen päivittäminen onnistuu myös helpommin kuin staattisen dokumentin ja näin vältetään myös dokumentin versionhallinnan ongelmilta.

Myös viimeistely Adobe Xd -tiedosto siirrettiin pilveen muiden saataville, jotta sitä voitaisiin käyttää helposti pohjana CT Publisherin web-editorin muiden toiminnallisuuksien käyttöliittymien suunnittelussa.

## 5.11 Käyttöliittymän implementointi

Alkuperäisen projektisuunnitelman mukaan käyttöliittymän toteutukselle oltiin jätetty kuukausi aikaa suunnittelutyön päätyttyä. Esitutkintavaihe vei kuitenkin oletettua enemmän aikaa, eikä ohjelmointityötä ehditty aloittaa diplomityöprojektin aikana.

Käyttöliittymä on tarkoitettu implementoida käyttäen Javascript:iä ja sen *Vue.js*-sovelluskehystä. *Vue.js* valittiin sen vuoksi, koska sen arvioitiin olevan nopeasti omaksuttavissa ja sillä saa tehtyä kevyitä front-end sovelluksia.

Käyttöliittymän implementointivaiheessa ohjelmoijien pitää suurella todennäköisyydellä tehdä kompromisseja käyttöliittymän toteutuksen ja teknisten mahdollisuuksien kanssa. On tärkeää jatkaa testausta myös tässä vaiheessa niin että käyt-

töliittymää testataan kun jotain sen elementtejä muutetaan. Jonkin käyttöliittymän suunnittelutyön ratkaisun toteuttaminen voi esimerkiksi viedä enemmän aikaa tai henkilöresursseja kun implementointiin on varattu, mistä johtuen käyttöliittymää joudutaan muokkaamaan. Jatkuvalle testaamiselle taataan, että tuotteen käytettävyyttä säilyy hyvänä.

## 5.12 Pilotointi ja julkaisu

Ennen lopullista julkaisua on myös syytä testata käytettävyyttä uudestaan käyttämällä aikaisempaa SUS-menetelmää käytettävyyden arviointiin. Koska järjestelmä ei tässä vaiheessa ole enää pelkkä prototyyppi, vaan kaiken toiminnallisuuden sisältävä oikea versio, voi SUS-testaamisella saatua arvosanaa pitää luotettavampana mittarina järjestelmän käytettävyydelle.

Ennen lopullista julkaisua olisi myös tärkeää, että järjestelmää saataisiin testattua oikeilla loppukäyttäjillä. Jos tilanne ja resurssit sallivat, olisi käytettävyydestien järjestäminen loppukäyttäjien oikeassa työympäristössä suotava ratkaisu laadukkaamman palautteen keräämiseksi.

# 6 Lopputulos ja johtopäätökset

## 6.1 Projektista

Projektin evaluointi on hyvin tärkeä osa prosessia, joka suoritetaan tuotteen julkaisun ja projektin päättymisen jälkeen. Evaluointi oli auttamatta diplomityöprojektin aikarajoitteiden ulkopuolella, mutta se tulisi tehdä prosessin seuraavassa vaiheessa. Evaluoinnin tärkeänä mittarina voidaan pitää ajankäyttöä: kuinka paljon käyttäjälähtöinen käyttöliittymäsuunnittelu vei aikaa verrattuna aikaisemmin ATR Soft:illa käytettyihin suunnitteluprosesseihin?

Diplomityöprojektin lopussa pidettiin tuotteen omistajan kanssa tapaaminen, jossa arvioitiin itse prosessia ja sen tuloksia. Evaluoinnissa käsiteltiin itse prosessia ja sen toimivuutta tämänkaltaisessa projektissa, sekä prosessissa kohdattuja haasteita.

Prosessi ja käyttäjälähteinen käyttöliittymäsuunnittelu koettiin onnistuneeksi. Käyttäjäläheisestä prosessista otettuja ideoita yritetään soveltaa enemmän tuotesuunnittelussa ja olla enemmän asiakaslähtöisiä tiimin sisällä. Toimivia ideoita ja metodeja koitetaan myös työntää muille kehitystiimeille. Varsinkin käyttöliittymäsuunnittelu prototyyppien avulla koettiin hyväksi lähestymistavaksi, jota tullaan tulevaisuudessa käyttämään enemmän.

