

Excel-työkalun kehitys tuotevastaavuuksien hakuun

Tiivistelmä

Tekijä(t) Minna Siivo	Julkaisun laji Opinnäytetyö, AMK Sivumäärä 34	Valmistumisaika 2023
Työn nimi Excel-työkalun kehitys tuotevastaavuuksien hakuun		
Tutkinto ja koulutusala Tradenomi (AMK), tietojenkäsittely		
Toimeksiantajaorganisaatio (jos opinnäytetyöllä on toimeksiantaja) Tehohydro Oy		
Tiivistelmä <p>Opinnäytetyön tavoitteena oli suunnitella ja toteuttaa tuotevastaavuuksien löytämisen tueksi työkalu. Työkalulla haluttiin automatisoida prosessia, joka oli tähän asti vaatinut paljon manuaalista työtä.</p> <p>Työkalu toteutettiin Excel-makroilla hyödyntäen Visual Basic for Applications -ohjelmointikieltä. Excel-makrojen käyttö mahdollisti työkalun helpon käyttöönoton yhteistyökumppanin tietokoneilla. Työkalun tietokannasta ei löydy kaikkia yhteistyökumppanin käyttämiä tuotteita, mutta se helpottaa vastaavuuksien selvittämisessä ja nopeuttaa aiemmin vaivalloiseksi koettua työvaihetta. Paikallisen tietokannan lisäksi työkalulla pystytään tekemään automatisoituja tuotevastaavuushakuja kahden eri valmistajan nettisivujen hakukoneilta.</p> <p>Työkalu toimii suunnitellusti, koekäytössä se kuitenkin sai loppukäyttäjiltä lisäkehitysideoita. Kaikkia ideoita ei ollut mahdollista toteuttaa annetussa aikataulussa valmiiksi lopulliseen työkaluun. Näin ollen tehtyä työkalua ja sen käyttämää tietokantaa on mahdollista edelleen kehittää tunnistamaan yhä uusien valmistajien tuotteiden korvaavuuksitietoja.</p>		
Asiasanat käytettävyys, käyttäjäkokemus, käyttöliittymäsuunnittelu, ohjelmistosuunnittelu, VBA		

Abstract

Author(s) Minna Siivo	Type of Publication Thesis, UAS	Published 2023
	Number of Pages 34	
Title of Publication Development of an Excel tool for identifying compatible parts		
Degree, Field of Study Bachelor of Business Administration, Business Information Technology (UAS)		
Organization of the client (if the thesis work is commissioned by another party) Tehohydro Oy		
Abstract <p>The purpose of the thesis was to design and implement a tool to automate a process of finding compatible products from multiple manufacturers. The process of finding this information used to be time consuming and required a lot of tedious manual work.</p> <p>The tool was implemented with Excel macros using the Visual Basic for Applications programming language. Excel macros enable easy deployment of the tool. Part of the product data was provided by the company as Excel spreadsheets and PDF files. The provided data was augmented with automated web queries into the parts manufacturers search-engines. The tool combines data from multiple sources to be accessible through single spreadsheet.</p> <p>To ensure that the tool works as intended user feedback was collected, and it gave some additional development ideas for the tool. Some of the improvement ideas were too vast to implement with the time allocated to the project. Therefore, it is possible to further develop the tool and the database which it utilizes, to identify data from even wider range of manufacturers.</p>		
Keywords software design, usability, user experience, user interface design, UX, VBA		

Sisällys

1	Johdanto.....	1
2	Laskentataulukkosovelluksen suunnittelu	3
2.1	Laskentataulukkosovellus	3
2.2	VBA	4
2.3	Ohjelmistoprojektin vaiheet.....	4
2.4	Vaatimusmäärittely	5
2.5	Visuaaliset elementit.....	6
2.6	Käyttöliittymän muotoilu	7
2.7	Ohjelmistoprojektin dokumentointi	8
2.8	Ohjelmiston toiminnallisuuden arviointi	9
3	Käytettävyys ja käyttökokemus	10
3.1	Hyvän käytettävyyden periaatteet	10
3.2	Vuorovaikutussuunnittelu.....	12
3.3	Käyttäjäkokemus	15
3.4	Käyttäjästävällisyys.....	15
3.5	Arvokeskeinen suunnittelu	16
4	Työkalun suunnittelu.....	18
4.1	Toimeksianto	18
4.2	Menetelmän valinta.....	18
4.3	Vaatimukset työkalulle	20
4.4	Toiminnallisuuksien suunnittelu	22
5	Työkalun toteutus	23
5.1	Työn vaiheet.....	23
5.2	Työkalun rakenne	24
5.3	Käytettävyyden arviointi.....	29
6	Yhteenveto ja pohdinta	31
	Lähteet	33

1 Johdanto

Digitalisaation myötä yrityksissä käsiteltävän datan määrä on kasvanut valtavasti. Tietoa on saatavissa helposti, ja sitä kerätään monista eri lähteistä. Muistikapasiteetti ei nykyisin rajoita säilytettävän tiedon määrää, sillä erilaisista pilvipalveluista tallennustilaa on saatavissa suhteellisen edulliseen hintaan. Haasteeksi onkin muodostunut kyky käsitellä ja hyödyntää saatavissa olevaa valtavaa tietomäärää tehokkaasti.

Microsoft Excel mahdollistaa helposti käyttöönotettavien laskentataulukkosovellusten kehittämisen tiedonhallinnan tueksi. Excelissä on sisäänrakennettuna tuki *Visual Basic for Applications (VBA)* ohjelmointikielelle. VBA:n avulla on mahdollista toteuttaa niin pieniä ja yksinkertaisia kuin hyvinkin isoja ja monipuolisia sovelluksia, joiden käyttöönoton vaatimuksena on vain koneelta löytyvä riittävän uusi versio Excel-ohjelmistosta.

Tässä opinnäytetyössä toteutetaan Excel-laskentataulukkosovellus tuotteiden yhteensopivuuksien löytämisen tueksi. Opinnäytetyön tavoitteena on toteuttaa toimeksiantaja yrityksen, Tehohydron, käyttöön työkalu, joka auttaa löytämään tietyn tuotekategorian tuotteista tietoja eri valmistajien keskenään yhteensopivista tuotteista.

Tehohydro Oy on vuonna 1983 perustettu lappeenrantalainen hydraulikkaan ja pneumaatiikkaan erikoistunut asiantuntijayritys. Sen palveluihin kuuluvat hydraulikka- ja pneumaattikkajärjestelmien kunnossapito- ja huoltotyöt, modernisoinnit, asennukset sekä uudisprojektit asiakkaiden tuotantolaitoksissa. Asiakkaina heillä on teollisuuden pieniä ja suuria tehtaita, konepajoja, energialaitoksia, julkishallinto sekä erilaisia muita yrityksiä. (Tehohydro.)

Asiakkaita palvellaan sekä projektimuotoisesti että kumppanuuspohjaisesti. Hydraulikka- ja pneumaattikkajärjestelmien kunnossapitotöiden lisäksi projekteihin voi kuulua myös koneasennuksia, metallitöitä ja hitsauksia. (Tehohydro.)

Toimeksiantajan puolelta ei ole rajattu työkalun koodaamiseen käytettävää alustaa tai ohjelmointikieltä eikä asetettu muitakaan reunaehtoja käyttöliittymän toteutukselle. Tähän asti toimeksiantaja oli käyttänyt tuotevastaavuuksien selvittämiseen pääasiassa eri valmistajien tarjoamia Excel- tai pdf-listauksia. Näiden lukuisten listojen läpikäyminen vaatii paljon manuaalista ja tarkkuutta vaativaa työskentelyä. Automatisoimalla tuotevastaavuuksien etsintäprosessia pystytään säästämään työntekijöiden käyttämää aikaa, sekä kasvattamaan löydettyjen tuotevastaavuuksien määrää. Työkalun avulla tehty työvaiheiden automatisointi tuo yritykselle kustannussäästöjä, kun sekä työajankäyttöä ja työn tulosten tarkkuutta pystyttiin tehostamaan.

Exceliä ja VBA:ta hyödyntämällä pystytään toteuttamaan työkalun toiminnallisuuden koodauksen lisäksi helppokäyttöinen ja selkeä käyttöliittymä, jolla työkalun käyttäminen onnistuu ilman, että sen käyttöä tarvitsee työntekijöille erikseen kouluttaa. Excelin käyttäminen mahdollistaa myös laajojen tietomäärien esittämisen selkeässä ja helposti luettavassa muodossa.

Työkalun kehittäminen alkaa tutustumalla toimeksiantajan toimittamaan lähdeaineistoon, joista tuotevastaavuuksia on tarkoitus löytää ja yhdistellä. Lähdeaineisto koostuu Excel- ja pdf-tiedostojen lisäksi kahden eri valmistajan verkkosivuilta löytyvistä hakukoneista. Verkkosivuilta saatavan tiedon lisäksi työkalu rajataan lukemaan vain Excel-tiedostoja, joten yhteistyökumppanin toimittamat pdf-tiedostot on muunnettava Excel-taulukoiksi, ennen kun työkalu osaa niiltä löytyvät tiedot lukea.

Hakuominaisuuden lisäksi toimeksiantajan toiveena on, että työkalu mahdollistaa myös uusien vastaavuuksien lisäämisen aineistoon. Uusia yhteensopivuustietoja voi tulla työntekijälle vastaan käytännön työtehtävien parissa työskenneltäessä tai niitä löydetään tuotevalmistajien verkkosivujen kautta. Kaikkia yhteistyökumppanin käyttämiä tuotteita ja niiden vastaavuuksien ristitaulukoita ei ole saatavissa omina tiedostoina tuotteiden valmistajilta.

Tehohydrolla on useita eri työtehtäviä, joissa työkalua pystytään hyödyntämään. Työkalulle yksi tärkeä ominaisuus on helppokäyttöisyys. Työkalun käytön pitäisi onnistua kaikilta sitä tarvitsevilta. Pyrkimyksenä on, että työkalun toimintojen oppinen ei vaadi käyttöohjeita. Työkalun kehityksessä huomioidaan se, että työkalun toiminnallisuuden saattaa olla tarpeen kehittää myös jatkossa. Ohjelmakoodista tehdään helppolukuista, mikä helpottaa muutoksien tekemistä jatkossa.

Työkalu toteutetaan etsimään yhteensopivuustietoja vain yhden tuoteryhmän tuotteista. Varsin pienillä muutoksilla työkalun avulla olisi mahdollista etsiä tietoa yhteensopivuuksista myös muissa tuoteryhmissä. Tätä muokattavuutta ei kuitenkaan toteuteta tässä opinnäytetyössä.

2 Laskentataulukkosovelluksen suunnittelu

2.1 Laskentataulukkosovellus

Walkenbach (2013, 97) määrittelee laskentataulukkosovelluksen laskentataulukkotiedostoksi (tai useammaksi toisiinsa liittyväksi tiedostoksi), joka on suunniteltu jonkun muun kuin ohjelmoijan itsensä käytettäväksi. Laskentataulukkosovelluksella käyttäjän pitäisi pystyä tekemään hyödyllisiä toimintoja ilman pitkää opettelua.

Työnkulun automaatio vähentää manuaalisen työn tarvetta. Kun automaatiolla yhdistetään useat työvaiheet yhden komennon taakse, hyötynä on lisääntynyt nopeus, tehokkuus ja työn laadun parantuminen. Tämä johtaa myös taloudellisiin säästöihin ja kasvavaan asiakastytyväisyyteen. Osana digitalisaation aiheuttamaa vallankumousta ohjelmistorobotiikka tai työkulun automatisaatio tulee vaikuttamaan merkittävästi työpaikkojen tulevaisuuteen. Työvaiheiden automatisointia voi periaatteessa hyödyntää kaikenlaisessa tietojenkäsittelyyn liittyvissä tehtävissä, sillä ohjelmistokoodilla voidaan hoitaa samoja rutiinimaisia tehtäviä samoilla ohjelmistoilla kuin ihminen niitä tekisi. (Hänninen 2022, 193–197.)

Tilajalle tehtävässä laskentataulukkosovelluksen kehitystyössä on oleellista, että jo suunnittelussa otetaan huomioon sovelluksen käytettävyys. Hyvän laskentataulukkosovelluksen ominaisuuksiin kuuluu se, että se mahdollistaa loppukäyttäjän tehdä sellaisia tehtäviä, joiden tekeminen olisi ilman sovellusta haasteellista. Tärkeää on myös se, että sovellus tekee ne tehtävät, joita sen oletetaan tekevän luotettavasti ja virheettömästi. (Walkenbach 2013, 97.)

