

Logistiikkaohjelmiston kehittäminen museossa

Satu Haapakoski



20.2.2021

Tekijä(t) Haapakoski Satu	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Opinnäytetyön otsikko Logistiikkaohjelmiston kehittäminen museossa	Sivu- ja liitesivumäärä 50 + 40
Opinnäytetyön otsikko englanniksi Developing logistics software in museum environment	
Museovirastossa on käytössä Muskatti-niminen esinelogistiikkajärjestelmä. Järjestelmällä hallitaan kokoelma esineiden paikkatietoja ja liikkeitä. Järjestelmän rikkoutuminen ja siellä olevan tiedon katoaminen olisi merkittävä menetys organisaatiolle. Järjestelmän kehittäjä on virkavapaalla ja sen viimeistely on jäänyt tekemättä. Järjestelmässä olevat puutteet pitäisi kuitenkin saada korjattua ja saattaa järjestelmä ylläpitovaiheeseen. Kehitystyön hankinta on haastavaa, koska organisaatiosta puuttuu henkilö, joka ymmärtäisi sekä järjestelmän tekniikkaa että siihen liittyvää työtä. Tämän työn on tarkoitus yhdistää molempien puolien tietämystä. Tarkoitus on kartoittaa tärkeimmät kehityskohteet ja muodostaa niiden perusteella lista korjattavista asioista. Kehitystarpeita oli kerätty jo ennen tämän työn alkua. Mutta niiden käsittely selkeäksi ja priorisoiduksi listaksi oli tarpeellista ennen tarjouksen pyytämistä. Kehityskohteiden kartoitus aloitettiin kartoittamalla järjestelmän merkityksellisimmät käyttäjäryhmät. Näille käyttäjille tehtiin kysely, jonka avulla pohdittiin muutostarpeiden priorisointia. Osa muutostoveista oli hieman epäselviä. Näiden toimintojen kohdalla tehtiin käyttäjätestaus, haastatteluja ja tarkempia määrittelyjä. Opinnäytetyön lopputuotoksena muodostettiin priorisoitu lista kehitystarpeista. Tämä tuotos toimii vaatimusmäärittelynä tietojärjestelmän kehittäjälle sekä hankinnan ohje- nuorana tietohallinnolle. Lisäksi pohdittiin järjestelmän tulevaisuutta ja mahdollista ylläpidon ulkoistamista.	
Asiasanat Ohjelmiston kehittäminen, vaatimus määrittely, käyttäjäkokemus, käyttäjätutkimus	

Sisällys

1	Johdanto	1
2	Käsiteluettelo	4
3	Käyttäjäkokemus ja sen kehittäminen	5
3.1	Käyttäjäkokemuksen merkitys	5
3.2	Käyttöliittymän visuaalinen ulkoasu	6
3.3	Käyttöliittymän toiminta	7
3.4	Käyttöliittymän tietoturva	8
3.5	Käyttöliittymän toimintaympäristön vaikutus käyttäjäkokemukseen	9
3.6	Käyttäjäkokemuksen tutkiminen	10
4	Muutoshallinta	13
4.1	Digitaalinen muutos	14
5	Kehitettävä järjestelmä	16
5.1	Muskatti-järjestelmä	16
6	Käyttäjäryhmät	20
7	Muskatin tilanne ennen kehitystä	22
8	Kehitystarpeiden kartoitus	26
8.1	Kehitystarpeiden alustava kartoitus	26
8.2	Kysely	28
8.3	Käyttäjätestaus	30
8.4	Haastattelu tilastotoiminnosta	32
8.5	Selvitys kuvien siirtymisestä kahden järjestelmän välillä	33
8.6	Selvitys asiakaslaskutukseen liittyvän toiminnon korjauksesta	35
9	Muskatin kehityssuunnitelma	37
9.1	Priorisointi	38
9.2	Muutoksien toteutus	40
9.3	Ylläpito	43
10	Johtopäätökset	46
11	Pohdinta	49
	Lähteet	51
	Liitteet	54

1 Johdanto

Työskentelen tällä hetkellä Suomen kansallismuseossa. Kansallismuseo on Museoviraston alainen museo, joka säilyttää, tutkii ja esittelee suomalaista historiaa ja siihen liittyvää esineistöä. Työskentelen yksikössä, jossa tehtävänä on huolehtia museoesineiden säilytyksestä sekä logistiikasta. Meillä on käytössämme Muskatti niminen logistiikkajärjestelmä, johon kirjaamme museoesineiden sijainnit. Käytämme järjestelmää päivittäin ja se on toiminnallemme hyvin tärkeä. Se on helppokäyttöinen ja meille tärkeä työkalu. Järjestelmän rikkoutuminen ja siellä olevan tiedon katoaminen aiheuttaisi organisaatiollemme merkittäviä taloudellisia menetyksiä sekä hidastaisi ja hankaloittaisi useita toimintoja.

Järjestelmä on tehty ratkaisuna ongelmaan. Aiemmin puolen miljoonan museoesineen sijaintitiedot olivat hajanaisesti ja niiden kirjanpidossa ei ollut yhtenäistä järjestelmää. Tämä aiheutti paljon ylimääräistä työtä ja esineiden paikantaminen oli hankalaa. Tätä helpottamaan päätettiin tehdä järjestelmä, johon paikkatiedot saadaan helposti ja missä niiden hallinnoiminen on tehokasta. Projektista innostui työntekijä, joka rupesi rakentamaan järjestelmää. Päämääränä on ollut saada järjestelmä valmiiksi, mutta sen teon yhteydessä ei kuitenkaan ole mietitty sen ylläpitoa, elinkaarta tai päivityksiä.

Tietojärjestelmän tehnyt työntekijä on nyt virkavapaalla. Museoviraston tietohallinto on hyvin pieni yksikkö ja siellä ei ole töissä ohjelmistokehittäjää. Kyseisen tietojärjestelmän ylläpitoa ja kehitystä ei tällä hetkellä pysty tekemään kukaan organisaation työntekijöistä. Järjestelmä on osittain jäänyt viimeistelemättä ja sen toiminnassa on ominaisuuksia, jotka eivät toimi halutulla tavalla. Kansallismuseon henkilökunnan työn kannalta olisi tärkeää, että järjestelmä saataisiin korjattua ja sen ylläpito olisi varmistettu. Jos järjestelmä ja kaikki siinä olevat tiedot menetetään, voivat siitä aiheutuvat kustannukset olla pahimmillaan satoja tuhansia euroja.

Olen ollut kiinnostunut järjestelmän teknisestä puolesta ja olen päässyt tutustumaan läheisesti sen toimintaan. Olen päässyt tarkastelemaan ohjelmistoa kooditasolla. Tämä on auttanut ymmärtämään sitä, miten järjestelmä toimii. Olen myös alkanut saada käsitystä siitä, mitä järjestelmällä on mahdollista tehdä ja miten sitä voi kehittää. Tästä minulle heräsi ajatus, että voisin tehdä opinnäytetyöni järjestelmään liittyen.

Organisaatio koostuu pääasiassa museotyöläisistä, joista hyvin harvoilla on kokemusta tietojärjestelmien kehittämisestä. Tämä aiheuttaa ongelmia, kun hankitaan IT-palveluita. Valtiolla on tarkat säännöt hankinnoista varsinkin, kun kyse on isoista projekteista. Ennen hankintaa pitää osata määritellä hankinnan raamit oikein. Hankinnoissa hinta on usein

yksi merkittävimpiä valintaan vaikuttavia seikkoja. IT- hankinnoissa pitäisi miettiä sitä, miten saadaan paras hyöty käytetystä rahasta, ei pelkästään sitä, mikä ratkaisu on halvin (Juvonen 2018, 146). Jos hankinnan määrittelyä ei ole tehty riittävän tarkasti, saattaa toimittaja tarjota sellaisia palveluita, jotka eivät ole tarpeeseen sopivimpia. Silloin voi lopputulos olla, että järjestelmä tai toimenpide ei ole se, mitä on haluttu. Tämän aiheuttaa riskin, että tyydyttävään lopputulokseen päästään vain hankkimalla lisätoimia, jotka taas nostavat helposti hankinnan hintaa.