Projektin aikataulusta pidettiin hyvänä, vaikka alkuperäistä aikataulutusta jouduttiinkin muuttamaan ja varsinainen ohjelmointivaihe jättämään diplomityön ulkopuolelle. Tuotteen omistaja oli myös yhtä mieltä siitä, että status-palaverejä pi-

dettiin sopivan usein ja niistä oli paljon hyötyä. Hyvin tehty prosessisuunnitelma ja prosessin tuloksena syntyneet artefaktit helpottivat tiedonvälitystä sidosryhmille.

Projektin suurin haaste oli puuttuva data projektin alussa. Tuotteen omistaja pahoitteli puutteellista pohjatyötä, kuten käyttäjien ja kohdemarkkinoiden määrittelyä jotka jouduttiin suorittamaan osana projektia. Toisena haasteena oli projektin aikana sattunut covid-19-epidemian aiheuttama poikkeustila. Epidemiasta johtuen työpajat ja testaussessiot piti järjestää etänä, mikä aiheutti omat haasteensa.

Hyvin menneinä asioina pidettiin varsinkin kilpailijoiden ja kilpailevien tuotteiden analyysyjä. Vaikka nämä eivät olleet työssä pääaiheessa, oli ne silti tehtävä jotta saataisiin riittävää pohjadataa suunnittelua varten. Myös käyttäjäpersoonia pidettiin onnistuneina. Käyttäjistä muodostetuista persoonista oli myös hyötyä muualakin kuin käyttöliittymäsuunnittelussa, esimerkiksi tuotteen markkinoinnin priorisoinnissa ja kohdentamisessa tärkeimmille avainpersoonille.

Omalta osaltani arvioisin projektin aikataulutuksen menneen osittain pieleen. Alunperin tarkoitus oli käyttää viimeinen työkuukausi käyttöjärjestelmän varsinaiseen toteutukseen ja ohjelmointiin, mutta aikataulutusta suunniteltaessa ei oltu vielä tietoisia tuotteen puutteellisista markkinatutkimuksista. Näin ollen alkuperäinen tavoite saada osa käyttöliittymän toteutuksesta ja ohjelmointityöstä sisällytettyä tähän diplomityöhön ei toteutunut. Myös käytettävyydestien järjestäminen vei suunniteltua enemmän aikaa: suurin osa vapaaehtoisista testaaajista oli vapaata aikaa vain iltaisin tai viikonloppuisin pidempiä käytettävyydestaus sessioita varten. Projektin loppua kohden myös testaaajien motivaatio sekä saadun palautteen laatu ja määrä heikkenivät hieman, mikä oli osittain odotettavissakin.

Paperiprototyyppien käyttöä pystyi pitämään osittain onnistuneena. Nopeasti luoduilla paperiprototyypeilla pääsi hyvin alkuun suunnittelutyössä, mutta niiden testaaminen etänä oli ongelmallista. Havaintojen kerääminen testauksesta olisi mahdollisesti onnistunut paremmin, jos testaaajien paperiprototyyppien käyttöä oltaisiin

pystytty seuraamaan samassa tilassa.

Tietokoneella tehdyt prototyypit osoittautuivat erinomaisiksi työkaluiksi suunnittelutyössä. Vaikka niiden avulla ei pystytty toteuttamaan lopullisen tuotteen käyttöä vastaavaa kokemusta, antoivat ne kuitenkin erinomaisen käsityksen lopullisen käyttöliittymän ulkoasusta ja toiminnoista.

Myös työn todellinen laajuus tuli yllätyksenä ja selvisi tekijälle vasta sitä tehdessä. Pelkästään käytettävyyden mittaamisesta ja testaamisesta, sekä niissä käytetyistä menelmistä olisi voinut kirjoittaa diplomityön. Työhön oli tarkoitus sisällyttää myös kuvaus toteutusprosessista ja siinä käytetyistä metodeista ja työkaluista, mutta ne jäivät ajanpuutteen takia tekemättä tämän työn puitteissa.

## 6.2 Menetelmästä

Valittu menetelmä oli hyvin painottunut alkupäässä tehtävään tutkimus ja suunnittelutyöhön. Leijonanosa projektiin käytetystä ajasta kului tutkimusvaiheessa, mikä on mielestäni välttämätöntä tämän kaltaisessa suunnittelutyössä. Käyttäjäläheinen suunnittelu vaatii loppukäyttäjien tuntemisen ja näihin tutustumiseen kannattaa käyttää aikaa, jotta suunnitteluprosessi pohjautuu mahdollisimman realistiseen ja luotettavaan dataan.