Laskentataulukkosovellusta suunniteltaessa on Walkenbachin (2013, 98) mukaan pyrittävä löytämään käyttöön tehokkaat menetelmät ja funktiot annettujen tehtävien suorittamiseen. Koodin kirjoittamisessa huomioidaan se, että virheet saadaan kiinni, ennen kuin käyttäjä joutuu kohtaamaan niitä. Lisäksi huolehditaan siitä, ettei käyttäjällä ole mahdollisuutta muuttaa tai poistaa ohjelman toiminnan kannalta tärkeää tietoa. Hyvässä laskentataulukkosovelluksessa käyttöliittymä on selkeä, jotta loppukäyttäjä tietää, miten ohjelma toimii ja miten hän pystyy etenemään työtehtävänsä kanssa. Käytetyt funktiot on hyvin dokumentoitu ja rakennettu siten, että koodiin pystyy tekemään myöhemmin muutoksia, mikäli muutoksille ilmenee tarvetta.

Laskentataulukkosovellus on helposti käyttöön otettavissa, sillä se saadaan toimimaan millä tahansa laitteella, johon sovelluksesta, eli tässä tapauksessa Microsoft Excelistä on asennettu riittävän uusi versio. Laskentataulukkosovelluksille voidaan löytää monenlaisia

käyttökohteita, lisäksi on mahdollista tehdä laskentataulukkosovelluksia useille erilaisille käyttäjäryhmille. Sovelluksesta on mahdollista rakentaa hyvin yksinkertainen, lomake tyyppinen ohjelma, jossa käyttäjät antavat vastauksia valmiiksi määriteltyihin kysymyksiin tai hyvin monipuolinen sovellus, joka ei ulkoisesti edes muistuta laskentataulukkoa. (Walkenbach 2013, 98.)

Kun uutta ohjelmistoa aletaan suunnitella, on ensin selvitettävä tarkasti, millaisia tarpeita loppukäyttäjillä sovellukselle on. Jos käyttäjätarpeita ei arvioida huolellisesti kehittämissä varhaisessa vaiheessa, johtaa se herkästi ylimääräisiin työvaiheisiin kehitystyön myöhemmissä vaiheissa. Käyttäjävaatimusten määrittelyn jälkeen on hyvä käyttää aikaa sovelluksen toteutuksen suunnitteluun ennen varsinaisen ohjelmointityön aloittamista. Suunnitteluvaiheessa kannattaa pohtia ainakin laskentataulukkosovelluksen tiedostorakennetta, eli sitä toteutetaanko sovellus yhdelle Excel-tiedostolle ja siinä oleville useille laskentataulukoille vai usealle yksitaulukkoisille tiedostoille. Toiseksi on hyvä suunnitella myös se, miten sovelluksen käyttämät tiedot on paras tallentaa, tarvitaanko sovelluksen käyttöön esimerkiksi ulkoinen tietokanta. (Walkenbach 2013, 99–100.)

2.2 VBA

Ensimmäisen kerran *VBA (Visual Basics for Applications)* tuotiin markkinoille Excel 5 -ohjelmiston mukana vuonna 1994. *VBA*:ta pidetään Microsoftin yleisimmin käytettynä sovelluksen komentosarjakielenä ja nykyisin se sisältyy suurimpaan osaan Microsoftin Office paketin ohjelmistoista. Lisäksi sitä voi käyttää joissakin muiden tuottajien valmistamissa sovelluksissa. Näin ollen, jos osaa hyödyntää *VBA*:ta Excel-ohjelmistossa, on helposti mahdollista kirjoittaa makroja myös muihin Microsoftin (sekä joidenkin muiden valmistajien) tuotteisiin. Lisäksi on mahdollista ohjelmoida kokonaisratkaisuja, joiden toiminnot hyödyntävät useita eri ohjelmistoja. (Walkenbach 2013, 120.)

VBA on olio-ohjelmointikieli, ja eri ohjelmistojen hyödyntäminen vaatii ymmärryksen vain siitä, miten näiden eri ohjelmistojen oliot rakentuvat. Excelissä *VBA* pystyy hyödyntämään siitä löytyviä sisäänrakennettuja ominaisuuksia, kuten laskentataulukoita, pivot-taulukoita, kaavioita, sekä matemaattisia kaavoja. *VBA*:n avulla näitä ominaisuuksia voidaan hallita ja siten automatisoida erilaisia prosesseja monipuolisesti. (Walkenbach 2013, 120.)

2.3 Ohjelmistoprojektin vaiheet

Ohjelmistojen tuotantoon on olemassa paljon erilaisia lähestymistapoja. Valittavaan lähestymistapaan vaikuttavat muun muassa ohjelmistoprojektin koko ja se tuleeko ohjelmisto

omaan käyttöön vai valmistetaanko se asiakkaalle tilaustyönä. (Haikala & Mikkonen 2011, 29–30.)

Yleisimmin ohjelmistoprojektiin kuuluu ainakin määrittelyä, suunnittelua, ohjelmointia ja testausta sekä käyttöönottoon ja ylläpitoon liittyviä asioita. Nämä ohjelmistoprojektin osat pystytään näkemään vaiheina, joilla on alku- ja loppuajat. Osa näistä vaiheista voi toistua useamman kerran saman ohjelmistoprojektin aikana, ja osa vain kerran. (Haikala & Mikkonen 2011, 30.)

Ohjelmistoprojekti koostuu yleensä erilaisista etapeista, joilla on sidokset kehitystyön eri vaiheisiin. Projektin ensimmäiset etapit sisältävät esitutkimustyyppistä työskentelyä, jossa selvitetään esimerkiksi projektissa käytettävä toteutusteknologia. Esitutkimusvaiheessa pyritään varmistamaan se, että saadaan riittävästi tietoa varsinaista tuotekehitysprojektia varten. Riittävän tiedon keräämisellä pyritään minimoimaan ohjelmiston kehitystyöhön, aikatauluun ja käyttöönottoon liittyviä riskejä (Juvonen 2018, 61–62.)

Ohjelmistosuunnittelussa vaatimustenmäärittelyllä on ratkaiseva merkitys projektin onnistumisen kannalta. Haikalan & Mikkosen (2011) mukaan useat tutkimukset osoittavat, että yli 60 % ohjelmistoprojektien epäonnistumisista johtuu nimenomaan puutteista vaatimusten määrittelyssä. Vaatimus on ohjelmistotuotteessa, jotain mitä tuotteella pystyy tekemään tai ominaisuus, joka tuotteesta tulee löytyä. Vaatimukset voidaan jakaa kolmeen eri kategoriaan, joita ovat toiminnalliset vaatimukset, ei-toiminnalliset vaatimukset ja sovelluksen reunaehdot.

Ohjelmistoprojektin määrittelyllä tarkoitetaan asiakasvaatimusten ja niistä johdettujen ohjelmiston toimintojen dokumentointia asiakasnäkökulmasta. Suunnittelu tarkoittaa ohjelmiston teknistä suunnittelua. Testauksella yritetään löytää järjestelmästä mahdolliset virheet. Projektin valmistuttua se otetaan käyttöön lopullisessa toimintaympäristössä, jonka jälkeen alkaa ohjelmistoprojektin ylläpitovaihe. (Haikala & Mikkonen 2011, 30.)

2.4 Vaatimusmäärittely

Uuden palvelun tai sovelluksen suunnittelua aloitettaessa on usein olemassa vain osittainen ja sumea kuva siitä, millainen valmiin sovelluksen pitäisi olla. Vaatimusmäärittelyvaiheen tavoitteena on muuntaa tämä mielikuva selkeäksi kuvaukseksi uudesta sovelluksesta tai palvelusta. Vaatimuksia on usean tyyppisiä. Toiminnalliset vaatimukset kuvaavat sitä, mitä sovelluksella voi tehdä. Käytettävyyksivaatimukset selvittävät sen, millaisia ominaisuuksia sovelluksessa tulisi olla, jotta saavutettaisiin haluttu käyttäjäkokemus. Lisäksi sovellukselle asetetaan toimintaympäristövaatimuksia. Niitä ovat fyysisen ympäristön lisäksi sosiaalinen ympäristö ja se millaisissa olosuhteissa ja millaisilla laitteilla sovellusta

pitää pystyä käyttämään. Vaatimusmäärittelyssä on huomioitava myös projektiin liittyviä vaatimuksia, kuten esimerkiksi aikataulu, kustannukset ja riskittömyys. (Sinkkonen ym. 2009, 49.)

Projektit alkavat yleensä liiketoiminnallisista tarpeista. Uutta sovellusta voidaan kehittää toteuttamaan uusia ansaintamahdollisuuksia tai tuomaan kustannussäästöjä. Projektin liiketoiminnallisena perustana voi olla myös tarve päivittää olemassa olevaa sovellusta nykyaikaisemmaksi tai korjata vanhassa järjestelmässä olevia vikoja sekä puutteita. Liiketoiminnallisia tarpeita on pohjimmiltaan vain yksi: raha. Rahan määrää voidaan kasvattaa kahdella keinolla, joko karsimalla kustannuksia tai lisäämällä voittoja. Liiketoiminnalliset tavoitteet voivat olla joko selkeitä numeroita esimerkiksi myynnin kasvattaminen tietyllä euromäärällä kuukaudessa tai epämääräisempiä, mutta tähtäimenä on silti liikevoiton kasvuun. (Sinkkonen ym. 2009, 51.)

Sinkkonen ym. (2009, 50—59.) korostavat, että strategisen tason suunnittelussa määrittelyyn perusteet sille, mitä ollaan tekemässä eli se mitä hyötyä toteutettavasta sovelluksesta tai palvelusta halutaan saada. Strategiasuunnittelunvaiheessa selkiytyy se, mihin käyttäjätarpeisiin halutaan vastata ja miten se tehdään. Yksi keino kerätä tietoa käyttäjätarpeista on järjestää tarveanalyysityöpajoja tai -haastatteluja. Työpajan avulla voi nopeasti selvittää, ovatko kohderyhmän käsitykset sovelluksen toiminallisistatarpeista jotakuinkin realistiset. Käyttäjätöpajat kuitenkin hyvin harvoin riittävät kaikkien käyttäjätarpeiden ja toimintaolosuhteiden selvittämiseen, sillä se miten ihmiset toimivat ja miten he sanovat toimivansa voivat olla kaksi eri asiaa. Ihmisten työskentelyä heidän omassa toimintaympäristössään seuraamalla saa tarkempia ja oikeampia vastauksia käyttäjätarpeiden määrittelyn tueksi.

2.5 Visuaaliset elementit

Ohjelmiston visuaalinen ulkonäkö välittää käyttäjille kahta viestiä, joista ensimmäinen on sisällön ja käyttäjän toimintamahdollisuuksien esittäminen visuaalisessa muodossa. Sovelluksen ulkonäön tulee auttaa käyttäjää huomaamaan, jäsentämään ja ymmärtämään ne asiat, jotka ovat sovelluksen käyttämisen kannalta välttämättömiä. Tämä on sovelluksen visuaalista käytettävyyttä. (Sinkkonen ym. 2009, 242.)

Toinen ulkoasun tehtävä on välittää käyttäjälle viestiä sovelluksen kokonaisilmeestä. Tähän sisältyy palvelun tai yrityksen brändin välittäminen sovelluksessa. Yleisilmeen tulee tavalla tai toisella kuvastaa juuri kyseessä olevaa yritystä. (Sinkkonen ym. 2009, 242.)

Visuaalisen suunnittelun keinoja ovat Sinkkonen ym. (2009, 251) mukaan muun muassa katseen ohjaaminen, visuaalinen hierarkia, ryhmittely, sommittelu sekä värivalinnat. So-

vellukseen asetetut elementit ohjaavat käyttäjän katsetta. Taitavalla visuaalisella suunnittelulla saadaan käyttäjän katse ohjattua oikeaan paikkaan. Tärkeää on suunnitella kokonaisuus siten, että käyttäjä löytää kaikki tarvittavat elementit ja pystyy visuaalisten vihjeiden avulla ymmärtämään niiden käyttötarkoituksen.

Värit ovat voimakas osa sommittelua, ihmisten reagointi väreihin on usein hyvin tunteenomaista. Kovin kylläisiä, puhtaita ja lämpimiä värejä ei kannata käyttää taustaväriä. Värien käytössä on hyvä huomioida myös se, että mitä pienempi tummuuskontrasti on tekstin ja taustan välillä, sitä suurempaa kirjaisinkokoa täytyy käyttää. Vaikka värien lisääminen sovellukseen on helppoa, on kuitenkin hyvä muistaa, että usein hallittu ja maltillinen värien käyttö antaa sovelluksesta toimivamman ja tyylikkäämmän vaikutelman. Lopulta käyttäjät näkevät suunnitellut värit vain suurin piirtein suunnitellun kaltaisina. Värien toistuminen riippuu aina käyttäjän näytön kunnosta ja säädöistä. (Sinkkonen ym. 2009, 252–254.)