Olen huomannut, että organisaatiossamme on paljon hiljaista tietoa, jota ihmiset pitävät itsestään selvänä. Tämä aiheuttaa usein tilanteita, joissa olennaisista asioista unohdetaan kertoa työkavereille. Lisäksi kyseessä olevaan järjestelmään liittyviä prosesseja ei ole kuvattu. Ne henkilöt, jotka eivät tee tätä työtä jatkuvasti eivät tiedä, miten toisten yksiköiden työ käytännössä tapahtuu. Osaa prosesseista ei ole myöskään yhteisesti sovittu ja ihmiset toimivat, miten sattuu. Nämä ilmiöt ovat aiheuttaneet sen, että järjestelmä ei aina vastaa todellisia tarpeita eikä toimi yksiin tosielämän kanssa. Järjestelmän rakentamisen aikana on ollut tilanteita, joissa järjestelmään on pyydetty muutoksia, mutta ne onkin toteutettu eri tavalla kuin pyytäjä on halunnut. Tämä johtuu siitä, että tarpeita ei ole osattu kuvailla tarpeeksi tarkasti ja kehittäjä on käsittänyt ne väärin.

Koen, että minulla on sellaista tietoa järjestelmien kehittämisestä ja käyttäjien huomioimisesta, jonka avulla pystyn kartoittamaan käyttäjien tarpeet ja kuvailemaan ne tarvittavalla tarkkuudella. Tämä tarkoittaa sitä, että minun on osattava toimia oman työroolini ulkopuolella ja katsottava tilannetta kokonaisvaltaisesti. Jotta tämä on mahdollista, on hankittava tietoa kaikista käyttäjäryhmistä ja heidän tarpeistaan.

Opinnäytetyössä haluaisin selvittää, mitkä ovat olennaisimmat muutokset, jotka järjestelmään tulisi tehdä ja miksi, ja onko muutoksia kannattavaa tehdä kyseiseen järjestelmään. Tämän lisäksi on tarkoitus määritellä kehityskohteet ja priorisoida ne hankintaa varten. Tarkoitus on luoda kehittäjälle selkeät suuntaviivat, joiden avulla muutokset on mahdollista tehdä niin kuin on tarkoitettu. Johtopäätöksissä ja pohdinnoissa on tarkoitus lyhyesti käsitellä ohjelman tulevaisuutta, tulevia kehitystarpeita sekä ohjelmiston ylläpidon tarpeita.

Opinnäytetyön lopputuotoksena on tarkoitus kuvata järjestelmän kehitykselle tällä hetkellä merkittävimmät toimenpiteet. Tämä tuotos toimii vaatimusmäärittelynä tietojärjestelmän kehittäjälle sekä hankinnan ohjenuorana tietohallinnolle. Opinnäytetyö sisältää myös kevyen järjestelmäkuvauksen, jossa painotetaan enemmän ohjelman ominaisuuksia ja käyttöä, kuin sen teknistä toteutusta.

Järjestelmä pitäisi ottaa myös laajempaan käyttöön, jotta siitä saadaan täysi hyöty. Nyt järjestelmää käytetään vain joissain yksiköissä. Järjestelmän kehittämisellä ja käyttäjäkokemuksen parantamisella voidaan helpottaa järjestelmän laajempaa käyttöönottoa. Tämä opinnäytetyö keskittyy kuitenkin merkittävimpiin tämänhetkisiin käyttäjäryhmiin ja heidän tarpeisiinsa.

2 Käsiteluettelo

CSV-tiedosto on tekstitiedosto, jossa taulukkomallinen tieto on järjestetty välimerkeillä eroteltuna. CSV-tiedostot voidaan tehdä taulukkolaskentaohjelmilla. CSV-tiedosto on helppompi käsitellä tietojärjestelmissä kuin esimerkiksi Excel-tyyppinen tiedosto.

Digitointi muistiorganisaatioissa tarkoittaa sitä, että niiden säilyttämää aineistoa dokumentoidaan digitaalisiin järjestelmiin. Käytännössä tämä tarkoittaa aineiston tietojen viemistä kokoelmienhallinta- tai arkistointijärjestelmiin. Usein esineiden tiedot kirjataan sanallisesti ja lisäksi kohteet kuvataan. Digitointi voi tarkoittaa myös arkistoaineistojen koneellista lukemista.

Finna on palvelu, jonka kautta voi hakea tietoa kirjastojen, arkistojen ja museoiden aineistoista. Finna on alun perin kehitetty kirjastoille ja arkistoille, mutta sen kehittyttyä se on laajentunut myös museoiden aineistojen jakamiseen. Finnalla on oma avoin API-rajapinta ja organisaatiot voivat tehdä omia sivuja omien Finna-aineistojensa jakamiseen ja esittelyyn.

Järjestelmän rajapinta on se järjestelmän osa, jonka kautta tuodaan tai viedään tietoa järjestelmän ulkopuolelta. Käyttäjät käyttävät käyttöliittymää ja se toimii tässä kohtaa rajapintana. Ohjelmistot voivat toimia myös toistensa kanssa rajapintojen kautta.

Kehitysympäristö on ohjelmisto, jolla kehittäjä rakentaa ohjelmiston. Se myös kääntää koodin konekieliseksi. Siihen kuuluu usein useita ohjelmia ja sen kautta voidaan luoda ohjelmiston rakenne. Kehitysympäristössä voi olla myös automaattisia testejä, joilla havaitaan virheitä ohjelmiston toiminnassa.

Metatieto on tietoa tiedosta. Metatiedoissa voidaan kuvailla tiedon tallennustyyppi, tiedon keräämiseen liittyvät seikat ja tekijät. Esimerkiksi valokuvien metatiedot sisältävät usein: kameran tyyppin, kuvan ottamisajan, valotusajan, aukon koon, tiedosto koon ja tiedostotyyppin. Lisäksi metatiedoissa voi olla tieto valokuvaajasta, kuvauspaikasta ja käytetystä linsistä.

Muskatti on Museovirastossa käytetty logistiikkajärjestelmä. Tämä työ keskittyy Muskatintin kehittämisen ympärille.

Palvelin on tietokone, jossa pyöritetään ohjelmistoja, joilla on useita samanaikaisia käyttäjiä. Niitä käytetään usein internetin välityksellä. Palvelimet voivat olla saatavilla kaikille netin käyttäjille tai niihin voidaan rajata pääsy.

3 Käyttäjäkokemus ja sen kehittäminen

Käyttäjäkokemuksen kehittäminen osana tuotteiden tai järjestelmien tuotantoa on alkanut jo 50-luvulla. Sen merkitys on kasvanut, kun tietokoneet ovat yleistyneet ja tulleet osaksi ihmisten jokapäiväistä elämää. Ensin käytettiin pääasiassa tekstiin pohjautuvia käyttöliittymiä, mutta pian graafiset käyttöliittymät korvasivat nämä. Graafiset käyttöliittymät olivat helpompia käyttää ja niiden suosio kasvoi nopeasti. Tällöin alettiin myös kiinnittää huomiota niiden käyttömukavuuteen. Internet avasi yrityksille uusia mahdollisuuksia ja monet veivät toimintaansa verkkoon. Silloin yritysten välinen kilpailu nosti käyttäjäkokemuksen huomioimisen tärkeäksi kilpailutekijäksi. Nykyään hyvän käyttäjäkokemuksen rakentaminen on yrityksille elinehto ja sen tärkeys tulee korostumaan vielä enemmän tulevaisuudessa. (Nielsen 2017.)

3.1 Käyttäjäkokemuksen merkitys

Käyttäjäkokemuksen huomiointia ei kannata ajatella pelkästään yritysten välisenä kilpailuetuna. Käyttäjäkokemus on järjestelmän käyttöön liittyvä kokonaisvaltainen kokemus. Jos ohjelmiston käyttäjäkokemus on huono se vaikuttaa usein negatiivisesti järjestelmän käyttöön: järjestelmää ei haluta käyttää, sen käyttäminen on hidasta ja vaatii usein paljon opastusta. Nämä asiat voivat johtaa siihen, että järjestelmän käyttäjien on vaikeaa ja hidasta tehdä haluamansa toiminnot. Tämä voi johtaa järjestelmän käytön välttelyyn ja sen kehitykseen käytetyt resurssit menevät hukkaan.