Suurimpana haasteena tutkimustyössä olikin datan kerääminen persoonia varten, ongelmana oli lähinnä erittäin kiireisten jälkimarkkinointihenkilöiden tavoittaminen ja heiltä datan saaminen persoonia varten. Lähestymistavaksi valitun kyselyn avulla saatiin kuitenkin riittävästi dataa kerättyä, jotta suunnitteluprosessissa päästiin eteenpäin. Olisi kuitenkin ollut suotavaa saada haastateltua tarkemmin loppukäyttäjryhmän edustajia.

Käyttöliittymän suunnittelutyö itsessään on jotain, mikä riippuu järjestelmän tarkoituksesta ja kohderyhmistä. Suunnittelijan osaaminen ja kokemus ovat kuitenkin tärkeitä. Vaikka käytetyt testit paljastivatkin käytettävyyden ongelmia, on silti



suunnittelijan itsensä varassa niiden korjaaminen. Suunnittelijalta vaaditaan myös luovuutta. Tietenkin prosessin alkuvaiheessa tehty kilpailevien tuotteiden kartoitus auttaa ymmärtämään yleisesti käytettyjä suunnittelutapoja ja käytäntöjä omaa tuotetta vastaavissa järjestelmissä.

Suunnittelijan osaaminen ja kokemus ovat kuitenkin tärkeitä. Vaikka käytetyt testit paljastivatkin käytettävyyden ongelmia, on silti suunnittelijan itsensä varassa niiden korjaaminen. Suunnittelijalta vaaditaan luovuutta, tietoa käyttöliittymien suunnittelukäytännöistä sekä graafista silmää. Tietenkin prosessin alkuvaiheessa tehty kilpailevien tuotteiden kartoitus auttaa ymmärtämään yleisiä suunnittelukäytäntöjä vastaavissa järjestelmissä, mutta suunnittelijan on osattava tunnistaa myös niiden ongelmat ja miten parannella käytettävyyttä verrattuna kilpailijoiden tuotteisiin.

### 6.3 Testeistä

Haasteena oli myös testeissä löydettyjen käytettävyyden ongelmien korjaus. Kuten aikaisemmin mainittiin, testeillä voidaan löytää käytettävyyden ongelmia, mutta ne eivät suoraan kerro, kuinka ongelmat korjataan. Ongelmiin koitettiin löytää vastausta kysymällä suoraan testaaajilta heidän ajatuksiaan siitä, miten heidän mielestään järjestelmän kuuluisi toimia sekä tutkimalla olemassa olevia vastaavia käyttöliittymiä.

Olisi ollut myös parempi, jos testit oltaisiin saatua suoritettua niin, että testaaajien kanssa oltaisiin voitu olla samassa tilassa heidän käyttäessään järjestelmää. Vaikka videopuheluiden yli tehty ruudun jakaminen antoi kohtuullisen hyvän käsityksen käyttäjien ongelmista ja siitä miten he käyttivät järjestelmää, olisi varmasti ollut hyödyllisempää saada testejä järjestettyä myös niin, että valvoja ja testaaaja olisivat samassa tilassa.

Testaaajien käytössä olisi voinut myös toimia paremmin. Koska jälkimarkkinoin-

nin ammattilaista ei ollut saatavilla testaaajaksi, suoritettiin käytettävyydestä henkilöillä, jotka vastasivat mahdollisimman hyvin jälkimarkkinointivastaavan persoonaa, jonka rooliin he myös testeissä asettuivat. Vaikka näistä testeistä saatiin hyvin palautetta, ennen tuotteen lopullista julkaisua olisi tärkeää saada testattua järjestelmää oikealla loppukäyttäjällä.

Mutta kuitenkin, kun todellinen loppukäyttäjä on vaikeasti lähestyttävissä, suositteisin käyttämään persoonia ja niitä vastaavia henkilöitä jotka ovat helposti saatavilla, koska käytettävyyttä pitää testata useamman kerran. Persoonien simulointi onnistuu helpommin kuin kiireellisen loppukäyttäjän saaminen osallistumaan useisiin testaussessioihin.