2.6 Käyttöliittymän muotoilu

Sovelluksen suunnittelussa on tärkeä huomioida käyttöliittymän muotoon liittyviä näkökulmia. Muodon suunnittelu vaikuttaa yhtä lailla sovelluksen käytettävyyteen kuin sen esteettisiin ja symbolisiin arvoihin. Ihmisten välillä on suuria eroja siinä, kuinka keskeiseksi he mieltävät tuotteiden muotoon liittyvät kysymykset. Kapeasti ymmärrettynä muoto viittaa tuotteen kosmeettiseen pintaan. Näin ajateltuna tuotteen muoto on helposti muokattavissa vielä senkin jälkeen, kun käyttöliittymä on muutoin valmis. Mikäli muoto nähdään näin kapeasti, se ei näyttele suunnitteluprosessissa kovin suurta roolia. (Saariluoma ym. 2010, 84–85.)

Muoto voidaan kuitenkin ymmärtää myös laajemmin. Saariluoman ym. (2010, 94) mukaan hyvälle muodolle on olemassa monenlaisia kriteereitä, joista mainitaan usein kauneus, kestävyys ja käyttökelpoisuus. Muotoilun yksi iso haaste on suhteuttaa suunnittelupäätökset tieteelliseen ajatteluun. Päätösten tulisi perustua tieteelliseen tietoon ihmisestä ja inhimillisestä kokemisesta enemmän kuin yksittäisen muotoilijan kokemuksille siitä, mitä ihminen on ja miten hän kokee.

Yleinen muodon kokemiseen liittyvän näkemyksen ymmärtäminen on suunnittelutyössä tärkeämpää kuin yksittäisten kokemusten analysointi. Vaikka muodolla on olemassa tiettyjä ominaisuuksia, kuten mittasuhteet, paino ja väri, ihmiset tulkitsevat muotoja subjektiivisesti, suhteessa aikaisempiin kokemuksiinsa. Alkuperäiset kokemukset voivat muovata kokemusta muodosta niin negatiivisesti kuin positiivisestikin. (Saariluoma ym. 2010, 94–95.)

Muodon kokeminen on hyvin monitahoinen tutkimusongelma, jota voi lähestyä useasta eri viitekehuksesta. Yhtä lailla estetiikka, semiotiikka kuin insinööritieteetkin tarjoavat työkaluja muotoiluun liittyvien ongelmien ratkaisemiseksi. Muodon ja käyttäjäkokemuksen suhdetta tarkasteltaessa on hyvä huomioida, että muotoa arvioidaan niin käytettävyyden kuin sen esteettisten ja symbolisten ominaisuuksien perusteella. (Saariluoma ym. 2010, 97.)

Käytettävyyden näkökulmasta arvioituna muodossa keskeistä on muun muassa helppo opittavuus, selkeä toiminnallisuus ja suhteellisen yksinkertainen design. Esteettisiä ominaisuuksia ovat puolestaan kauneus ja harmonisuus sekä dynaamisuus. Muodon symboliset ominaisuudet puolestaan liittyvät siihen, että muoto on sopusoinnussa persoonallisuutemme kanssa. Tutkimukset ovat osoittaneet, että käyttöliittymän esteettisyydellä ja symbolismilla on jopa käytettävyyttä painavampi rooli tuotetta valittaessa. (Saariluoma ym. 2010, 97.)

2.7 Ohjelmistoprojektin dokumentointi

Usein ohjelmistoprojektiin liittyy jatkuvia tukitoimia, joita hallinnoidaan projektin taustalla, koko projektin alusta loppuun saakka. Tällaisia tukitoimia ovat muun muassa projektinhallinta, vaatimustenhallinta, muutostenhallinta, testaus ja dokumentointi. (Haikala & Mikkonen 2011, 30.)

Sovellusta suunniteltaessa koodin kirjoittamisen on tärkeä vaihe, sillä seuraavat vaiheet ovat siitä riippuvaisia, tämä korostuu etenkin ohjelmiston ylläpitovaiheessa. Koodin ylläpitoa, luettavuutta ja uudelleen käytettävyyttä kannattaa pohtia jo koodia alun perin kirjoitettaessa. Ohjelmistokoodin luettavuutta voidaankin pitää merkittävänä tekijänä määrittäessä koko ohjelmiston laatua. (Eleyan ym. 2020, 1–3.)

Ohjelmiston ylläpidossa aikaa vievin vaihe on Eleyan ym. (2020, 1–3) tutkimusten mukaan alkuperäisen lähdekoodin lukeminen ja ymmärtäminen. Koodin kommentointi on koodin ohessa olevaa dokumentaatiota koodin toiminnallisuuksista. Huolellisesti kommentoitu koodi parantaa koko ohjelmistokoodin laatua. Kommentoinnin olemassaolo korostuu etenkin silloin, kun ylläpitovaihetta hoitaa joku muu kuin ohjelmiston kehittäjä.

Koodin kommentoinnissa täytyy kuitenkin muistaa, että sillä ei voida korvata koodin puutteita. Jos koodaus ei ole riittävän laadukasta on koodi korjattava, ei riitä, että koodin puutteet avataan kommenteissa. On osoitettu, että huonot kommentit tekevät koodin lukemisesta hankalampaa kuin kokonaan puuttuvat kommentit. Koodin kommentoinnissa olisi hyvä huomioida, että kommenttien laatu on määrää tärkeämpää, kommentoinnin tulee olla selkeää ja johdonmukaista, ja olisi hyvä, että koodin funktiot alkaisivat niiden toimintaa selittävällä kommentilla. (Eleyan ym. 2020, 4.)

2.8 Ohjelmiston toiminnallisuuden arviointi

Ohjelmiston käytettävyyttä arvioidaan ohjelmistoprojektin edetessä, jotta kehitystyötä pystytään ohjaamaan kohti käyttäjäystävällisempää lopputulosta. Sovelluksen toimivuutta ja käytettävyyttä voidaan arvioida muun muassa vaikuttavuuden, tehokkuuden, tyytyväisyyden näkökulmista. (Carter 2005, 149.)

Ohjelmiston vaikuttavuutta on se, miten hyvin ohjelmisto tekee sille annetut tehtävät. Jotta, ohjelmistosta saadaan loppukäyttäjilleen tehokas, on ohjelmistosuunnittelijoiden täytynyt tunnistaa mahdollisimman kattavasti kaikki erilaiset käyttäjäryhmät ja käyttötilanteet, joita varten ohjelmistoa on kehitetty. (Carter 2005, 149.)

Ohjelmiston tehokkuutta arvioitaessa, pyritään suhteuttamaan ohjelman kehitykseen käytetyt resurssit saavutettuun lopputulokseen. Kehitystyössä on valittava huolellisesti sellaiset menetelmät, joilla lopputuloksesta saadaan mahdollisimman käyttäjäystävällinen. (Carter 2005, 149.)

Carterin (2005, 149) mukaan tyytyväisyyttä arvioitaessa pyritään mittaamaan positiivisia asenteita ohjelmiston käyttöä kohtaan sekä epämieluisuuden tunteen puuttumista ohjelmisto käytettäessä. Parhaiten tyytyväisyyden tavoitteessa voidaan onnistua, kun ymmärretään, ettei parasta mahdollista lopputulosta saavuteta yhdellä kertaa. Kehitystyön kannattaa edetä asteittain, kehittämällä sovellusta edellisestä versioista saadun palautteen perusteella.

3 Käytettävyys ja käyttökokemus

3.1 Hyvän käytettävyyden periaatteet

Käytettävyys on ihmisen kykyä käyttää jotakin tuotetta tai järjestelmää helposti ja tehokkaasti. Kun suunnittelua tehdään käytettävyys huomioiden, tulee järjestelmien käyttämisestä tehokkaampaa ja vähemmän stressaavaa. Tuotteen hyödyllisyys on sen kykyä suorittaa ohjelmistolla aiotut toimenpiteet. Sovelluksen helppokäyttöisyys puolestaan vastaa sitä, kuinka hyvin käyttäjä pystyy käyttämään sen toimintoja. (Soares ym. 2021, 5–6.)

Jotta voidaan kehittää helppokäyttöinen tuote tai palvelu, on välttämätöntä omaksua käyttäjälähtöiset suunnittelukäytänteet osaksi ohjelmiston kehitystyötä. Käyttöliittymää on pystyttävä muokkaamaan ja tarkentamaan iteratiivisesti yhdistäen suunnitteluvaiheita ja käytettävyysarviointeja. Käyttökelpoisen tuotteen tai palvelun käyttöliittymän kehittäminen vaatii vuorovaikutuksen perusluonteen ymmärtämistä. Tyypillinen vuorovaikutus käyttöliittymän kanssa koostuu käyttäjän tekemistä havainnoista, joiden pohjalta käyttäjä tekee päätöksiä ja lopulta toimii tekemiensä päätösten mukaisesti. Nämä toiminnat voidaan tunnistaa havainto-, kognitio- ja motorisiksi toiminnoiksi. (Keates 2018, 96.)

Yleisin tapa käsitellä ihmisen ja teknologian suhdetta on Saariluoman ym. (2010, 40) mukaan asettaa ihminen käyttäjän rooliin. Näin siitäkkin huolimatta, että teknologiasuunnittelussa on pitkään ollut vallalla älykkään vuorovaikutuksen visio, jossa ihmisen rooli käyttäjänä hämärtyy ja muuttuu osallisuudeksi vuorovaikutustapahatumassa.

Perinteisesti teknologian 'käyttäminen' on ollut perustana suunnitteluajattelulle. Käyttäjälähtöisyyttä korostavassa suunnittelussa (*Human-Centered Design – HCD*) suunnittelu perustuu käyttäjän tarpeiden kartoittamiseen ja käyttäjävaatimusten määrittelyyn. Kun käyttäjät ovat mukana tuotteiden suunnittelussa alusta saakka, tuote vastaa niin laadullisesti kuin toiminnallisesti käyttäjien tarpeisiin, sovelluksesta tulee mukava käyttää ja sen käytettävyysongelmat pystytään minimoimaan tai parhaassa tapauksessa jopa poistamaan kokonaan. Käyttäjälähtöistä suunnittelua leimaa interaktiivinen tuotekehitysprosessi, jossa tuotteen eri toteutusvaiheita arvioidaan käyttäjien avulla tai heidän kanssaan. (Saariluoma ym. 2010, 40.)

Käytettävyyttä arvioitaessa mitataan sitä, kuinka hyvin käyttäjä pystyy saavuttamaan sovelluksella tavoittelemansa lopputuloksen. Usein sekoitetaan termit käytettävyys, käyttökokemus ja helppokäyttöisyys. Käytettävyys on osa käyttäjäkokemuksen (*UX – User Experience*) suunnittelua. Käytettävyys on tavallaan käyttäjäkokemuksen toinen taso. Tärkeää hyvässä käytettävyydessä on se, että käyttäjä pystyy ensimmäisestä käyttökerrasta

alkaen löytämään sovelluksesta tarvitsemansa ratkaisut riittävän helposti. Sovelluksen käytettävyys ja hyödyllisyys ratkaisevat yhdessä sovelluksen käyttökelpoisuuden. Ei ole väliä, jos jonkin asian tekeminen sovelluksella olisi helppoa, jos sovellus ei tee sitä, mitä sen haluttaisiin tekevän. Sovellus on hyödytön myös silloin, jos se periaatteessa pystyy tekemään kaiken tarvittavan, mutta tehtävien toteutus ei onnistu koska käyttöliittymä käyttäminen on liian vaikeaa. (Nielsen 2012.)

Kehitystyössä käytettävyyttä voi arvioida esimerkiksi hyödyntämällä Nielsenin (2020) käyttöliittymäsuunnittelun heuristiikkoja. Se on asiantuntijoille kehitetty lista, jota on käytetty käytettävyyden arviointiin ja parantamiseen usean vuosikymmenen ajan. Lista on avuksi kehitettäessä käyttäjätavallisia ja intuitiivisia digitaalisia ohjelmistoja. Alun perin 1990-luvun alkupuolella kehitytty heuristiikka listaus on päivitetty 2020-luvulla tukemaanmaan paremmin nykyaikaista käyttöliittymäsuunnittelua.

Nielsenin (2020) periaatteet ovat seuraavat:

1. Käyttöliittymässä on selkeä ulkonäkö ja sen elementit noudattavat totuttuja lainalaisuuksia: Käyttöliittymässä on kerrallaan esillä vain tarpeellinen sisältö ja visuaalinen ilme auttaa korostamaan tärkeimpiä elementtejä. Käyttöliittymässä vältetään turhaa koristelua, ja visuaalisilla ratkaisuilla on perusteltu tarkoitus. Käyttäjää tiedotetaan visuaalisin vihjein siitä, mitä sovelluksessa on tapahtumassa.
2. Informaatio tuodaan käyttäjän saataville loogisessa järjestyksessä: Käyttöliittymän kieli ja toimintalogiikka on arkista, ja muistuttaa ihmisten välistä kommunikointia. Vältetään erikoistermien käyttöä.
3. Vähäinen muistitavakka ohjelmiston käyttäjälle: Oikeat toiminnot pystyy päättelemään ennemmin kuin opetelemaan ulkoa. Käyttäjän tarvitsema tieto on helposti saatavissa. Mahdollisuus myös peruuttaa, jos tekee virhepäinalluksen, ilman että tarvitsee aloittaa koko prosessia alusta
4. Johdonmukaisuus rakenteessa ja viestinnässä: Sovelluksen toiminnot, viestintä ja visuaaliset elementit mukailevat yleisesti käytössä olevia periaatteita sekä totuttuja toimintatapoja, näin käyttökokemuksesta saadaan intuitiivinen.
5. Virhetilanteiden hallinta, selkeät virheviestit: Ensiarvoista on kuitenkin huolellinen suunnittelu, jolla yritetään ehkäistä virhetilanteiden syntymisen
6. Intuitiiviset toiminnot: Minimoidaan käyttäjän muistin kuormitusta pitäen elementit näkyvillä. Käyttäjän ei tarvitse muistaa yhden toiminnon sisältöjä sovelluksen toisessa osassa. Valikkojen tulisi olla käyttöliittymässä näkyvillä tai ainakin helposti saatavilla.