Käyttäjäkokemus kannattaa huomioida järjestelmien elinkaaren kaikissa kohdissa. Käyttäjäkokemusta voidaan tarkastella järjestelmien suunnitteluvaiheessa tutkimalla niiden prototyyppejä. Tällöin saadaan tietoa siitä, mihin suuntaan järjestelmiä kannattaa ruveta kehittämään. Käyttäjäkokemuksen merkitys kasvaa, kun järjestelmää valmistellaan käyttöä varten. Tällöin käyttäjäkokemusta tutkimalla voidaan löytää keskeisiä korjattavia kohtia. Järjestelmien käyttö, ympäristö ja tarpeet saattavat muuttua ajan myötä. Kehitystilanteissa käyttäjäkokemusta voidaan tutkia ja pyrkiä parantamaan, jotta järjestelmä vastaa paremmin sen hetkistä tilannetta.

Kaikki käyttäjät eivät ole samanlaisia, joten käyttäjäkokemus voi vaihdella henkilöittäin. Onkin tärkeää löytää juuri kyseisen järjestelmän olennaisimmat käyttäjäryhmät ja tarkastella tilannetta heidän kauttaan (Nielsen 1993, 27). Esimerkiksi nuori digitaaliseen maailmaan jo lapsena tottunut henkilö käyttää ohjelmistoja hyvin eri tavalla kuin vanhempi teknologiaan tottumaton käyttäjä. Tämän takia käyttäjäkokemuksen kehittämisessä on hyödyllistä testata asioita usein juuri todennäköisimmillä käyttäjäryhmillä.

Käyttäjäkokemusta tutkittaessa voidaan myös tarkastella saavutettavuutta. Saavutettavuudella tarkoitetaan sitä, että järjestelmän kehityksessä on huomioitu käyttäjien mahdollisia heikkouksia tai erityistarpeita. Saavutettavuuden ensimmäinen asia on hyvä käyttäjäkokemus. Lisäksi tulisi käyttöliittymän visuaaliseen selkeyteen kiinnittää erityistä huomiota. Saavutettavuutta parantaa myös kuvien ja videoiden tekstitys sekä niiden metatiedot. Lisäksi vielä selkeä otsikointi auttaa ruudunlukijoiden toiminnassa. Saavutettavuus on tärkeää huomioida varsinkin julkisissa tai suurelle yleisölle suunnatuissa palveluissa. (Aalto & Paavilainen 2019.)

Käyttäjäkokemus on tärkeää huomioida, koska se vaikuttaa moneen asiaan ohjelmiston käytössä. Hyvä käyttäjäkokemus on sitä, että käyttäjän on helppo ja miellyttävä suorittaa järjestelmän avulla haluamansa toiminnot. Kun käyttäjäkokemus on hyvä, johtaa se usein siihen, että järjestelmän käyttöönotto on helppoa ja nopeaa (Nielsen 1993, 30). Lisäksi järjestelmällä tehtävä työ on tehokasta. Tällöin järjestelmään käytetyistä resursseista saadaan paras hyöty.

3.2 Käyttöliittymän visuaalinen ulkoasu

Käyttäjäkokemukseen vaikuttaa järjestelmän visuaalinen ulkoasu. Käyttöliittymän tulisi olla rauhallinen ja sen näkymässä tulisi olla vain hillitty määrä tekstiä ja objekteja. Jos näkymässä on liikaa asioita, niin käyttäjiltä menee enemmän aikaa löytää haluamansa. Liiallinen liike ja rauhattomuus tekee taas sen, että käyttäjän huomio saattaa keskittyä epäolennaisiin asioihin. Tekstien tulisi olla selkeästi näkyviä ja erottua taustasta ja fonttien tulisi olla helppolukuisia. Muuten tekstiä voi olla hankala nähdä ja lukea.

Järjestelmässä olevien ikonien käyttö tulisi miettiä tarkkaan. Ikoneilla voidaan helposti viestiä käyttäjille esimerkiksi, mitä napin painallus aiheuttaa. Ikonien tulisi kuitenkin olla helposti ymmärrettäviä. Monista ikoneista on muodostunut standardeja. Yksi näistä on tallennusikoni, joka kuvaa levykettä. Vaikka levykkeet ovat poistuneet käytöstä, niin tallennusikoni tunnistetaan helposti. Erityisen varovainen tulisi olla, jos halutaan tällä tavalla vakiintunutta kuvioita käyttää erilaisen toiminnon kuvaamiseen. Ikonien selkeyttä lisää, jos niiden yhteyteen laitetaan toimintoa kuvaava teksti. (Grant 2018, 355 & 347.)

Käyttöliittymässä olevat kuvat vaikuttavat ihmisten kokemukseen. Kuvat ovat voimakkaita elementtejä, niillä voidaan tuoda järjestelmään oikeanlaista tunnelmaa sekä ohjata käyttäjiä. Ensinnäkin kuvien tulisi olla hyvälaatuisia, muuten niiden sisällöstä ei saa selvää ja ne aiheuttavat käyttäjälle tunteen huonosta laadusta. Kuvien tulisi liittyä aiheeseen, muuten ne tuntuvat turhilta ja häiritseviltä. Kuvat kiinnittävät usein käyttäjän huomion kuvaan ja sen ympäristöön. Tätä voidaan käyttää ohjaamaan käyttäjän huomio käyttöliittymän

olennaisiin elementteihin asettelemalla ne kuvien lähistölle. Ohjeistuskuvien kanssa kannattaa huomioida se, miten käyttäjät niitä katsovat. Kuvat kannattaa olla sellaisesta kuvakulmasta, miten käyttäjä tilannetta seuraisi. Tällöin ohje on helpompi omaksua. (Groom & Nomm 2015; Babich 2019.)

3.3 Käyttöliittymän toiminta

Käyttäjäkokemus voi vaihdella paljon eri henkilöillä, johtuen ihmisten välisistä eroista. On kuitenkin olemassa paljon totuttuja tapoja, joista on muodostunut standardeja. Nämä on hyvä tiedostaa, kun valitaan järjestelmässä käytettäviä symboleita sekä suunnitellaan järjestelmän toimintalogiikkaa. Tuttujen kuvioden ja toimintalogiikan käyttö helpottaa järjestelmän oppimista ja käyttöönottoa (Nielsen 1993, 227).

Hyvään käyttäjäkokemukseen kuuluu, että ohjelmisto on helppo oppia. Tämän lisäksi ohjelmiston toiminnot ovat sen verran intuitiivisia, että siihen on helppo palata pitkänkin tauon jälkeen. Ohjelmiston tulisi myös toimia tehokkaasti, niin että sillä suoritettavat toiminnot ovat nopeita tehdä. Ohjelmiston tulisi myös toimia loogisesti ja oikein, niin että käyttäjän toimet aiheuttavat mahdollisimman vähän virheitä. Käyttäjäkokemukseen vaikuttaa myös se, miten miellyttävää ohjelmistoa on käyttää. (Nielsen 1993, 26.)

Vaikka käyttäjiä pyritään miellyttämään, ei ole kuitenkaan hyvä yrittää miellyttää kaikkia. Jos kaikkien käyttäjien toiveet pyritään toteuttamaan ja heille annetaan liikaa mahdollisuuksia ja ominaisuuksia, on ohjelmisto usein monimutkainen ja hankala oppia (Nielsen 1993, 121). Käyttöliittymissä tulisi olla selkeät yksinkertaiset reitit saavuttaa haluttu lopputulos. Tämän lisäksi visuaalisuus tulisi olla rauhallinen ja näkymään kannattaisi sijoittaa vain käyttäjälle olennaisia elementtejä.