SUS-testejä olisi voinut käyttää aikaisemmin. Koska prototyyppiä testaamalla saatu pisteytys ei ole suoraan verrattavissa oikeiden järjestelmien pisteisiin, olisi ollut hyödyllisempää suorittaa jokaisen prototyypin kohdalla oma SUS-testaussessio. Näin prototyyppien pisteitä oltaisiin voitu verrata toisiinsa ja näin kerätty data olisi saattanut olla hyödyllisempää käytettävyyden parantumisen arvioinnissa prototyyppien välillä.

## 6.4 Yhteenveto

Diplomityön tavoitteena oli selvittää mitkä käyttäjakeskeisen suunnittelun työkaluita ja metodeista ovat hyödyllisimpiä käyttäen apuna case-tapausta tutkimuksessa. Tämän työn avulla lukija voi valita omaan tapaukseensa sopivimmat menetelmät.

Loppukäyttäjien löytämiseen ja tärkeimpien käyttäjien tunnistamiseen käytettyjä metodeja voitiin pitää onnistuneina. Tehokkaimmaksi keinoksi käyttäjien löytämiseksi havaittiin olevan kilpailevien tuotteiden käyttötapausten ja asiakkaiden tutkiminen. Tämän datan kerääminen ei vaatinut paljoa resursseja saatuihin hyötyihin nähden ja kerätystä datasta pystyttiin muodostamaan pohjat eri käyttäjäpersoonille.

Myös käyttäjäpersoonien tuomat hyödyt olivat niiden luomiseen käytettyihin resursseihin nähden huomattavia. Tässä case-tapauksessa kuitenkin osa persoonien luomisen hyödyistä jäi saavuttamatta, sillä suurimmat hyödyt personista saadaan kun ne luodaan heti projektin alussa, ei vasta siinä vaiheessa kun aletaan suunnittelemaan käyttöliittymää.

Prototyypin käyttäminen käyttöliittymän suunnitteluvaiheessa oli niin ikään tehokas menetelmä. Niin paperiprototyypeillä kuin suunnitteluohjelmilla toteutuilla tietokoneprototyypeillä saatiin kerättyä nopeasti ja helposti palautetta, sekä testattua erilaisia ratkaisuja käyttöliittymässä.

Koska tutkimusta tehtiin vain yhden case-tapauksen avulla, ei työn tuloksia voida yleistää. Valituilla menetelmillä ja työkaluilla kuitenkin pystyttiin luomaan käytettävyydeltään hyvä sovellus ja todentamaan kappaleissa 2 ja 3 esitettyjä teorioita sekä menetelmiä. Jatkotutkimuksia varten olisi suotavaa, jos tutkimus käyttäjäläheisestä suunnittelutavoista voitaisiin aloittaa jo tuotesuunnitteluprosessin alussa. Olisi myös hyvä päästä testaamaan valmista tuotetta ja sen käytettävyyttä, jotta saataisiin realistisempi kuva käytettävyydestä kuin testaamalla pelkkiä prototyyppejä.

Toivottavasti tämän työn lukemisen jälkeen lukijalle jää selkeä kuva käyttäjäläheisestä käyttöliittymäsuunnittelusta, sen eduista sekä siitä, miten mitata käytettävyyttä tehokkaasti ja luotettavasti.

# Lähdeluettelo

- [1] J. Nielsen, *Usability Engineering*. Academic Press, 1993, ISBN: 0-12-518406-9.
- [2] A. Cooper, R. Reimann ja D. Cronin, *About Face 3: The Essentials of Interaction Design*. Wiley Publishing, inc, 2007, ISBN: 978-0-470-08411-3.
- [3] J. Nielsen. (2000). ”*End of Web Design*”. Viitattu: 2020-11-1, url: <https://www.nngroup.com/articles/end-of-web-design>.
- [4] —, (2001). ”*Error Message Guidelines*”. Viitattu: 2020-03-02, url: <https://www.nngroup.com/articles/error-message-guidelines/>.
- [5] A. Cooper, *The Inmates Are Running the Asylum*. Paul Boger, 2004, ISBN: 0-672-32614-0.
- [6] S. Gibbons. (2015). ”*Empathy Mapping: The First Step in Design Thinking*”. Viitattu: 2020-03-14, url: <https://www.nngroup.com/articles/empathy-mapping/>.
- [7] T. Brinton. (2015). ”*User Stories: A Foundation for UI Design*”. Viitattu: 2020-08-16, url: <https://www.uxbooth.com/articles/user-stories-a-foundation-for-ui-design/>.
- [8] S. Gibbons. (2017). ”*UX Stories Communicate Designs*”. Viitattu: 2020-07-27, url: <https://www.nngroup.com/articles/ux-stories/>.
- [9] C. Snyder, *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. Morgan Kaufmann Publishers, 2003, ISBN: 978-1-55860-870-2.