7. Käytön joustavuus ja ohjeistus: Käyttäjälle tarjotaan mahdollisuuksia mukauttaa soveluksen ominaisuuksia tai käyttää pikakomentoja omien tarpeidensa mukaan. Näin käyttö on tehokasta kaiken tasoisille käyttäjille. Prosesseja on mahdollista toteuttaa eri tavoin, niin käyttäjä voi valita tavan, joka sopii hänen työskentelylleen parhaiten.

8. Minimalistinen design: Käyttöliittymän ei pitäisi sisältää sellaista tietoa, jota tarvitaan harvoin. Jokainen käyttöliittymän harvemmin käytetty komponentti vie huomiota tärkeämmiltä komponenteilta. Tämä ei kuitenkaan tarkoita, että käyttöliittymän olisi suunniteltava latteaksi, vaan varmistettava, että visuaaliset elementit tukevat käyttäjän toimintaa parhaalla mahdollisella tavalla.

9. Autetaan käyttäjiä tunnistamaan ja ratkaisemaan virhetilanteet: Virheviestien tulee sisältää selkeitä toimintaohjeita epämääräisten virhekoodien sijaan. Virhe ilmaistaan selkeästi ja annetaan ehdotus virheen korjaamiseksi.

10. Apu ja ohjeet helposti saatavilla: Ohjeiden tehdään selkeiksi ja helposti löydettäviksi. Parasta olisi, että ohjelmisto ei tarvitsisi erillisiä ohjeita, usein dokumentaatio kuitenkin helpottaa käyttäjiä saavuttamaan tavoitteensa.

Yksi suurimmista virheistä mitä sovellusta kehitettäessä voi tehdä on se, ettei käytettävyyttä ja käyttäjäkokemusta aleta suunnitella heti projektin alusta lähtien. Käyttäjänäkökulman huomioiminen on sovellukselle elintärkeää jo senkin takia, että erilaiset käyttäjäryhmät tarvitsevat sovellukselta erilaisia asioita. Käytettävyyssuunnittelu määrittelee milaista koodia sovellus tulee tarvitsemaan, eikä vain sitä miltä sovellus ulkoisesti näyttää. (Platt 2016, 10.)

Pienimmissäkin sovelluksissa on huomioitava käyttäjäkokemus. Vaikka kyseessä olisi ohjelmallinen verkkopalvelu, siinäkin tarvitaan virhetilanneraportointia, tilanne ja toimintalokien seuraamista ynnä muuta. Voisi jopa ajatella niin, että kun sovelluksessa on vain vähän käyttöliittymäominaisuuksia, on entistäkin tärkeämpää, että ne pienet osiot ovat hyvin toimivia ja huolellisesti suunniteltu tukemaan käyttäjiensä tavoitteita. (Platt 2016, 11.)

3.2 Vuorovaikutussuunnittelu

Saariluoma ym. (2010, 115–116) kuvaavat suunnittelua jättelua prosessina, jossa on yleensä tarkoituksenmukaista edetä tiettyjen vaiheiden kautta kohti lopullista päämäärää. Suunnittelu jaetaan vaiheisiin ja osaongelmiin. Ihmisen ja teknologian vuorovaikutussuunnittelu on haasteellista, sillä ei ole mahdollista toteuttaa teknologioita ratkaisematta käyttä-

jäongelmia. Koska käyttäjä on aina olennainen osa ihmistä palvelevaa tekniikkaa, ovat käyttäjän toimintoihin liittyvät toimivat ratkaisut välttämättömiä.

Suunnitteluprosessi voidaan jaotella erilaisiin tyypeihin, joilla on oma merkityksensä prosessin eri vaiheissa. Karkeasti jako voidaan tehdä kahteen tyyppiin, jotka ovat divergentti eli hajautuva, vapaasti assosioiva ajattelu sekä konvergentti eli tiivistyvä ja keskittyvä ajattelu. (Saariluoma ym. 2010, 112.)

Divergentti suunnitteluajattelu tarkoittaa ajattelua, jossa ideoidaan vapaasti, ja pyritään tuottamaan mahdollisimman paljon vaihtoehtoja, ilman että huolehditaan liikaa erilaisista reunaehdoista. Useat luovuusmenetelmät pohjaavat divergenttiin ajatteluun. Usein tällaisella ajattelulla on suunnitteluprosessia edistävä ja vapauttava merkitys. (Saariluoma ym. 2010, 113.)

Konvergentti ajattelu pyrkii määrittämään ongelman mahdollisimman tarkasti. Usein ongelma pyritään muotoilemaan uudella tavalla, jotta suunnitteluprosessissa olevista pysähdyksistä päästään eteenpäin. Luonnonlait, materiaalit, voimat ja kustannukset määrittävät reunaehdot suunnitteluajattelulle. Näiden ehtojen puitteissa tuote on pakko saada toimimaan. (Saariluoma ym. 2010, 113.)

Divergoiva ja konvergoiva suunnittelu eivät ole toistensa vastakohtia tai toisiaan poissulkevia ajattelutapoja. Jokainen ongelman uudelleen jäsentely merkitsee samalla uuden idean luomista, siksi olisi tärkeää hallita kumpikin luovan ajattelun perusmuoto ja nähdä niiden roolit suunnitteluprosessissa. (Saariluoma ym. 2010, 114–115.)

Kuvassa 1 on esitetty vuorovaikutussuunnittelun prosessimalli. Vuorovaikutusprosessin ytimen muodostaa suunnitelma käyttäjätiedon hallinnasta. Suunnittelua voidaan kehittää muun muassa käyttäjätutkimuksella, ideoinnilla, luonnostelulla ja prototyypin kehittämällä. Suunnitteluprosessin vaiheisiin ei pidä suhtautua liian kaavamaisesti. Kuvattu kehä ei tarkoita sitä, että eri vaiheet tulee aina toteuttaa juuri kyseisessä järjestyksessä tai että suunnitelma on valmis, kun kehä on kertaalleen kierretty. Suunnitteluprosessi on iteratiivinen, ja on tarkoituksenmukaista edetä suunniteltavan sovelluksen ehdoilla. (Saariluoma ym. 2010, 117.)



Kuva 1. Vuorovaikutussuunnitteluprosessi (mukailtu Saariluoma ym. 2010, 118)

Käyttäjätiedon tehokkaan hallinnan tulisi olla keskeinen osa prosessia. Käyttäjätietoa erilaisissa muodoissa voivat sisältää esimerkiksi tutkimukset, käytettävyydestausraportit, ihmisen ja teknologian vuorovaikutusta käsittelevät artikkelit, asiakaspalauteraportit tai markkinatutkimusraportit. Lopulta riippuu tietenkin aina kehitettävästä tuotteesta tai palvelusta, millainen käyttäjätieto missäkin tapauksessa on suunnittelun kannalta oleellista. Yksittäistä tuotekehitysprosessia laajemmin ajateltuna olisi hyvä, että prosessin aikana tuotettu käyttäjätieto tallentuisi organisaation hyödynnettäväksi myös seuraavissa projekteissa. Tällainen käyttäjätietopääoman kasvattaminen mahdollistaisi tehokkaan suunnitteluprosessin myös suunnittelutyöryhmien vaihtuessa. (Saariluoma ym. 2010, 117–119.)

Periaatteessa käytettävyydeltään hyvän ohjelmiston suunnitteleminen on yksinkertaista ja suunnittelutyössä käytettävät menetelmät ovat loogisia. Ensin selvitetään, millaisia ovat palvelun käyttäjät, mitä heidän pitää saada sovelluksella tehdyksi ja miten ja missä ympäristössä he toimivat. Sitten tätä taustaa vasten tehdään tuote, jota jatkuvasti testataan. Suunnitteluratkaisut perustuvat tarvekartoitukseen ja aukikirjoitettuihin liiketoiminnallisiin tavoitteisiin. Tämä on käyttäjäkeskeisen suunnittelun perusidea, joka on kuvattu ISO 13407 standardissa ja joka toistuu kaikissa käyttäjäkeskeisissä menetelmissä. (Sinkkonen ym. 2009, 33.)

3.3 Käyttäjäkokemus

Platt (2026, 1–2) korostaa hyvän käyttöliittymän suunnittelun vaativan kehittäjiltään uudenlaista ajattelua. Nykyisin ohjelmistojen kilpailuetu tehdään ennen kaikkea huolellisella käytettävyyssuunnittelulla ja sovelluksen intuitiivisesti toimivalla käyttöliittymällä. Huonolla käyttöliittymällä olevat sovellukset eivät menesty, eikä laitteet, joiden toiminnasta ne vastaavat käy kaupaksi. Hyvä käyttöliittymä ei ole vain ekstraa muutoin toimivan tuotteen tai laitteen päälle vaan välttämättömyys koko sovelluksen olemassaololle.

Käyttäjäkokemusta ja käyttöliittymää suunniteltaessa on perehdyttävä siihen, mitä käyttäjäkokemus on ja mitä se ei ole. Käyttöliittymää ei pidä missään nimessä tyypistää pelkiksi sovelluksen fontti ja värivalinnoiksi. Käyttäjäkokemuksen suunnittelun viimeinen ja vähiten tärkeä osa ovat kauniisti muotoillut painikkeet ja pyöristetyt kulmat. (Platt 2016, 2.)

Ohjelmiston hyvä käyttäjäkokemus rakennetaan yleensä paljon ennen, kun sovelluksen käyttöliittymää aletaan niin sanotusti koristella. Käyttäjäkokemuksen perusta luodaan käyttäjämääräyksen pohjalta. Suunnittelussa on aloitettava siitä, mitä käyttäjä todella tarvitsee ja miten sovellus voisi parhaalla mahdollisella tavalla tyydyttää käyttäjän tarpeet. Vastata kun käyttäjäkokemus vastaa käyttäjän tarpeita voidaan siirtyä tekemään käyttöliittymästä miellyttävän näköistä. (Platt 2016, 3–7.)

Huomionarvoista käyttäjäkokemuksen suunnittelussa on Plattin (2016, 3–6) mukaan se, että käyttäjäkokemussuunnittelu ei saa perustua arvauksiin käyttäjien mieltymyksistä, vaan tutkittuun tietoon. Isoissa sovelluksissa on tunnistettava sovelluksen erilaiset käyttäjäryhmät ja vastattava näiden kaikkien tarpeisiin.

Suunnittelijan on hyvä huomioida käyttäjätarpeita kartoittaessa se, että usein käyttäjät ovat huonoja kertomaan todellisista tarpeistaan. Parasta on, jos käyttäjäkokemuksen suunnittelija pääsee tarkkailemaan käyttäjiä näiden omassa toimintaympäristössä. Samoin käyttökokemusta on myös testattava siinä ympäristössä, jossa sovellusta on tarkoitus käyttää. Esimerkiksi tarjoilijoiden tilausten kirjaamiseen käytettävää sovellusta on turha testata rauhallisessa toimistossa, kun sovellusta pitää pystyä käyttämään nimenomaan hektisessä ja meluisissa ravintoloissa. (Platt 2016, 3–7.)

3.4 Käyttäjystävällisyys

Ohjelmistoa suunniteltaessa käyttäjystävälliseksi on kiinnitettävä huomiota siihen, miten ohjelman erilaiset toiminnot on aseteltu. Elementtien sijoittelu, muoto, koko ja väri vaikuttavat siihen mitä toiminnallisuuksia käyttäjät olettavat elementeillä olevan. Kun ohjelmisto suunnitellaan siten, että elementtien suunnittelu huomioi ihmisen luontaisen tavan suun-

nata huomioitaan ja tehdä oletuksia, käyttökokemuksesta saadaan positiivisempi. Hyvin suunnitellulla ohjelmistolla käyttäjien on helpompi saavuttaa tavoitteensa, ja tarve turvautua käyttäjätukeen tai käyttöoppaisiin pienenee. (Evans 2017, 39–40.)

Evansin (2017, 40–42) mukaan käyttäjät olettavat alitajuisesti ohjelmiston elementtien toiminnallisuuksien liittyvän toisiinsa muun muassa silloin, jos elementit sijaitsevat lähellä toisiaan tai jos elementit muistuttavat toisiaan värinsä tai muotonsa puolesta. Jos käyttöliittymäsuunnittelussa huomioidaan nämä seikat, pystyvät käyttäjät ymmärtämään ohjelmiston eri elementtien käyttötarkoituksen ilman pohdintaa tai erillistä ohjeistusta.