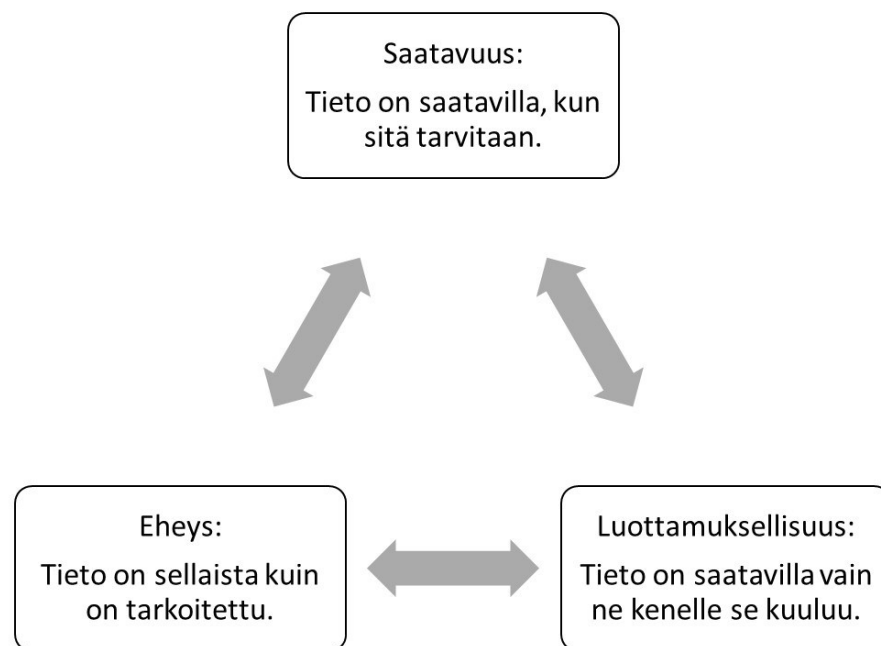
Helppokäyttöisyyteen vaikuttaa myös se, miten paljon käyttäjä pitää muistissaan asioita. Jos käyttäjä joutuu muistamaan koko ajan ison määrän asioita, vaikeuttaa se järjestelmän käyttöä. On todella raskasta, jos joutuu selaamaan usean sivun välillä, koska tarvittavia tietoja ei löydy samasta paikasta. Myös ohjeistuksen olisi olla helposti saatavilla. Järjestelmään voidaan laittaa käyttäjää ohjaavia tekstejä, jotta käyttäjän ei tarvitse muistaa kyseiseen toimintoon liittyviä yksityiskohtia ulkoa.

Käyttöliittymän käyttömukavuuteen vaikuttaa se, miten helppoa siellä on löytää haluamansa. Järjestelmän rakenne ja navigointi vaikuttavat tähän. Käyttäjän on hankala etsiä haluamiaan toimintoja, jos käyttöliittymän asetellut eroavat järjestelmän sivuilla toisistaan paljon. Järjestelmän tulisi noudattaa samaa logiikkaa koko ajan.

Käyttöliittymän tulisi toimia nopeasti ja elementtien latausaikojen tulisi olla kohtuulliset. Kuukaan ei pidä odottamisesta ja käyttäjistä on turhauttavaa odottaa käyttöliittymän elementtien latautumista tai tiedon hakua. Usein alle sekunnin aika tuntuu käyttäjistä normaalilta, mutta kymmenien sekuntien odotusaika koetaan jo häiritsevänä (Nielsen 1993, 135). Nopeuteen voi vaikuttaa sillä, että järjestelmän tietokanta on muotoiltu tehokkaasti tietojen hakua ajatellen. Lisäksi vaikutusta on koodin tehokkuudella. Internetissä saatavana olevien järjestelmien tai sivustojen nopeuteen vaikuttaa sivustolla olevien elementtien, kuten kuvien lataus, verkon nopeus ja tarvittavien koodikirjastojen latautuminen.

3.4 Käyttöliittymän tietoturva

Käyttäjäkokemukseen vaikuttaa myös se, miten paljon ohjelmistoon voi luottaa. Tietoturvallinen ohjelmisto on usein myös luotettava. Tietoturvaan kuuluu, että ohjelmisto toimii oikein silloin, kun käyttäjät sitä tarvitsevat ja siinä olevaan tietoon voi luottaa. Tämän lisäksi järjestelmiin tulisi olla rajattu pääsy ja siellä olevaan tietoon pääsevät vain ne ihmiset, kenen on kyseiseen tietoon päästävä käsiksi.



Kuvio 1. Tietoturvan osa-alueet.

Tietoturvaan kuuluu olennaisesti myös se, että järjestelmässä olevat tiedot ovat oikeanlaisia. Tähän vaikuttaa paljon se, miten paljon virheitä käyttäjän on mahdollista tehdä. Käyttäjät ovat ihmisiä ja he tekevät virheitä. Parasta olisi, jos käyttäjän tekemät virheet minimoidaan ja ohjelmistokehityksessä niiden syntyminen estetään kokonaan. Tämä ei ole aina mahdollista, mutta käyttöliittymään voidaan lisätä ominaisuuksia, jotka muistuttavat käyttäjää ja estävät mahdollisia virheitä. Käyttäjälle tulisi viestittää, kun hän on tekemässä

virheen. Hyvin yleinen esimerkki tilanteesta on, että käyttäjä on poistumassa sivulta, mutta ei ole tallentanut tietoja. Tällöin usein varoitetaan käyttäjää, mutta ei estetä toimintoa. Lisäksi käyttäjän tulisi voida korjata virheet mahdollisimman helposti. (Nielsen 1993, 32-33.)

Käyttöliittymän tulisi myös kommunikoida käyttäjälle, jotta käyttäjä ymmärtää, mitä milloinkin tapahtuu (Nielsen 1993, 134). Tämä on erityisen tärkeää virhetilanteissa. Esimerkiksi, kun käyttäjä laittaa tietoja lomakkeeseen, tulisi käyttäjälle viestiä selkeästi, miten tiedot tulisi laittaa. Lisäksi olisi olennaista viestiä, jos tiedot eivät ole oikein. Kun lomake lähetetään, on vielä hyvä ilmoittaa, että se on mennyt järjestelmään.

3.5 Käyttöliittymän toimintaympäristön vaikutus käyttäjäkokemukseen

Monesti käyttäjäkokemuksen kehittäminen pyörii järjestelmän käyttöliittymän ympärillä. Käyttäjäkokemukseen kuuluu kuitenkin muutakin kuin ohjelmiston käyttöliittymä. Käyttäjäkokemukseen vaikuttaa sen käyttöympäristö. Käyttöympäristöön kuuluu järjestelmään liittyvä infrastruktuuri, viestintä, ohjeet ja sen käyttöön liittyvä työ. Käyttäjäkokemukseen voi vaikuttaa myös jopa mielipide järjestelmän kehittäjätahosta.

Käyttäjäkokemukseen vaikuttaa merkittävästi myös laitteisto, missä järjestelmää käytetään. Jos laitteisto ei toimi, niin ei toimi järjestelmäkään. Myös järjestelmään sopimattoman tai käyttäjille vaikean laitteiston aiheuttamat ongelmat voivat heijastua käyttäjäkokemukseen. Jos järjestelmä tarvitsee nettiyhteyden toimiakseen, niin internetin saatavuus on äärimmäisen olennaista positiivisen käyttäjäkokemuksen kannalta. Aina käyttäjille ei ole myöskään selvää, milloin järjestelmissä olevat viat johtuvat laitteista ja milloin järjestelmästä. Silloin helposti oletetaan, että järjestelmä on rikki.

Käyttäjäkokemukseen vaikuttaa myös muut ohjelmistot, jotka toimivat ohjelmiston kanssa yhteydessä. Jos järjestelmä toimii internetselaimella, niin selaimen valinta saattaa vaikuttaa järjestelmän toimintaan ja siten käyttäjäkokemukseen. Järjestelmissä on usein rajapintoja muihin järjestelmiin tai toimintoihin, joihin liittyy muita järjestelmiä. Tällöin muissa järjestelmissä olevat ongelmat voivat heijastua käytettävään järjestelmään ja haitata sen käyttöä. Joskus ohjelmistoissa on ominaisuuksia, jotka riippuvat muista järjestelmistä. Jos toinen järjestelmä tai sen rajapinta toimii epävarmasti tai hitaasti, tulisi näiden järjestelmien yhteistoiminta suunnitella niin, että toisen järjestelmän ongelmat eivät heijastu toiseen järjestelmään.