- [10] S. P. Dow, J. Fortuna, D. Schwartz, B. Altringer, D. L. Schwartz ja S. R. Klemmer. (2011). ”*Prototyping Dynamics: Sharing Multiple Designs Improves Exploration, Group Rapport, and Results*”. Viitattu: 2020-04-01, url: <https://dl.acm.org/doi/pdf/10.1145/1978942.1979359>.
- [11] J. Nielsen. (2003). ”*Paper Prototyping: Getting User Data Before You Code*”. Viitattu: 2020-03-19, url: <https://www.nngroup.com/articles/paper-prototyping/>.
- [12] S. A. Laakso ja K.-P. Laakso. (2004). ”*Hyvän käyttöliittymän varmistaminen GUIDe-prosessimallilla*”. Viitattu: 2020-08-15, url: <https://www.cs.helsinki.fi/u/salaakso/papers/GUIDe-suomeksi.pdf>.
- [13] J. E. Arnold. (1959). ”*CREATIVE ENGINEERING: Promoting Innovation by Thinking Differently*”. Viitattu: 2020-11-21, url: <https://www.inist.org/library/1959.John%20E%20Arnold.Creative%20Engineering.pdf>.
- [14] P. G. Rove, *Design Thinking*. The MIT Press, 1987, ISBN: 0-262-68067-X.
- [15] R. F. Dam ja T. Y. Siang. (2020). ”What is Design Thinking and Why Is It So Popular?” Viitattu: 2020-11-22, url: <https://www.interaction-design.org/literature/article/what-is-design-thinking-and-why-is-it-so-popular>.
- [16] S. Gibbons. (2016). ”*Design Thinking 101*”. Viitattu: 2020-11-21, url: <https://www.nngroup.com/articles/design-thinking>.
- [17] K. J. Heffernan. (2017). ”*The Double Diamond model: what is it and should you use it?*” Viitattu: 2020-11-22, url: <https://medium.com/seek-blog/design-thinking-101-the-double-diamond-approach-ii-4c0ce62f64c7>.
- [18] D. Nessler. (2016). ”*How to apply a design thinking, HCD, UX or any creative process from scratch*”. Viitattu: 2020-11-22, url: <https://medium.com/>

- digital-experience-design/how-to-apply-a-design-thinking-hcd-ux-or-any-creative-process-from-scratch-b8786efbf812.
- [19] J. Nielsen. (2001). ”*Usability Metrics*”. Viitattu: 2020-03-10, url: <https://www.nngroup.com/articles/usability-metrics/>.
- [20] —, (2000). ”*Why You Only Need to Test with 5 Users*”. Viitattu: 2020-03-10, url: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>.
- [21] —, (1994). ”*10 Usability Heuristics for User Interface Design*”. Viitattu: 2020-05-12, url: <https://www.nngroup.com/articles/ten-usability-heuristics/>.
- [22] —, (1994). ”*Severity Ratings for Usability Problems*”. Viitattu: 2020-05-12, url: <https://www.nngroup.com/articles/how-to-rate-the-severity-of-usability-problems/>.
- [23] C. Wharton, J. Rieman, C. Lewis ja P. Polson. (1993). ”*The Cognitive Walkthrough Method: A Practitioner’s Guide*”. Viitattu: 2020-05-18, url: <https://www.colorado.edu/ics/sites/default/files/attached-files/93-07.pdf>.
- [24] M. H. Blackmon, P. G. Polson, M. Kitajima ja C. Lewis. (2002). ”*Cognitive Walkthrough for the Web*”. Viitattu: 2020-05-18, url: [https://www.researchgate.net/publication/221514534\\_Cognitive\\_walkthrough\\_for\\_the\\_Web](https://www.researchgate.net/publication/221514534_Cognitive_walkthrough_for_the_Web).
- [25] J. Nielsen ja R. L. Mack, *Usability Inspection Methods*. Wiley, 1994, ISBN: 978-0-471-01877-3.
- [26] J. Brooke. (1986). ”*SUS-A quick and dirty usability scale*”. Viitattu: 2020-05-15, url: <https://hell.meiert.org/core/pdf/sus.pdf>.