Käyttäjäystävällisen sovelluksen kehitystyön tueksi tietoa käyttäjien tarpeista voi kerätä monella tavalla. Kun tietoa kerätään suoraan käyttäjiltä, heidän on helpoin kuvailla tarpeitaan, kun heillä on mahdollisuus konkreettisesti näyttää, millaisissa tilanteissa ohjelmistoa tullaan käyttämään, ja millaisia ongelmia sillä pyritään ratkaisemaan. Myös sovelluksen suunnittelijan on helpompi muistaa käyttäjien kertomia yksityiskohtia, kun haastattelun lisäksi pääsee näkemään käytännön esimerkkejä. (Kujala 2005, 116–117.)

Pelkkä käyttäjien tarpeiden kartoittaminen ei kuitenkaan riitä hyvän ohjelmiston suunnittelun pohjaksi. Kaikkia käyttäjätarpeita on mahdoton täyttää, sillä tarpeita on usein paljon, ja yksittäiset tarpeet voivat olla ristiriidassa keskenään. Ohjelmistosuunnittelijoiden on siis analysoitava, priorisoitava ja kuvailtava käyttäjätarpeet yksityiskohtaisiksi ohjelmistovaatimuksiksi. Vaatimusten huolellisen dokumentoinnin perusteella ohjelmiston kehittäjien on helpompi tehdä järkeviä päätöksiä valitessaan lopullisen ohjelmiston kannalta toimivimmat käyttäjätarpeet, joita lähdetään toteuttamaan. (Kujala 2005, 117.)

3.5 Arvokeskeinen suunnittelu

Tuotesuunnittelua ei ole mahdollista tehdä ilman siihen liittyvää arvopohdintaa. Teknologia mahdollistaa erilaisten sovellusten kautta lukuisia vaihtoehtoja toimintaympäristön kehittämiseen. Suunnittelussa kohdataan muun muassa sellaisia arvokysymyksiä, kuin kenelle ja kenen ehdoilla kehittämämme palvelu on suunniteltu käytettäväksi. Arvokeskeinen suunnittelu voidaan käsittää usealla tavalla. Sen voidaan ajatella koskevan teknologian ja inhimillisten arvojen välistä suhdetta, siis sitä miten vuorovaikutustutkimus tuo arvoa teknologian käyttäjälle. Arvosuunnittelu voi toisaalta painottaa myös rahallista arvoa, jonka teknologia tuo käyttäjälleen esimerkiksi aikasäästönä. (Saariluoma ym. 2010, 50.)

Teknologian kehittämiseen ja käyttöönottoon liittyy aina ristiriitaisia kysymyksiä, sillä suunnittelussa on kyse ihmisen toimintaa tukevan teknologian kehittämisestä. Sen takia arvojen lisäksi myös eettisten kysymysten tarkastelu on suunnittelussa väistämätöntä, niin itse tuotteen kuin suunnitteluprosessinkin kannalta. (Saariluoma ym. 2010, 51.)

Eettiset kysymykset ratkaistaan joko tietäen, mitä tehdään ja miksi tai tiedostamatta. Usein ratkaisut ajaudutaan tekemään tiedostamatta, sillä eettisten kysymysten hahmottaminen on haasteellista. Kuitenkin eettisten kysymysten huomioiminen vain epäsuorasti voi johtaa huonoihin ratkaisuihin. Kehitettäessä sovellusta tai palvelua otetaan samalla kantaa siihen, millaista tulevaisuuden maailmaa rakennetaan. Tämän takia teknologian kehityksessä tarvitaan selkeitä eettisiä näkemyksiä, joiden perusteella teknologisia valintoja voidaan tehdä. (Saariluoma ym. 2010, 51–52.)

Eettinen suunnittelu tarkoittaa Saariluoman ym. (2010, 52) mukaan ensinnäkin eettisten periaatteiden tietoista valintaa suunnittelupäämääriä ja eri suunnitteluvaihtoehtoja pohdittaessa. Toiseksi se on itse suunnitteluprosessin pohdintaa ja toteuttamista eettisestä näkökulmasta. Eettisenä kannanottona voidaankin siis pitää jo sitä, jos suunnittelua on lähdetty tekemään käyttäjälähtöisesti.

4 Työkalun suunnittelu

4.1 Toimeksianto

Tarve työkalulle huomattiin, kun yrityksessä jouduttiin usein selvittämään, löytyykö tietylle tuotteelle vastaavaa tuotetta joltain toiselta valmistajalla. Eri tuotevalmistajilla oli erilaisia listauksia, joiden avulla vastaavuuksia pystyi selvittämään. Tehohydrolla oli useita eri toimittajia ja laaja tuotevalikoima. Keskenään vertailtavia listoja oli useita, lisäksi piti selvittää, oliko kyseistä tuotetta heti saatavilla liikkeen hyllystä vai oliko tuote tilattava valmistajalta, ja millainen oli tilattavan tuotteen toimitusaika asiakkaalle.

Työkalun avulla tuotehaku prosessia oli tarkoitus osin automatisoida, ja näin säästää työhön käytettävää aikaa ja parantaa hakutulosten tarkkuutta, kun kone tekee vertailun ihmisen puolesta. Tähän asti yhden tilattavan tuotteen löytämiseen saattoi kulua jopa puolituntia työaikaa. Kun työkalun käyttöön saadaan lisättyä riittävän laaja tietokanta, tämä aika voidaan lyhentää muutamiiin minuutteihin.

Työkalu tulee käyttöön Tehohydron usealla eri osastolla. Tämän takia työkalu oli suunniteltava yksinkertaiseksi ja helppokäyttöiseksi, jotta käyttäminen onnistuu kaikilta, jotka sitä tarvitsevat. Työkalun lisäksi opinnäytetyössä tehtiin yksinkertaiset ja lyhyet ohjeet työkalun käyttämiseen tueksi. Työkalun täytyy olla toimiva myös silloin, kun käytettävään lähdeaineistoon tulee muutoksia. Työkalu suunniteltiin siten, että sitä olisi mahdollisimman helppo muokata toimimaan toimeksiantajan muuttuvien tarpeiden mukaan.

Tähän asti tuotevastaavuuksien etsintään käytetty aika oli koettu varsin turhauttavaksi. Hankala vastaavuuksien seuranta johti osin myös varaston käytön tehottomuuteen. Varastossa saattoi olla sama tuote kahteen kertaan eri tuotenimillä, kun tuotteiden vastaavuutta ei ollut tunnistettu tuotteen tilausvaiheessa.

Toimeksiantajan kanssa keskusteltaessa kävi ilmi, että vastaavalla työkalulle voisi olla yrityksessä käyttöä useammankin tuoteryhmän tuotteiden vastaavuuksia etsittäessä. Toimeksiantajalle olisi siis hyötyä, jos työkalusta pystyy suunnittelemaan sellaisen, että se toimii yhtä hyvin vaikka haussa käytettävä lähdeaineisto muuttuisikin.

4.2 Menetelmän valinta

Opinnäytetyönä kehitettävä työkalu luotiin Microsoftin Excel -ohjelmistolla, hyödyntäen Visual Basic for Applications ohjelmointikieltä. Tämä toteutustapa valittiin sen takia, että Excel oli jo ennestään ollut Tehohydrolla käytössä tuotevastaavuustietojen tallentamises-

sa ja etsinnässä. Näin käyttöön otossa ei tarvinnut huolehtia uuden ohjelmiston asentamisesta.

Työkalu olisi ollut mahdollista toteuttaa myös esimerkiksi sovelluksena, jota varten korvaavuustiedot olisi tallennettu erilliseen, tätä tarkoitusta varten suunniteltuun tietokantaan. Erilliseen tietokantaan luetut tuotetiedot olisi ollut mahdollista strukturoida sovelluksella helposti haettavaan muotoon. Tällöin tietokantaan tehtävistä hauista saatavat hakutulokset olisivat voineet olla paremmin etukäteen ennakoitavissa. Ulkoisen tietokannan haasteeksi olisi kuitenkin muodostunut tietokannan päivitettävyyttä.

Yksittäisten vastaavuustietojen lisääminen tällaisella sovelluksella olisi ollut mahdollista toteuttaa samalla periaatteella, kuin nyt toteutetussa työkalussa. Ongelmaksi olisi voinut muodostua suurempien tietomäärien lisääminen käytettävään tietokantaan. Isojen tiedostojen rakenteen muokkaus tietokantaan sopivaksi vaatisi osaamista ja ymmärrystä tietokantojen ylläpidosta.

Nyt työkalu mahdollistaa isojenkin kokonaisuuksien lisäämisen työkalun käyttöön hyvin helposti. Korvaavuustaulukoita sisältävän Excel-tiedoston pystyy lisäämään työkalun käyttöön, kunhan tarkistaa, että tiedoston muotoilu on oikeanlainen.

Lisäksi tuotetietojen ylläpito on Excel-tiedostoja käytettäessä yksinkertaista. Jos tietokannasta huomataan virheellistä tietoa, on virheellinen tieto mahdollista käydä poistamassa tai muokkaamassa tiedostolta.

Vanhentuneen tiedon poistamisessa tässä työkalussa ei tarvitse juuri huolehtia, sillä tietokannassa on järkevää säilyttää vanhojakin tuotekorvaavuustietoja. Monesti vanhentuneella tuotekoodilla täytyy löytää saatavilla oleva nykyinen korvaava tuote. Näin on esimerkiksi silloin, kun asiakkaan vanha tuotetta ei ole enää myynnissä. Tällöin on hyvä, että hakutyökalun tietokannasta löytyisi myös vanha tuotenumero ja sitä vastaavat nykyiset korvaavat tuotteet.

Monesti uuden ohjelman käytön opettelu vie aikaa. Nyt kun toteutus näyttää tutulta Exceliltä, on työkalun käyttö työntekijöille helppo aloittaa. Käyttöön otosta oli pyrittävä tekemään työntekijöille mahdollisimman helppoa, sillä kun mahdollisimman moni työntekijä hyödyntää työssään työkalua, sen tuomista aikasäästöistä ja työn laadun parantumisesta saadaan suurin mahdollinen hyöty yrityksessä.

Excel makroilla tehtävää toteutusta puolsi myös se, että VBA ohjelmointi on helposti opittavaa, ja sitä hyödyntämällä on mahdollista, että Tehohydrolla pystytään jatkossa tarpeen vaatiessa päivittämään työkalua. Päivittämismahdollisuudella pyritään varmistamaan, että työkalu todella tulee käyttöön ja että se pysyy käytössä pitkän aikaa.

4.3 Vaatimukset työkalulle

Työkalun toimintojen suunnittelu alkoi vaatimusmäärittelyllä. Toiminnalliset vaatimukset määriteltiin tiiviissä yhteistyössä toimeksiantajan kanssa. Toiminnalliset vaatimukset määrittivät sen, mitä työkalun tulee tehdä. Näiden lisäksi toteutukseen vaikuttivat erilaiset tekniset yksityiskohdat, jotka liittyivät esimerkiksi työkalun rakenteeseen.

Excelin laskentataulukkosovelluksen voi toteuttaa siten, että kaikki työkalun käyttämä materiaali kootaan yhteen Excel-tiedostoon esimerkiksi eri välilehdille. Toinen vaihtoehto on pitää tuotetietoaineistot omissa tiedostoissaan, ja työkalun avulla käydään nämä olemassa olevat tiedostot läpi. Tämä työkalu toteutettiin siten, että käyttöliittymä on omassa Excel-tiedostossaan ja sen käyttämä lähdeaineisto on omissa tiedostoissaan. Näin vähennetään riskiä, että käyttäjä poistaisi vahingossa työkalusta joitain sen käyttämiä tietoja.

Työkalu toteutettiin siten, että olemassa olevaan aineistoon tarvitsi tehdä mahdollisimman pieniä muutoksia. Tällä pyrittiin siihen, että jatkossa uusien tuotteiden vastaavuustaulukoiden lisääminen työkalun käyttämään tietokantaan olisi mahdollisimman yksinkertaista.

Kuvassa 2 näkee, kuinka työkalun ohjeissa esitetään uusien tiedostojen lisääminen. Uuden tiedoston lisääminen tietokantaan onnistuu varmistamalla, että tiedosto noudattaa työkalun muotoiluvaatimuksia. Tämän jälkeen riittää, että tiedosto siirretään siihen kansioon, jota työkalu käyttää korvaavuustietojen etsimiseen.

Lisää kokonainen Excel taulukko

Työkalun käyttämään kansioon voi viedä uusia ristilistauksia Excel-tilaukkoina. Muotoile uudet Excel-tiedostot siten, että keskenään korvaavat tuotteet löytyvät aina samalta riviltä, valmistajia voi olla useita vierekkäin. Alla esimerkki rakenteesta, jollaista työkalu osaa lukea.