Nykyään nettisivut ja internetselaimissa käytettävät ohjelmat tulisi tehdä niin, että niitä on mukava käyttää myös mobiililaitteilla. Älykännyköiden ja tablettien käyttö on yleistynyt voimakkaasti. Niiden käyttö ei ole kasvanut pelkästään yksityisillä ihmisillä vaan yhä

useammin ne ovat työvälineitä. Internetsivut on hyvä tehdä mobiiliystävällisiksi, jotta niitä on helppo käyttää myös mobiililaitteilla.

Käyttäjäkokemukseen vaikuttaa myös järjestelmään liittyvä viestintä. Viestintää on esimerkiksi se, miten järjestelmästä puhutaan käyttäjille, miten siihen liittyvistä asioista kerrotaan ja miten järjestelmän käyttöön perehdytetään. Tällöin on tärkeää huomioida se kohderyhmä, mille viestitään ja pyrkiä käyttämään kohderyhmän ymmärtämää kieltä. Tämä voi joskus tarkoittaa kirjaimellisesti käyttäjän äidinkieltä. Sen lisäksi on muistettava, että erilaisen kokemuspohjan omaavilla henkilöillä on hyvin erilainen tapa puhua asioista. Hyvin tekninen henkilö käyttää sujuvasti teknistä ammattisanastoa, mutta peruskäyttäjälle se voi kuulostaa aivan heprealta. Järjestelmissä kannattaa käyttää sellaista sanastoa, joka on käyttäjille tuttua. Kun järjestelmän kieli vastaa tosielämää, on käyttäjän helpompi ymmärtää, mitä järjestelmän osat vastaavat. (Nielsen 1993, 123-124.)

3.6 Käyttäjäkokemuksen tutkiminen

Hyvä käyttäjäkokemus on itsessään aika laaja ja osittain abstrakti käsite. Sille on vaikea antaa yhtä kuvaavaa mittaria, mutta tietenkin käyttäjien voimakas negatiivinen tai positiivinen palaute kertoo jotain. Käyttäjäkokemuksen vaikuttavia osa-alueita mitataan ja tutkitaan pääasiassa käyttäjäkyselyillä ja testeillä (Nielsen 1993, 27). Kyselyiden ja testauksen perusteella voidaan löytää ne kohdat, joissa on suurimpia käyttäjiä vaivaavia ongelmia. Ammattimainen näkemys on taas tarpeen, jotta käyttäjien palaute saadaan tulkittua ja muotoiltua parannustoimenpiteiksi.

Käyttäjäkokemuksen tutkimiseen on tullut myös monenlaisia digitaalisia työkaluja. Työkaluista saa dataa käyttöliittymän tehokkuudesta, latausnopeudesta, visuaalisuudesta, saavutettavuudesta sekä käyttäjän toiminnasta käyttöliittymässä. Käyttäjän kursorin tai silmän liikkeitä seuraamalla voidaan havaita, mihin käyttäjän huomio keskittyy sivulla. Tämän avulla voidaan löytää asioita, jotka ovat käyttäjille hankalia ja huomata, mitkä ominaisuudet jäävät käyttäjiltä huomaamatta. Latausnopeus kertoo siitä, että miten sujuvaa sivuston käyttö on. Myös saavutettavuuden tutkimiseen on tullut työkaluja, jotka kertovat sivuston visuaalisten elementtien selkeydestä sekä siitä, miten helppoa sivuilla on navigoida ruudunlukuohjelmilla. Työkaluja voidaan käyttää alustaviin analyysihin. Niillä on helppo saada käsiteltävää dataa ja käyttäjäkokemuksesta kertovia numeraalisia analyysieja. Lisäksi näillä työkaluilla voidaan löytää sellaisia kehityskohtia, mitä käyttäjän on vaikea tuoda esille.

(Iny s.a.)

Ammattilainen saattaa nähdä järjestelmää tutkiessaan selkeitä ongelmia, mutta välillä kehityksessä tarvitaan ulkopuolista näkökulmaa. Käyttäjia seuraamalla ja heillä järjestelmää testaamalla löydetään usein lisänäkökulmia. Käyttäjiltä saatava tieto on tärkeää, jotta järjestelmää voidaan kehittää vastaamaan juuri heidän tarpeitaan (Juvonen 2018, 58-59). Usein järjestelmiä ei tarvitse testata suurella ihmisjoukolla, vaan jopa muutaman ihmisen kanssa testaaminen auttaa käyttäjäkokemuksen kehittämisessä (Grant 2018, 382.).

Järjestelmiä tulisi testata oikeilla tai potentiaalisilla käyttäjillä, jotta käyttäjätestauksesta saadaan paras hyöty (Grant 2018, 360). Jos käytettävyyttä testata tehdään käyttäjillä, jotka tuntevat järjestelmän tulisi kiinnittää huomioita siihen, kuinka kokeneita käyttäjät ovat. Satunnaisen käyttäjän tapa ja nopeus tehdä järjestelmässä toimintoja voi erota kokeneen käyttäjän toiminnasta. Testattavat käyttäjät tulisi valita sen mukaan, mitä käyttäjätestauksella halutaan mitata. Jos esimerkiksi halutaan tutkia järjestelmän helppokäyttöisyyttä, ei ole järkeä valita kokeneinta käyttäjää testaajaksi.

Käyttäjätestausta suunnitellessa on kuitenkin tärkeää, että testin suunnittelijat tutustuvat testattavaan ohjelmistoon. Tällöin he usein huomaavat jo alustavia käytettävyyso ongelmia, joita käyttäjätestauksella validoidaan. Käyttäjäkokemukseen perehtynyt henkilö voi myös tehdä ohjelmistosta heuristien analyysin. Heuristisessa analyysissä asiantuntija tarkastelee ohjelmistoa oman kokemuksen ja tietopohjansa perusteella. Tämän analyysin tukena voidaan käyttää esimerkiksi Molichin ja Neilsenin 1990-luvulla tekemää listaa käyttäjäkokemukseen vaikuttavista asioista. (Nielsen 1993, 155.)

Käyttäjätestaamisessa voidaan käyttää erilaisia tapoja. Yksi käytetty metodi on, että käyttäjälle annetaan muutama tehtävä suoritettavaksi. Käyttäjä suorittaa tehtävät ja kuvailee sanallisesti, mitä hän tekee milloinkin. Tämän lisäksi valvoja seuraa tapahtumaa ja kirjaa ylös huomioita. Tätä menetelmää kutsutaan "Think aloud" -metodiksi. Testin valvoja kirjaa huomioita ylös ja kysyy testaajalta tarvittaessa kysymyksiä. Hän ei kuitenkaan auta tai ohjaa käyttäjää.

Testaamisella voidaan tutkia, miten helppo järjestelmää on käyttää. Tällöin voidaan käyttäjille antaa tehtäviä ja mitata aikaa, joka menee tehtävän tekemiseen. Saatua tuloksia voidaan verrata siihen, mitä tuloksia saadaan muista järjestelmistä tai mikä olisi kohtuullinen aika kyseisen toiminnon suorittamiselle (Nielsen 1993, 31). Toimintoihin kuluva aika voidaan käyttää sen mittaamiseen, että miten helppo ohjelmisto on oppia. Ohjelmistoa voidaan testata henkilöillä, jotka eivät ole sitä aiemmin käyttäneet. Tällöin heille annetaan tehtäviä ja mitataan aika, joka menee tehtäviin alussa ja sen jälkeen, kun he ovat käyttäneet järjestelmää. (Nielsen 1993, 29.)