- 
- [27] J. Sauro. (2011). ”*Measuring usability with the System Usability Scale (SUS)*”. Viitattu: 2020-05-15, url: <https://measuringu.com/sus/>.
- [28] J. Rubin ja D. Chisnell, *Handbook of Usability Testing, Second Edition: How to Plan, Design, and Conduct Effective Tests*. Wiley Publishing, Inc., 2008, ISBN: 978-0-470-18548-3.
- [29] D. E. Gray, *Doing Research in the Business World*. SAGE Publications Ltd, 2017, ISBN: 978-1-4739-1567-1.
- [30] R. Budiu. (2017). ”*Wizards: Definition and Design Recommendations*”. Viitattu: 2020-04-22, url: <https://www.nngroup.com/articles/wizards/>.
- [31] M. Penzo. (2006). ”*Label Placement in Forms*”. Viitattu: 2020-08-16, url: <https://www.uxmatters.com/mt/archives/2006/07/label-placement-in-forms.php>.

# Liite A Tutkimuskysely

Tämän kyselyn on tarkoitus selvittää eri after sales -ketjussa työskentelevien henkilöiden työnkuvaa ja prosesseja. Vastauksista saatua dataa käytetään tutkimuksessa diplomityötä varten.

1. Tämän kyselyn suorittaa Turun yliopiston maisterivaiheen opiskelija Markku Tuomola osana diplomityötään. Yksilöiviä henkilötietoja ei kerätä kyselyssä. Kysely sulkeutuu 13.3.2020 ja sen tekeminen kestää arviolta noin 5-10 minuuttia. Haluatko osallistua tähän kyselyyn?

Kyllä/Ei

## A.0.1 Yleiskysymyksiä

Näillä kysymyksillä halutaan selvittää yleisiä asioita after sales -ketjun työntekijöistä

2. Kuvailisitko mitä työnkuvaasi kuuluu?

3. Työskenteletkö läppärillä, pöytäkoneella vai tabletilla?

4. Mitä käyttöjärjestelmää suosit?



5. Top 3 eniten käyttämäsi sovellukset?

6. Käytätkö mitään seuraavista ohjelmista:

- Google Drive

- Dropbox

- OneDrive

7. Top 3 eniten käyttämäsi mobiilisovellukset?

### **A.0.2 Kysymyksiä varaosa/huoltopalveluiden prosesseista**

8. Kuvailisitko miten lisäät tuotteen/varaosan? (Esimerkiksi "haen kuvat mistä", mitä ohjelmia käytät)

9. Mistä saat tarvittavat tiedot varaosiin mitä olet lisäämässä? (Ihmisiltä, katalogista tms.)

10. Onko kaikki tieto aina saatavilla heti, vai lisäätkö jotain myöhemmin?

Kyllä/Ei

11. Tarvitseeko sinun tarkistaa tai muokata varaosien tietoja usein?

Kyllä/Ei

12. Keräättekö jotain dataa tuotteistanne (klikkejä, kysymyksiä, myynnin määrää)?

Kyllä/Ei

13. Mihin tarkoitukseen käytätte kerättyä dataa? (Jätä vastaamatta, jos vastasit "Ei" edelliseen)

# Liite B Järjestelmän käytettävyysarvio

Vapaa käänös System Usability Scale (SUS) arviointilomakkeesta.

Arviointiasteikko:

- 1) Täysin eri mieltä
- 2) Eri mieltä
- 3) Ei mielipidettä
- 4) Samaa mieltä
- 5) Täysin samaa mieltä

Vastaa kaikkiin kysymyksiin, kiitos.

1. Käyttäisin tätä järjestelmää usein.
2. Järjestelmä oli mielestäni tarpeettoman monimutkainen.
3. Järjestelmää oli mielestäni helppo käyttää.

- 
4. Luulen, että tarvitsen teknisen henkilön tukea jotta voisin käyttää tätä järjestelmää.
  5. Järjestelmän eri osat toimivat mielestäni hyvin yhteen.
  6. Mielestäni järjestelmässä oli liian paljon eri lailla toimivia asioita.
  7. Luulen, että useimmat oppivat käyttämään tätä järjestelmää hyvin nopeasti.
  8. Järjestelmää oli mielestäni hyvin hankala käyttää.
  9. Tunsin itseni hyvin itsevarmaksi käyttäessäni järjestelmää.
  10. Minun piti opetella paljon asioita, ennen kuin pystyin käyttämään tätä järjestelmää.