SAP NRO	NIMITYS	Korvaava	Korvaava		
310000	1.11 000000N4	000000 FKT8H	.	.	.
310005	000000 BN	7500	.	.	.
310000	0020 BN	7400	.	.	.
120000	000000 03 BN4	960000	.	.	.

Jos uudessa taulukossa on välissä tyjiä soluja, ne kannattaa korvata pisteellä. Korvaus työkalun saa Excelistä auki painamalla 'Ctrl + F' panikkeita yhtäaikaan. Etsittävä -kohta jätetään tyhjäksi ja korvaava -kohtaan merkitään piste (.)

Etsi ja korvaa

Etsi Korvaa

Etsittävä: Muotoilua ei ole määritetty Muoto...

Korvaava: Muotoilua ei ole määritetty Muoto...

Kohteesta: Työkirja Sama kirjainkoko

Etsi: Riveittäin Koko solun sisältö

Käy läpi: Kaavat Asetukset <<

Korvaa kaikki Korvaa Etsi kaikki Etsi seuraava Sulje

Kuva 2. Ohje Excel-tiedoston lisäämiseen

Työhön liittyvät vaatimukset koottiin listaksi, jonka avulla pystyy seuraamaan työhön liittyvien vaatimusten määrää sekä arvioita aikataulusta, mikä yksittäisen ominaisuuden kehittämiseen menee. Vaatimusluettelo toimi samalla ikään kuin projektin työlistana, josta oli mahdollista seurata kehitystyön aikataulua ja etenemistä. Vaatimukselle oli perusteltua määrittää myös tärkeysluokitus.

Tärkeyskriteerillä voitiin määrittää sitä, kuinka tärkeä jokin ominaisuus oli saada valmiiksi työkalun kokonaisuuden kannalta. Hyvä oli miettiä etukäteen myös sitä, mitkä ominaisuudet työkalussa ovat toisistaan riippuvaisia. Näin toteutusta voitiin suunnitella etenemään tärkeimmistä toiminnallisuuksista kohti vähemmän tärkeitä, huomioiden kuitenkin se, että keskinäisten riippuvuuksien takia oli ajoittain tärkeä kehittää jokin kokonaisuus loppuun ennen seuraavaan vaiheeseen siirtymistä.

Kuvassa 3 näkyy osa tätä työkalua varten tehdystä vaatimusmäärittelylistasta. Listan avulla pystyi seuramaan mitä osioita työkalusta oli saatu valmiiksi, ja mitkä olivat kehitteillä. Lisäksi tärkeysjärjestyksen asettaminen helpotti ratkaisemaan sen, mitä toimintoa lähdetiin kehittämään seuraavaksi.

Tunnus (ID)	Vaatus	Tärkeys	Kommentit	Työlistalla	Kehitteillä	Testauksessa	Valmis	kommentit
001	Haku löytää vastaavat tuotekoodit annetusta aineistosta	1					x	
002	PDF aineisto muunnetaan excel- taulukoksi	1					x	
003	Osittainen hakusana riittää haun tekemiseen	1					x	
004	Haku toimii myös ilman erikoismerkkejä	1				x		
005	Haku toimii vaikka tuotekoodissa olevia välilyöntejä ei syötetä haku kenttään	1			x			
006	Käyttöliittymässä kenttä hakusanalle	1				x		
007	Käyttöliittymässä sarakkeet yleisimmin käytetyille valmistajille	1					x	
008	Työkalulla tieto missä kansiossa olevia exceleitä sen pitää käydä läpi	1	Ensimmäisellä käynnistyksellä kysyy sijainnin, joka tallennetaan muistiin jatkoa varten		x			

Kuva 3. Vaatimusluettelo

Vaatimusluettelon mukaan eteneminen helpottaa myös työn rajaamista. Jos työkalun valmiiksi saaminen osoittautuisi liian vaativaksi tehtäväksi annetussa ajassa, pystytään vaatimuslistan perusteella pohtimaan, olisiko lopullisesta työkalusta hyvä jättää joitain vähemmän tärkeitä ominaisuuksia pois, jotta kokonaisuus saadaan valmistumaan halutussa aikataulussa. Vaatimuslistasta sai parhaan hyödyn projektin seurantaan, kun sitä päivitti jatkuvasti kehitystyön edetessä. Kesken projektin löytyi myös joitain uusia ominaisuuksia,

joiden mukaan ottaminen tekisi työkalusta hyödyllisemmän toimeksiantajalle. Näissä tilanteissa vaatimusluetteloon lisättiin uusia vaatimuksia kesken kehitystyön.

4.4 Toiminnallisuuksien suunnittelu

Tarkempi kuva työkalun tarpeellisista ominaisuuksista selvitettiin toimeksiantajan edustajien kanssa keskustelemalla. Yhteistyössä toimeksiantajan kanssa kartoitettiin mahdollisimman tarkasti toimeksiantajan eri työntekijöiden tarpeet ja odotukset toteutettavalle työkalulle. Työnetenemistä seurattiin verkko- ja lähitapaamisia, joissa pystyttiin selvittämään, että työ eteni kohti haluttua lopputulosta.

Työkaluun toivottuja ominaisuuksia käytiin läpi usean työntekijän kanssa, niin sähköpostilla kuin kasvokkainkin keskustelemalla. Näin toimimalla pystyttiin selvittämään miten tarpeet vaihtelevat esimerkiksi yrityksen eri osastojen välillä. Työkalulle oli selkeä tarve yrityksen useissa eri työtehtävissä.

Työkalun suunnitteluvaiheessa oli ratkaistava myös se mistä kaikesta aineistosta haku olisi pystyttävä tekemään, ja miten toimitaan, jos lähdeaineistoon tulee muutoksia. Työkalun olisi hyvä pystyä toimimaan luotettavasti myös sen jälkeen, kun sen käyttöön lisätään uusia korvaavuustiedostoja. Kehitystyön yhteydessä työkaluun tehtiin lyhyet ohjeet työkalun käyttöönottoon, hakujen tekemiseen ja siihen, miten työkalun käyttöön pystytään lisäämään uusia tuotevastaavuuksien ristitaulukoita. Tarkoitus kuitenkin oli, että työkalun peruskäyttö onnistuu intuitiivisesti myös ilman ohjeiden lukemista.

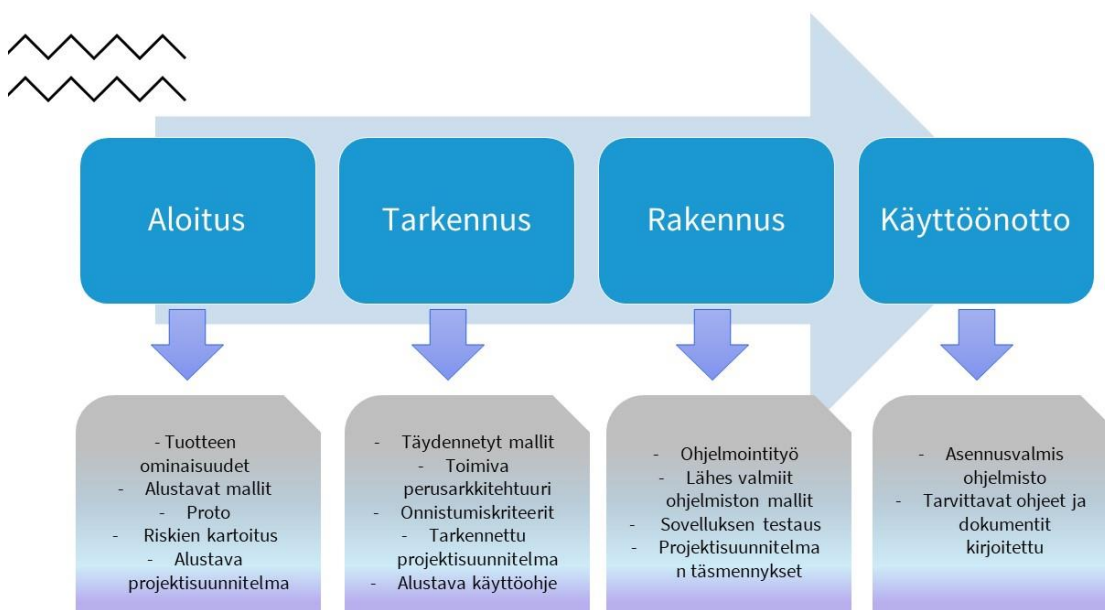
Tietokantahaun lisäksi työkaluun tehtiin myös verkkohakutoiminto, jolla haetaan tuotetietoa kahden eri valmistajan verkkosivuilta löytyvien korvaavuus hakukoneiden kautta. Näin työkalun käyttöön saatiin laajempi ja jatkuvasti päivittyvä tuotetietojen valikoima toisin kuin pelkkien olemassa olevien listojen kautta olisi ollut käytettävissä. Verkkohaun ongelmana on kuitenkin se, jos tuotevalmistajan verkkosivustolla tapahtuu muutoksia, ei työkalu ehkä pysty hakemaan tietoa muuttuneilta verkkosivuilta. Loppukäyttäjän on tällaisissa tilanteissa hankala varmistua siitä, tuoko verkkohaku luotettavasti tiedot nettisivulta annetun hakusanan perusteella.

Työkalu voisi myös pystyä tekemään annettuun lähdeaineistoon tarvitsemiaan muutoksia tiedoston rakenteeseen. Riskinä tällaisessa ratkaisussa on kuitenkin se, ettei kaikkia lähdeaineiston rakenteita pysty etukäteen ennakoimaan, eikä ohjelmistokoodissa tehtävä tiedoston rakenteen asettelu näin ollen olisi riittävää. Toiseksi lähdeaineiston rakenteen ohjelmistollinen tarkistaminen olisi käytännössä tehtävä jokaisen tiedonhaun yhteydessä, ja tämä hidastaisi olennaisesti työkalun toimintaa. Näitten syitten takia päädyttiin etukäteen määrittämään muotovaatimukset hauissa käytettäville Excel-taulukoille.

5 Työkalun toteutus

5.1 Työn vaiheet

Työkalun suunnittelu eteni vaiheittain, osin nämä kehitysvaiheet olivat keskenään päällekkäisiä, mutta vaiheiden voi nähdä silti kulkeneen tietyssä järjestyksessä. Alla olevassa kuvassa 4 on kuvattu työskentelystä tunnistetut neljä vaihetta, jotka olivat projektin aloitus-, tarkennus-, rakennus- ja käyttöönottovaiheet.



Kuva 2. Kehitysvaiheet

Aloitusvaiheessa korostui yhteistyö Tehohydron edustajien kanssa. Yhteistyökumppanin kanssa kartoitettiin heidän tarpeitaan toteutettavalle työkalulle. Esiinnousseiden tarpeiden perusteella luotiin työkalun toiminnallisuuksista alustavia malleja ja projektille suunnitelmaa, johon koottiin muun muassa aikataulu ja vaatimusmäärittelyä. Projektin alkuvaiheessa yhteistyökumppani esitteli toimitiloissaan niitä erilaisia tilanteita ja työtehtäviä, joissa työkalun käyttöä olisi tarkoitus hyödyntää.

Tarkennusvaiheessa työkalun määrittelyä tarkennettiin ja toiminnallisuuksia koottiin valmiimpaan vaiheeseen. Tässä vaiheessa käytettävät menetelmät oli lyöty lukkoon ja työkalun rakennetta tarkennettiin yhteistyökumppanin havaintojen ja toiveiden pohjalta. Tarkennusvaiheessa työkalun ominaisuuksia asetettiin yksityiskohtaisemmin tärkeysjärjestykseen, jotta rakennusvaiheessa näkisi mistä toiminnallisuuksista rakentaminen kannatti

aloittaa. Tässä vaiheessa pyrittiin tunnistamaan mahdollisimman hyvin työkalun eri osioiden keskinäisiä riippuvuuksia. Näin havainnollistettiin miten eri osat vaikuttavat toisiinsa, ja mitä osioita on järkevä kehittää yhtäaikaisesti.

Rakennusvaiheessa työkalun koodattiin mahdollisimman valmiiksi. Tehtyä koodia testattiin aina tehtyjen muutosten yhteydessä ja näin varmistettiin, että lopputuloksesta muodostuu toimiva kokonaisuus. Tässäkin vaiheessa oli tärkeä pitää yhteyttä yhteistyökumppanin kanssa, jotta työkalusta saatiin mahdollisimman onnistunut. Rakennusvaiheessa ratkaistiin millä metodeilla työkalu oli tehokkain toteuttaa. Testauksella pyrittiin siihen, että ajonaikaiset virheet saadaan kiinni ilman, että loppukäyttäjät joutuvat niistä huolehtimaan. Tilanteissa, joissa virhetietoja joudutaan tuomaan käyttäjän ratkaistavaksi, rakennettiin virheilmoitukset selkeiksi. Ne suunniteltiin siten, että käyttäjälle annetaan toimintaohjeet, miten hän pystyy ratkaisemaan virheen ja jatkamaan työskentelyään.