Käyttäjätestaamisen avulla voidaan kehittää myös järjestelmien saavutettavuutta. Tällöin testauksessa voidaan hyödyntää myös kokemusasiantuntijoita. He ovat ihmisiä, joilla on omakohtaista kokemusta asioista, jotka aiheuttavat vaikeuksia järjestelmien käytössä. Saavutettavuutta voidaan testata myös automaattisilla järjestelmillä.

Testauksen lisäksi käyttäjäkokemuksen kehittämisen apuna voidaan käyttää kyselytutkimuksia. Lomakkeella tehtävät käyttäjäkyselyt ovat parempia silloin, kun halutaan tietoa isolta joukolta ihmisiä. Tällöin on myös hyvä, miettiä kenelle kysely tehdään? Jotta järjestelmää kehityksessä huomioidaan oikeat ihmiset, tulisi kysely osoittaa ihmisille, jotka ovat sitä käyttäjäryhmää, kenen palautetta halutaan. Tätä voidaan myös rajata taustakysymyksillä, jotka auttavat löytämään tietyt käyttäjät.

Kyselyjä tehdessä tulee tarkasti miettiä, mitä kohdeyleisöltä halutaan kysyä. Kysymysten muotoilu on tärkeää, jotta ne eivät ohjaile vastauksia ja niillä saadaan käyttäjiltä oikeaa tietoa. Kyselyt ovat hyvä valinta silloin, jos halutaan kvantitatiivista tietoa. Varsinkin kysymyksistä, joissa on ennalta määritetyt vastausvaihtoehdot, saadaan selkeää kvantitatiivista dataa. Kyselyissä voidaan käyttää myös avoimia kysymyksiä, jolloin käyttäjät pääsevät ilmaisemaan sellaisia tarpeita, jotka eivät muuten kyselyssä tulisi esille. Näiden kysymysten avulla voi tulla esille sellaisia asioita, joita kyselyn laatija ei olisi osannut muuten huomioida.

(Hokkanen 2018, 11.)

Näiden lisäksi tietoa voidaan myös hankkia haastatteluilla. Haastatteluissa on se etu, että haastattelija voi esittää tarkentavia kysymyksiä. Haastattelujen ja käyttäjätestauksen teko on kuitenkin aikaa vievää ja niiden tulosten läpikäyminen työläämpää kuin lomakkeella tehtävässä kyselyssä. Ne voivat olla kuitenkin havainnollisempia ja niissä saattaa tulla esille asioita, jotka muuten olisi jäänyt tutkijalta huomaamatta.

Käyttäjäkokemuksen kehittämisessä empaattisuus ja objektiivisuus ovat tärkeitä ominaisuuksia. Niiden avulla on mahdollista ymmärtää käyttäjiä ja heidän tarpeitaan. Lopulta käyttäjäkokemuksen kehittämisessä on kyse siitä, että muodostetaan käsitys käyttäjien tarpeista ja tehtävistä, mitä he haluavat suorittaa järjestelmällä. Tämän lisäksi pitää pyrkiä miettimään, miten he järjestelmää käyttävät. Saadun tiedon avulla järjestelmää tulisi kehittää sellaiseksi, että käyttäjän on mahdollisimman helppo suorittaa haluamansa tehtävät järjestelmän avulla. (Grant 2018, 2.)

4 Muutoshallinta

Kun organisaatio kohtaa muutoksia on niitä hallittava. Muutoshallinnassa on hallittava muutokseen meneviä resursseja, muutoksen alaisia sidosryhmiä ja muutokseen liittyviä riskejä. Muutoksien on oltava tietoisia ja niiden vaikutuksia tulisi tarkastella, jotta varmistetaan, että ne aikaan saavat halutun tuloksen. Muutoshallinnalla myös varmistetaan, että muutos tapahtuu ja siihen liittyvät toiminnot otetaan käyttöön. Muutoksen hallinnassa on paljon yhtäläisyyksiä projektin hallinnan kanssa. Useissa tilanteissa on myös etua, jos tietojärjestelmänkehitystä varten tehdään projekti.

Muutoshallinnan yksi tärkeä osa-alue on riskien hallinta. Riskejä aiheutuu monista erilaisista syistä ja muutosten vaikutuksia ihmisiin ja järjestelmiin on arvioitava laajalta näkökulmalta (Green 2007, 157). Järjestelmänkehityksessä riskejä aiheutuu sekä teknisistä seikoista että ihmisistä. Kaikkia riskejä ei voida ennakoida. Riskejä kannattaa kuitenkin miettiä ja pyrkiä ehkäisemään ainakin todennäköisimmät riskit, jotka aiheuttavat suurimmat vahingot.

Muutoshallinnassa on tärkeä tunnistaa ne ryhmät, joihin muutokset vaikuttavat. Heidän tulisi ymmärtää miksi muutoksia tehdään. Tämän lisäksi muutoksista on viestittävä heille selkeästi. On tärkeää ymmärtää, mikä motivoi kyseisiä ryhmiä. Tällöin on mahdollista muotoilla muutokset ja niihin liittyvä viestintä niin, että muutos saadaan helpoimmin toteutettua. (Green 2007, 168-169.)

Muutokset voivat joskus aiheuttaa vastarintaa. Ihmisillä on luontainen tarve pysyä tutussa ja turvallisessa ympäristössä. Muutos herättää helposti pelkoa ihmisissä. Muutosvastarintaa esiintyy erityisesti silloin, kun muutos tapahtuu heistä riippumatta, kun he eivät pääse vaikuttamaan muutoksen vaiheisiin tai muutoksen lopputulos on heille epäselvä (Piha & Sutinen 2020, 17).

Organisaation sisäinen kulttuuri vaikuttaa paljon siihen, miten organisaatiossa suhtaudutaan muutoksiin. Organisaatio on kokonaisuus, jonka toimintaan vaikuttaa siinä toimivat henkilöt ja sen ympäristö. Organisaatiossa on ajan mittaan syntynyt tapa tehdä ja reagoida asioihin. IT-pohjainen start-up maailma ottaa muutoksia vastaan helpommin ja joustavammin kuin perinteiseen liiketoimintaan keskittynyt hierarkkinen yritys tai valtion virasto. Kun muutoksia suunnitellaan, tulisi tämä ottaa huomioon ja joskus muutoksien teko vaatii muutoksia myös organisaation sisäiseen kulttuuriin.

(Piha & Sutinen 2020, 23.)

Muutoksien hallitsemisessa on tärkeää, että muutoksen kokevalle yleisölle viestitään muutoksesta ja sen vaikutuksista. Viestinnän avulla käyttäjät ymmärtävät, mitä tapahtuu ja milloin. Tämän lisäksi on tärkeää kommunikoida, miksi juuri kyseiset muutokset tehdään. Vaikka kaikki muutokset eivät ole kaikkien käyttäjien mieleen, niin heidän on helpompi hyväksyä myös negatiiviset muutokset, jos he ymmärtävät miksi niihin on päädytty. Muutoksista kannattaa viestiä niin, että niistä tulee positiivinen, mutta todenmukainen kuva (Green 2007, 169). Näin ihmisille saadaan luotua kuva, että muutos tuo positiivisia asioita mukanaan. Muutoksen viestinnässä tulisi kuitenkin olla rehellinen ja avoin, koska muuten ihmiset voivat pettyä ja tämä voi vaikuttaa heidän motivaatioonsa muutokseen sitoutumisessa. Oikeanlainen viestintä vähentää muutosvastarintaa (Piha & Sutinen 2020, 39).

Viestinnän lisäksi muutoksen alaisia henkilöitä pitäisi myös kuunnella. He ovat usein järjestelmien loppukäyttäjiä ja käytännön työntekijöitä. Johdolla ei ole aina kattavaa kuvaa käytännön työstä. Varsinkin ohjelmistokehityksessä tämä aiheuttaa tilanteita, joissa vaatimusmäärittelyjä saattaa tehdä henkilöt, joilla ei ole täyttä ymmärrystä työprosesseista. Tämä voi johtaa tilanteeseen, jossa ohjelmiston kehitystoimet on määritelty niin, että ne eivät tue käytännön työtä. Tällöin voidaan muutoksilla aiheuttaa jopa haittaa työnteolle. Henkilökunnan mielipiteiden kysyminen on tärkeä osa riskien hallintaa ja sitouttaa henkilökuntaa muutokseen.