Rakennusvaiheessa kehitettävä työkalun toiminta esiteltiin toimeksiantajalle, ja toimitettiin muutamille työntekijöille testikäyttöön. Testikäytön avulla pystyttiin löytämään ohjelmistokoodiin jääneitä virheitä sekä korjaamaan niitä. Testaus auttoi löytämään käyttöliittymän toiminnallisuuksiin liittyviä puutteita, jotka voitiin saadun käyttäjäpalautteen perusteella korjata.

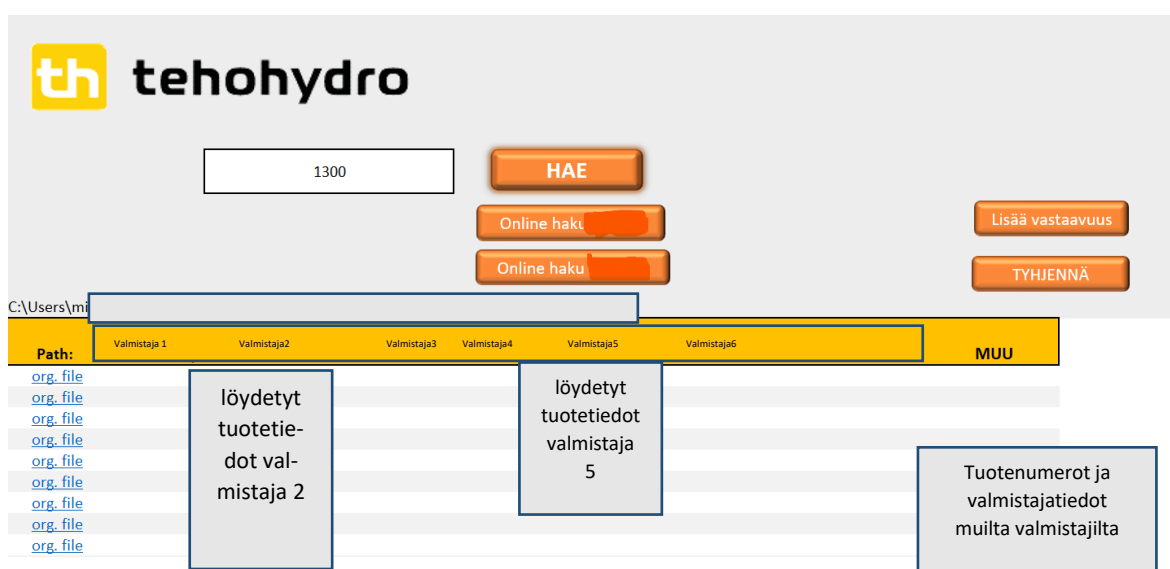
Käyttöönottovaiheessa työkalu on saatu valmiiksi ja kaikki sen komponentit on testattu. Työkalu on asennusvalmis ja sen tarvitsemat ohjeet ja muut dokumentit on kirjoitettu valmiiksi. Käyttöön otettua työkalua ei enää käytetä yksittäisen työntekijän omalta kovalevyiltä vaan verkkolevyiltä. Jatkossa työkalulla tallennettavat uudet tuotevastaavuustiedot tulevat tallentumaan yhteiseen tietokantaan.

5.2 Työkalun rakenne

Työkalun rakenteen toivottiin olevan selkeä visuaaliselta ilmeeltään. Käyttöliittymässä olisi hyvä olla näkyvillä Tehohydron logo, lisäksi toiveena oli, että useimmin käytettävien valmistajien tuotteet olisivat helposti löydettävissä. Tärkeimmät komponentit työkalussa ovat hakutoiminto ja taulukko, jossa tehdyn haun tulokset tuodaan käyttäjän näkyville.

Hakukenttä sijoitettiin työkalun avausnäkyymään, laskentataulukon yläosaan. Hakukenttä on selkeästi erotettu sovelluksen taustasta. Annetun hakusanan perusteella työkalu etsii sille määritellystä kansiossa kaikki Excel-tilit ja poimii näistä taulukoista hakusanaa vastaavat tuotetiedot työkalun päänäkyymän tulostaulukkoon.

Kuvassa 5 näkyy elementtien sijoittumisen käyttöliittymään. Hakusana kenttä on sijoitettu keskeiselle paikalla hakukenttää samoin kuin haku painikkeet. HAE-painike tekee vastaavuuksia haun käyttäjän antaman hakusanan perusteella paikalliseen Excel-taulukoista koostuvaan tietokantaan. Kahdelle eri valmistajan verkkosivulle on omat hakupainikkeensa. Verkkohaut ja haku paikalliseen tietokantaan on haluttu pitää toisistaan erillisinä toimintoina, sillä jos verkkosivujen rakenteisiin tulee suuria muutoksia, on käytännössä lähes mahdotonta taata, että työkalu toimii normaalisti. Kun jokainen haku on jaettu oman painikkeen taakse, voidaan yksittäistä *online*-hakua päivittää tarpeen vaatiessa tai poistaa siihen liittyvä hakupainike työkalusta kokonaan.



Kuva 3. Käyttöliittymä

Alkuperäiset tuotelistauksista PDF-listat muunnettiin Excelin *OCR (optical character recognition)* toiminnolla Excel-taulukoksi. Kaikki Excel listaukset muotoiltiin siten, että samalta riviltä löytyy vain keskenään vastaavia tuotenumeroita. Eri valmistajien keskenään korvaavia tuotteita voidaan sijoittaa monta vierekkäin ja eri tuotemerkejä voi sijoittaa samaan sarakkeeseen useita allekkain.

Tähän muotoiluun päädyttiin siitä syystä, että toimeksiantajan käytössä olevista listauksista valtaosa noudatti suurin piirtein tällaista muotoilua jo valmiiksi. Tarkoituksena on, että toimeksiantajan omat työntekijät pystyvät jatkossa itse lisäämään uusia tiedostoja työkalun käyttöön.

Hakutulosten etsinnän nopeuttamiseksi, hakutyökalu etsii vastaavuustietoja vain Excel-tiedoston arvoja sisältävistä soluista. Tämän takia haussa käytettävissä tiedostoissa ha-

kualueen sisällä olevat tyhjät solut täytettiin pisteellä. Näin varmistetaan, ettei haku pysähdy, vaikka alkuperäisessä tiedostossa olisikin ollut välissä tyhjiä rivejä. Samalla hakutoiminto pysyy tehokkaana, kun työkalun ei tarvitse käydä Excel-tiedoston kaikkia soluja läpi.

Paikallisesta tietokannasta vastaavuustietoja hakiessa, käydään läpi kaikki Excel-tiedostot, jotka on tallennettu työkalun käyttämään aineisto kansioon. Jokaisesta Excel-tiedostosta luetaan tiedot kaikilta sen sisältämillä välilehdiltä.

Kuvan 6 koodileike havainnollistaa, kuinka etsittäville välilehdiltä dataa sisältävien solujen arvot kootaan taulukkomuuttujaan. Taulukkomuuttujan arvot pystytään käymään läpi, ja etsimän niistä vastaavuudet annetulle hakusanalle. Hakusanaa vastaavat tuotenumerot löytyvät taulukosta samalta riviltä, mutta eri sarakkeista. Kokonaiset vastaavuus rivit tallennetaan toiseen taulukkomuuttujaan, joka myöhemmin tulostetaan käyttäjän näkyville. Kuvissa 6–8 vihreällä kirjoitetut tekstit ovat koodin kommentointia, jonka on tarkoitus avata koodin lukijalle sitä, mitä missäkin kohtaa ohjelmistokoodia tapahtuu. Kommentoimalla kaikki koodin funktiot pyritään siihen, että toimeksiantajalla on halutessaan mahdollisuus jatkossa kehittää tai ylläpitää työkalun toiminnallisuuksia.

```
For Each WS In wbSlave.Worksheets
    'Ignore the search worksheet
    If (WS.Name <> "Search" And WS.Name <> "Data") Then
        'get the range values into a variable that can be looped through
        dataArray = WS.Range(WS.Cells(dataRowStart, dataColumnStart), WS.Cells(WS.Cells(Rows.Count, dataColumnStart).End(xlUp).Row, dataColumnEnd))
        'Increase size of array that will hold the data to display to its max possible size
        ReDim datatoShowArray(1 To UBound(dataArray, 1), 1 To UBound(dataArray, 2))
```

Kuva 4. Haku usealta välilehdeltä

Kuvan 7 koodi pätkästä näkee kuinka makro käy läpi järjestyksessä etsittävältä alueelta sekä rivit että sarakkeet. Yksittäisen solun arvoa verrataan käyttäjän antamaan hakuarvoon, 'searchValue'. Käyttäjän hakuarvo on määritetty siten, että työkalu tulkitsee tuotteen nimen vastaavaksi, oli se kirjoitettu isoilla tai pienillä kirjaimilla. Lisäksi haussa jätetään huomiotta tyhjät välilyönnit tuotenumeroissa. Näiden lisäksi asteriksi merkillä (*) ohjelmoidaan makro etsimään vastaavuuksia käyttäen osittaista hakuarvoa.

```
'Loop columns
For searchfield = 1 To UBound(dataArray, 2)

    'Loop through the rows in the data range
    For i = 1 To UBound(dataArray, 1)

        ' ***** Accepting partial search and ignoring whitespace in searchterm *****
        If Replace(dataArray(i, searchfield), " ", "") Like searchValue & "*" Then
```

Kuva 5. Haku sarakkeittain ja riveittäin osittaisella hakuarvolla

Kun vastaavuus tiedot on löydetty lähdeaineistosta, ne pitää tuoda käyttäjän nähtäville. Tulostus käyttöliittymään tehdään tuotteen valmistajan nimen perusteella omien sarakkeiden alle. Yleisimmin käytettyjen valmistajien tuotteille varataan käyttöliittymästä oma sarakkeensa, jonka alle tuotetieto tuodaan. Viereisiin sarakkeisiin tulostetaan tiedot toisten valmistajien vastaavista tuotteista. Jos joltain valmistajalta ei löydy haetulle tuotteelle vastaavuustietoa, jätetään tämä sarake tyhjäksi. Yksi sarake on varattu muille kuin yleisimmin käytetyille valmistajille. Muut-valmistajat sarakkeeseen tulostetaan löydetty tuotenumero ja sen viereiseen laskentataulukon soluun kirjataan valmistajan nimi.

Tuotevastaavuustietojen lisäksi käyttöliittymään tuodaan linkki, josta klikkaamalla pääsee siihen alkuperäiseen tiedostoon, josta vastaavuustieto on löytynyt. Koska osa alkuperäistiedostoista sisältää tuhansittain rivejä, on linkki rakennettu viemään käyttäjän oikeassa tiedostossa oikean rivin kohdalle. Tämän linkityksen avulla käyttäjä pääsee varmistumaan, että työkalu löytää oikeita tietoja. Monesti alkuperäistiedostolla toisiaan läheisesti muistuttavat tuotteet on listattu lähekkäin. Linkin avulla käyttäjä voi pystyä löytämään riittävän läheisiä tuotevastaavuuksia, joiden tuotenumerot eivät kuitenkaan ole samoja ja työkalu ei siksi ole niitä tunnistanut.

Kuvan 8 koodipätkä havainnollistaa kuinka tietoja tulostettaessa työkalu ensin etsii seuraavan tyhjän rivin, jolle tuotetietoja voidaan alkaa tuoda. Tämän jälkeen tuotetietoja tuodaan oikeiden valmistajien sarakkeiden alle. Työkalu tunnistaa muut sarakkeeseen tulevat tuotetiedot ja näiden kohdalle on tuotekoodin lisäksi tulostettava valmistajan nimi. Tuotetietojen tulostuksen jälkeen tulostetaan linkit alkuperäisiin tiedostoihin.

```
'Find next empty row in the search-sheet
nextRow = search.Cells.Find(What:="", After:=Range("A1"), LookAt:=xlPart, LookIn:=xlFormulas, SearchOrder:=xlByRows, SearchDirection:=xlPrevious, MatchCase:=False)

'put data into the search-sheet
'matching product numbers
search.Range(search.Cells(nextRow, searchDataColumnStart), search.Cells(nextRow + UBound(datatoShowArray, 1) - 1, searchDataColumnStart + dataColumnWidth)).Value = datatoShowArray
'other manufacturers
For i = 1 To 150
  If manufacturerInfo(i) <> Empty Then
    search.Cells(nextRow, 11).Value = manufacturerInfo(i)
  End If
  nextRow = nextRow + 1
Next

'Links to original files
On Error Resume Next
For i = 1 To UBound(addressfound, 1)
  If addressfound(i) <> Empty Then
    addlinkRow = search.Cells(Rows.Count, 1).End(xlUp).Row + 1
    wbmaster.Sheets("search").Hyperlinks.Add Anchor:=wbmaster.Sheets("search").Range("A" & addlinkRow).Offset(a, 0), Address:="" & addressfound(i), SubAddress:=addressfound(i)
  End If
Next
```

Kuva 6. tietueiden tulostus käyttöliittymään

Toimeksiantajan on tärkeä päästä myös lisäämään tunnistamiaan tuotevastaavuuksia paikallisiin tiedostoihinsa työkalun avulla. Kun tietoja haetaan valmistajien nettisivujen kautta, on käyttäjän mahdollista tallentaa haun löytämät vastaavuudet rivi kerrallaan paikalliseen tiedostoon. Tallennus painike tuodaan jokaisen haussa löydetyn tuoterivin kohdalle, jotta tallentaminen on käyttäjälle mahdollisimman vaivatonta. Alla olevassa kuvassa

9 näkyy taulukon rakenne, jossa tuotetietojen oikeaan reunaan tuodaan *Copy*-painike, jota painamalla kyseisen rivin tiedot viedään talteen paikalliseen tietokantaan. *Copy*-painikkeen painamisen jälkeen kopioidun rivin taustaväri muutetaan keltaiseksi, jotta käyttäjä näkee selkeästi mitkä rivit hän on tallentanut. Ponnahdus ikkunassa tuodaan myös tieto tallennuksen onnistumisesta tai siitä, jos kyseinen tieto löytyy muistista jo ennestään.

manuf part no	Identification	part no	
12			Copy
11			Copy
77			Copy
77			Copy
12			Copy
12			Copy
125			Copy

Kuva 7. painikkeet tietojen tallennukseen

Yhteistyökumppanilla voi tulla vastaan myös muualla tunnistettuja vastaavuustietoja, jotka halutaan saada talteen samaan työkalun käyttämän tietokantaan kuin muutkin tuotevastaavuustiedot. Tällaiseen tietojen tallennusta varten työkaluun on luotu käyttäjälomake. Lomakkeen avulla käyttäjä pystyy valitsemaan tai kirjoittamaan tuotteen valmistajan tiedot ja tuotenumeron. Jos tuotteelle on tiedossa enemmän kuin yksi vastaavuustieto, pystyy käyttäjä avaamaan lomakkeeseen lisää rivejä, joilla voidaan viedä useamman valmistajan tuotetietoja talteen.

Kuva 10 näyttää tallennuslomakkeen. Kun tiedot tallennetaan työkalu tarkistaa ensin, onko sama vastaavuustieto jo tallennettu aikaisemmin tietokantaan. Jos tuotetieto löytyy jo lisätynä tietokannasta, siitä ilmoitetaan käyttäjälle ponnahdusikkunassa. Myös onnistuneesta tiedon tallennuksesta tiedotetaan käyttäjää, avaamalla ponnahdusikkuna, jossa kerrotaan tallennuksen onnistuneen.

Syötä tieto tuotevastaavuudesta

Valitse Tuotenumero

Vastaavan suodattimen tiedot

Valitse Tuotenumero

Lisää rivi

Kuva 8. Vastaavuuksien tallennus lomakkeella

5.3 Käytettävyyden arviointi

Työkalun käyttökokemus suunniteltiin ja toteutettiin noudattaen hyvän käyttöliittymäsuunnittelun periaatteita. Ulkonäkö pidettiin yksinkertaisena ja selkeänä. Varimaailma valikoitui yrityksen värimaailmaa hyödyntäen. Käyttöliittymässä on kerralla esillä vain tarpeellisimmat ominaisuudet ja kaikki toiminnot on pyritty nimeämään yksiselitteisesti. Virhetilanteet on pyritty välttämään huomioimalla ne koodissa, jottei loppukäyttäjän tarvitsisi murehtia virheiden selvityksestä työkalua käyttäessään.

Käytön tueksi on luotu lyhyet ohjeet. Ohjeissa neuvotaan, miten työkalu pystytään ottamaan käyttöön ja miten työkalu etsii käyttämänsä lähdeaineiston. Ohjeissa neuvotaan myös, miten työkalun käyttöön voi lisätä uusia Excel-tiedostoja, ja millaiseksi lisättävät on muotoiltava, jotta työkalu osaa lukea niitä toivotulla tavalla.

Työkalulla saatiin vain muutaman viikon mittainen koekäyttö ennen lopullisen työkalun palautusta. Työkalun esittelyn ja koekäytön aikana nousi esiin jotain toiveita lisä toiminnallisuuksista sekä pieniä korjausehdotuksia työkalun toimintaan. Muutostöissä priorisoitiin virheiden ja puutteiden korjaus uusien ominaisuuksien edelle. Uusina ominaisuuksina työkaluun tuotiin kaksi painiketta, joiden takaa löytyy linkkejä sivustoille, joilta käyttäjä voi löytää lisää ristitaulukointeja tuotetiedoista.

Testikäytön perusteella on näyttänyt siltä, että tuotehakuihin käytettävää aikaa pystytään lyhentämään merkittävästi. Aikasäästöä saatiin aikaiseksi niiden tuotemerkkien kohdalla, joiden listauksia löytyy tietokannasta riittävän kattavasti. Huomattavaa on kuitenkin, että joidenkin käytettävien tuotemerkkien osalta tietokanta ei vielä ole riittävä parhaan mahdollisen hyödyn saamiseksi.

Koekäyttöönotto sujui työkalussa olevien ohjeiden avulla. Koekäytössä työkalu toimi oletetulla tavalla ja sen avulla pystyi löytämään tuotteille korvaavuustietoja. Hyväksi koettiin, että hakusanaksi riittää osittainen tuotekoodi, joka löytää tuotetiedot myös ilman, että hakusanaan kirjataan tuotekoodissa olevia erikoismerkkejä. Koekäytöstä saadun palautteen perusteella hakutulosten esittämistapaan tehtiin pieniä muutoksia. Tehtyjen muutosten jälkeen lopullinen työkalu on valmis käyttöönotettavaksi.

6 Yhteenveto ja pohdinta

Tässä opinnäytetyössä käytettiin teoriana tietoa VBA ohjelmoinnista ja sovelluksien kehittämisestä Microsoft Excel-ohjelmistolla. Excelin ja VBA:n käyttö mahdollistaa nopealla tuotekehitysaikataululla räätälöidyn ohjelmiston toteuttamisen. Excel toimii erinomaisesti juuri laajojen tietomäärien käsittelyssä ja esittämisessä. Excelin omat sisäänrakennetut funktiot mahdollistavat jo itsessään monipuolisen tietojenkäsittelyn, kun työkalun toimintoja lisäksi ohjataan tähän tarkoitukseen kehitettyjen makrojen avulla, saadaan olemassa oleva tieto muokattua kohtuullisen helposti vastaamaan yhteistyökumppanin tarpeita.

Lisäksi teoriatietona käsitellään ohjelmiston suunnittelua käyttäjälähtöisesti. Erilaisten käyttäjäryhmien huomiointi on kaikessa tuotesuunnittelussa ensiarvoisen tärkeää. Osa sovelluksen käytön mielekkyydestä syntyy toimintojen intuitiivisesta sijoittelusta. Käyttäjät huomioivan toteutuksen ansiosta työkalua pystytään käyttämään ilman ohjeisiin perehtymistä.

Mikään työkalun toiminto ei olisi mahdoton toteuttaa itsenäisesti ilman työkalun käyttöä. Työkalun arvo on siinä, että sen avulla on pystytty yhdistämään useampi työvaihe yhden painalluksen taakse. Lisäksi samalta laskentataulukolta pystytään hallitsemaan useampia erillisiä toimintoja samasta paikasta, kun tietoa voidaan hakea sekä yrityksen omista tietovarastoista, että valmistajien tarjoamilta nettisivuilta. Työkalun avulla löydetty tieto tulee selkeästi esille, eikä useita eri tiedostaja tarvitse pitää koneella samanaikaisesti avoimena. Tämä on omiaan helpottamaan työskentelyä, kun aiemmin useaan eri näkymään jakautunut tieto saadaan nyt koottua yhdelle sivulle.

Räätälöidyn pienohjelmiston käyttö voi tehdä työnteosta työntekijälle mielekkäämpää ja mahdollistaa työajan käyttöä oleellisempien työtehtävien tekemiseen. Ihmisen työajasta on hyvä vapauttaa niin paljon kuin mahdollista päättely- ja asiakaspalvelutyöhön ja muihin sellaisiin tehtäviin, joita koneet ja automatisaatio eivät vielä osaa tehdä.

Työkalua esiteltäessä yhteistyökumppanille nousi esiin muutamia jatkokehitysehdotuksia. Työkalun toivottiin lukevan tietoja useammilta verkkosivuilta kuin siihen nyt lisättyjen kahden valmistajan verkkopalveluista luettavat tuotetiedot. Uusien verkkohakujen tuominen työkaluun ei ollut mahdollista tämän opinnäytetyön aikataulussa. Nyt kehitetty työkalu on suunniteltu etsimään korvaavuustietoja vain yhden kategorian tuotteista, työkalun voisi muokata kuitenkin toimimaan myös muille tuoteryhmille.

Muissa tuoteryhmissä työkalun käyttöönottoon riittäisi parhaimmillaan käyttöliittymän päivittäminen siten, että siihen tuotaisiin paikat uuden kategorian yleisimmille tuotemerkeille.

Samoin ohjelmistokoodia olisi päivitettävä, siten että tulokset tunnistetaan ja lajitellaan uuden kategorian tuotemerkkien mukaan.

Työkalua voisi jatkossa kehittää sellaiseksi, että sen avulla voitaisiin kerätä tietoa siitä toistuvatko jotkut tietyt hakusanat usein ohjelmaa käytettäessä. Jos jotkin tietyt tuotteet osoittautuvat sellaisiksi, että korvaavuuksia niille etsitään toistuvasti, voisi työkalu ehdottaa, että kyseisiä tuotteita olisi jatkuvasti ostettuna varastoon. Muutoinkin olisi luontevaa jatkokehittää tätä työkalua siihen suuntaan, että se tietäisi myös varastotilanteen ja osaisi löytämiensä tuotetietojen lisäksi kertoa, mitkä löydetyistä tuotteista löytyvät jo valmiiksi varasosta ja kuinka monta kappaletta niitä varastossa on.

Lähteet

Carter, J. 2005. Transforming usability engineering requirements into software engineering specifications: from PUF to UML. Seffah, A., Gulliksen, J. & Desmarais, M. Teoksessa Human-Centered Software Engineering – Integrating Usability in the Software Development Lifecycle. Springer, 147-169.

Eleyan, D., Othman, A. & Eleyan, A. 2020. Enhancing Software Comments Readability Using Flesch Reading Ease Score. Viitattu 5.2.2023. Saatavissa <https://www.mdpi.com/2078-2489/11/9/430>.

Evans, D. 2017. Bottlenecks: Aligning UX Design with User Psychology. Kenmore, Washington: Apress.

Haikala, I. & Mikkonen, T. 2011. Ohjelmistotuotannon käytännöt. Helsinki: Talentum Media Oy.

Hänninen, P. 2022. Robotiikka ja tekoäly. Tammertekniikka / Amk-Kustannus Oy.

Juvonen, R. 2018. Ohjelmistoprojektin sudenkuopat ja miten ne vältetään. Helsinki: BoD – Books on Demand.

Keates, S. 2018. Designing for Accessibility: A Business Guide to Countering Design Exclusions. CRC Press.

Kujala, S. 2005. Linking User Need and Use Case-driven Requirements Engineering. Toim. Seffah, A., Gulliksen, J. & Desmarais, M. Teoksessa Human-Centered Software Engineering – Integrating Usability in the Software Development Lifecycle. Springer, 113–125.

Nielsen, J. 2012. Usability 101: Introduction to Usability. Viitattu 28.2.2023. Saatavissa www.nngroup.com/articles/usability-101-introduction-to-usability/

Nielsen, J. 2020. 10 Usability Heuristics for User Interface Design. Viitattu 12.2.2023. Saatavissa www.nngroup.com/articles/ten-usability-heuristics/.

Platt, D. 2016. The Joy of UX. Pearson Education, Inc.

Saariluoma, P., Kujala, T., Kuuva, S., Kymäläinen, T., Leikas, J., Liikkanen, L. & Oulasvirta, A. 2010. Ihminen ja teknologia. Hyvän vuorovaikutuksen suunnittelu. Helsinki: Teknologiainfo Teknova Oy.

Sinkkonen, I., Nuutila, E. & Törmä, S. 2009. Helppokäyttöisen verkkopalvelun suunnittelu. Helsinki: Tietosanoma Oy.

Soares, M., Rebelo, F. & Ahram, T. 2021. Handbook of Usability and User-Experience: Methods and Techniques. Taylor & Francis Group.

Tehohydro. Viitattu 24.1.2023. Saatavissa <https://www.tehohydro.fi/palvelut/>.

Walkenbach, J. 2013. Excel 2013 Power Programming with VBA. Viitattu 28.1.2023. Saatavissa <http://ebookcentral.proquest.com/lib/lab-ebooks/detail.action?docID=1144767>.