(Juvonen 2018, 58-59.)

4.1 Digitaalinen muutos

Digitaalinen kehitys on tällä hetkellä isoimpia yhteiskuntaamme vaikuttavia megatrendejä (Dufva 2020). Digitaalinen kehitys on nopeaa ja tämä on saanut aikaan sen, että ympäristömme muuttuu hyvin nopeasti. Tänä päivänä onkin olennaista, että organisaatiot pystyvät mukautumaan myös digitaalisen ympäristön muutoksiin nopeasti. Tämä ei tarkoita pelkästään uusien tekniikoiden käyttöön ottamista, vaan sitä, että digitaalisen ympäristön muutokset vaikuttavat merkittävästi organisaatioiden sisäisiin digitaalisiin järjestelmiin.

Tietojärjestelmien kohdalla muutoksenhallintaa on olennaista tehdä aina, kun järjestelmiin tehdään isompia kehitystoimia tai hankitaan uusia järjestelmiä. Nämä toimet eivät usein näy pelkästään järjestelmissä, vaan niillä on vaikutusta kaikkiin järjestelmään liittyviin sidosryhmiin sekä infrastruktuuriin. Joskus pelkän järjestelmän muuttaminen ei riitä, vaan samalla pitää tehdä muutoksia myös käyttäjien toiminnassa ja käyttöympäristössä (Talentum 2005, 19).

Riskien hallintaan kuuluu olennaisesti ohjelmistoon tehtävien muutosten vaikutuksien hallinta. Aina kun järjestelmän koodiin tehdään pieniäkin muutoksia, niin niistä voi aiheutua

myös epätoivottuja seurauksia. Järjestelmiin tehtävät muutokset olisi hyvä testata aina ennen kuin ne tehdään tuotantojärjestelmään. Testaus tulisi mieluusti tehdä erillisessä testijärjestelmässä. Tällöin nähdään, saavutetaanko koodin muutoksella haluttu muutos. Lisäksi voidaan havaita negatiiviset muutokset. Testijärjestelmän etu on myös se, että muutokset eivät vaikuta oikean järjestelmän toimintaan tai tietokantaan.

Testausta varten voidaan järjestelmien kehitysympäristöön rakentaa testiautomaatio. Testiautomaatiossa voidaan testata hyvin lyhyitä koodipätkiä tai monimutkaisia kokonaisuuksia. Yksikkötestausta tehdään, kun kehitetään ohjelmistoa ja näillä pienillä testeillä testataan yksittäisiä koodin osia. Kun ohjelmisto kehittyy ja monimutkaistuu, voidaan siihen tehdä myös monimutkaisempia automaattisia testejä. Tällöin testataan usein monimutkaisia toimintoja ja isompia kokonaisuuksia. Näillä integraatiotesteillä testataan, että järjestelmän osat toimivat yhdessä niin kuin pitää. (Kügelgen & Laukkonen 2021, 180-181.)

Kaikkia epähaluttuja ominaisuuksia ei kuitenkaan välttämättä havaita testiautomaatiossa. Testiautomaatiota on melko mahdoton rakentaa täysin aukottomaksi. Monimutkainen testiautomaatio on myös itsessään ohjelmiston ylläpitoa monimutkaistava asia. Epähalutut ominaisuudet voivat myös johtua käyttäjien laitteista tai muista odottamattomista tekijöistä. Tällöin voi käydä niin, että tuotantoon viedään koodia, joka ei toimi halutusti. Tällöin muutosten kanssa joudutaan menemään taaksepäin. Jos ohjelmistolle on tehty versionhallinta, on tämä helppoa. Silloin ohjelmistossa voidaan helposti palata tarvittaessa aiempaan versioon.

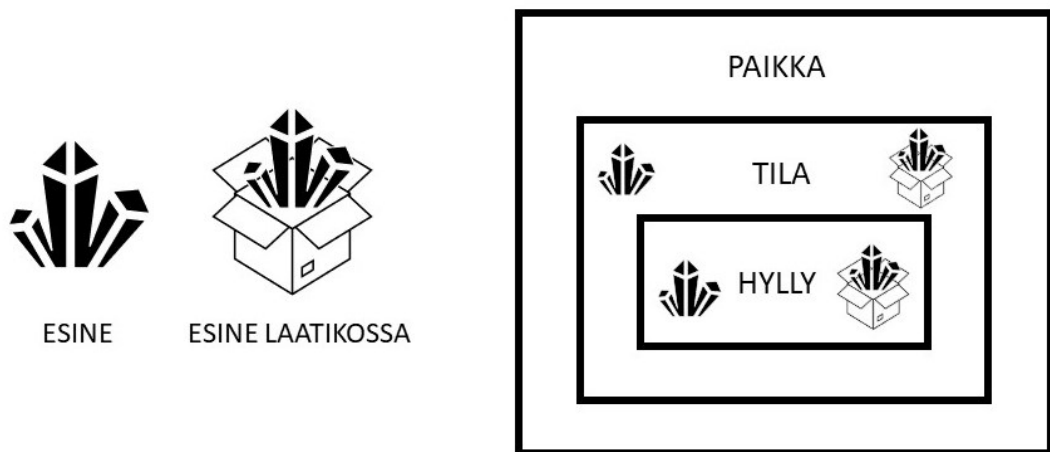
Jos ohjelmistokehityksessä käytetään ulkoisia toimijoita, on hyvä huomioida heidän vaikutuksensa prosessiin. Jos työ tehdään hankintana, on hyvä tarkastella toteuttavaa tahoja monelta kantilta. Osa yrityksistä tarjoaa edullisesti hyvin valmiita palveluita, jotka on tarkoitettu sopivan useille toimijoille ilman räätälöintiä. Silloin kuin hankitaan hyvin erikoistuneita palveluita ja hankintaa tehdään muuttuvaan ympäristöön, on etua siitä, että toimittaja on joustava ja pystyy reagoimaan asiakkaan muuttuneisiin tilanteisiin. Toimittajalla on myös oltava tarpeeksi oikeanlaista osaamista, jotta he pystyvät tarjoamaan sellaisia palveluita, jotka hyödyttävät asiakasta. (Plugge 2012, 145.)

5 Kehitettävä järjestelmä

Kyseessä on logistiikkajärjestelmä nimeltä Muskatti. Muskatilla pidetään kirjaa museoesineiden liikkeistä ja niiden sijainnista. Muskatissa on Museoviraston hallinnoimia tiloja ja niissä olevia sijainteja ja kokoelmien esineitä. Muskatissa on tieto esineiden sijainnista ja liikkeistä sekä museon sisäisistä paikoista missä esineet liikkuvat. Muskatilla kautta myös hallitaan esineiden siirtoja ja tilataan esineitä nähtäville.

5.1 Muskatti-järjestelmä

Muskatissa on esineitä, paikkoja, tiloja, hyllyjä ja laatikoita. Lisäksi Muskatissa on käyttäjiä, asiakkaita, tulostuskärryjä ja lokitietoja. Fyysisten tilojen hierarkia menee niin, että paikassa on tiloja ja tiloissa on hyllyjä. Hyllyillä ja tiloissa voi olla esineitä tai laatikoita, joissa on esineitä.



Kuvio 3. Esineiden ja sijaintien hierarkia Muskatissa.

Muskatissa oleville esineillä on esinenumero, nimi, kuvaus, kuva sekä niiden logistiset tiedot. Logistiset tiedot käsittävät tiedot siitä, missä paikassa, tilassa, hyllyssä tai laatikossa ne ovat, sekä tiedot mahdollisista lainoista tai siirroista. Logistisiin tietoihin kuuluvat myös esineen lokitiedot, joissa on muun muassa tietoa esineen aiemmista liikkeistä. Lisäksi esineille voi kirjata lisätietoja ja laittaa varoitusmerkinnän, jos niissä on käsittelyyn vaikuttavia riskejä, kuten hyönteismyrkyt tai räjähdysriskit valmistusmateriaalit.

Muskatissa on mahdollista muokata näitä tietoja, sekä lisätä esineelle kuvia. Muskatista on yhteys museon kokoelmanhallintajärjestelmään. Yhteyden avulla siirretään tietoa järjestelmien välillä rajapinnan kautta, jotta sitä ei tarvitse tehdä käsin. Tällä hetkellä Muskatista siirrytään esineen tietoja ja kokoelmanhallintajärjestelmään viedään vain sijainti.

Muskattiin luodaan esineitä joko yksittäin tai suurina erinä CSV-tiedoston avulla. Esineen numerolla kokoelmanhallintajärjestelmästä haetaan esineen nimi, kuvailu ja mahdollinen kuva. Samalla esineen tietoihin liitetään id-numero, jonka avulla ne yhdistetään, kun järjestelmät keskustelevat keskenään.

Muskatissa olevilla paikoilla ja tiloilla on numero, nimi sekä sijaintiin liittyviä tietoja. Näiden paikkojen ja tilojen sijaintitiedot eivät muutu. Muskatissa olevilla hyllyillä ja laatikoilla on nimi, numero, mitat (syvyys, leveys ja korkeus), tieto sijainnista ja lokitiedot.

Muskatin avulla esineitä voidaan siirtää paikasta ja tilasta toiseen isoina määrinä. Tämän lisäksi sekä yksittäisille esineille, että esinekokonaisuuksille voidaan merkitä tieto siitä, että ne on lainattu museon ulkopuolelle. Muskatissa on myös ominaisuus, jolla talon sisäiset toimijat voivat tilata esineitä. Esineet noudetaan näiden tilausten perusteella ennalta sovittuun tilaan.

Muskattiin on tehty tuki viivakoodien lukemiselle. Viivakoodit luetaan lukijalla, joka toimii samaan tapaan kuin ulkoinen näppäimistö. Viivakoodeilla voi lukijoiden avulla kirjoittaa tekstiä järjestelmän kenttiin.



Kuva 1. Konservattori lukee esineen numeron viivakoodilapusta.

Järjestelmään on liitetty tulostuskärkyjä, joista saa tulostettua viivakoodilappuja, -tarroja ja paperitulosteita. Esineelle voi tulostaa esinelapun, jossa on esineen nimi ja numero sekä numeroa vastaava viivakoodi. Esineen tiedot voi tulostaa myös paperille. Paperille

tulostuu esineen esikatselukuva, nimi, esinenumero ja esinenumero viivakoodina. Laatikolle voi tulostaa kahta erikokoaa olevat tarrat. Tarroihin tulee laatikon numero, nimi ja numero viivakoodina.



Kuva 2. Tulostuskärry.

Muskatissa on myös koritoiminto. Koriin voidaan kerätä esineitä ja koreja voi tallentaa. Tallennetut korit voidaan myös saada näkymään muille käyttäjille. Tämä auttaa, jos esineitä on ryhmiteltävä. Esimerkiksi tilanteessa, jossa esineille pitää tehdä sama toimenpide. Korissa olevien esineiden lista voidaan tulostaa paperille.

Muskatissa on sisäänkirjautumissivu käyttäjille. Tämän lisäksi heille on mahdollista palauttaa salasana. Järjestelmässä on kaksi eri käyttäjäroolia: normaali käyttäjä ja pääkäyttäjä. Pääkäyttäjä voi lisätä ja poistaa muita käyttäjiä järjestelmässä. Pääkäyttäjällä on pääsy kaikkeen järjestelmän dataan ja he voivat hallinnoida käyttäjien ja asiakkaiden tietoja. Käyttäjät pääsevät muokkaamaan omia tietojaan. He eivät voi kuitenkaan palauttaa salasanaansa tai tehdä itsestään pääkäyttäjää.

Ohjelmistossa on tilastotietoa. Käyttäjät näkevät määriä omista tekemistään toimenpiteistä. Paikoissa pääsee näkemään niissä olevien esineiden määriä. Näitä tietoja ei tois-
taiseksi saa määritetyiltä aikaväleiltä.

Muskatti on dynaaminen nettisivusto. Järjestelmän sisältämä tieto on tallennettu palvelimella toimivaan tietokantaan. Järjestelmää käytetään graafisen käyttöliittymän kautta internet-selaimella. Järjestelmää käytetään sekä pöytäkoneilla että mobiililaitteilla ja se skaalautuu hyvin erilaisille näytöille.

Muskatin kielet ovat HTML, CSS, javascript, ja PHP. Lisäksi järjestelmän tietokanta on MySQL-yhteensopiva. Järjestelmässä on kirjasto, joka tekee CODE 128 -standardin mukaisia viivakoodeja. Muskatti toimii kokoelmanhallintajärjestelmän kanssa käyttäen REST API -rajapintaa. Rajapinnan kautta data tulee XML-tyyppisenä.

6 Käyttäjärühmät

Muskatti on tehty museoesineiden logistiikan hallintaan, joten sen käyttäjinä ovat työntekijät, joiden työt liittyvät museoesineisiin ja niiden liikenteeseen. Tässä työssä esitellään tämänhetkiset merkittävimmät käyttäjärühmät. Tulevaisuudessa Muskatilla voi olla myös muunlaisia käyttäjiä ja käyttäjärühmien merkittävyys voi muuttua. Käyttäjärühmien lisäksi Muskatilla on myös sidosryhmiä, jotka eivät käytä ohjelmistoa, mutta joiden toimintaan järjestelmä liittyy välillisesti.

Muskatin suurin käyttö liittyy tällä hetkellä yhteen Museoviraston merkittävimmistä projekteista. Kansallismuseon esinekokoelmat siirtyvät vanhoista säilytystiloista uusiin. Samalla niiden tietoja täydennetään ja niitä digitoidaan. Digitointi tarkoittaa tässä yhteydessä esineen tietojen tuomista digitaaliseen kokoelmanhallintajärjestelmään fyysisistä luetteloista.

Aikaisemmin esineiden sijainti on merkitty digitaaliseen kokoelmanhallintajärjestelmään vain niille esineille, jotka ovat liikkuneet (näyttelyt, lainat, tutkimus). Muiden esineiden sijainti on merkitty paperilistoille ja Excel-taulukoihin. Tämän siirtoprojektin yhteydessä esineiden sijaintitieto tullaan siirtämään Muskattiin ja kokoelmat pyritään inventoimaan.

Projektin onnistuminen riippuu vahvasti ohjelmiston toiminnasta. Siirtoprojektiin on käytetty merkittävä määrä resursseja ja siinä on selkeä määräaika. Jos tämä projekti epäonnistuu, tulee siitä huomattavia taloudellisia tappioita. Tämän takia siirtoprojektissa toimivat käyttäjät ovatkin tällä hetkellä Muskatin merkittävimpiä käyttäjiä. Sen takia heidän tarpeitaan priorisoidaan kehitystyössä.

Nämä käyttäjät voidaan jakaa kahteen käyttäjärühmään sen mukaan, missä on heidän työpisteensä. Käyttäjät, jotka työskentelevät vanhoissa tiloissa ja lähettävät esineet, käyttävät Muskattia eri tavalla kuin vastaanottavassa päässä. Lähettävässä päässä työtehtäviin sisältyy esineiden tuominen Muskattiin, esineiden tietojen päivitys ja niiden siirtymisen suunnittelu ja kirjaus. Vastaanottavassa päässä Muskatista tarkistetaan ja päivitetään esineiden tiedot.

Näiden käyttäjien lisäksi järjestelmää käyttävät museossa toimivat tutkijat. He kirjaavat esineiden tietoja kokoelmienhallintajärjestelmään ja tilaavat Muskatin avulla esineitä tutkitavaksi, kuvattavaksi tai näyttelyyn.

Lisäksi ohjelmaa käyttää museon konservatorit, jotka kirjaavat järjestelmään esineiden sijaintitietoja ja ennen kaikkea esineiden lainaukset museon ulkopuolelle. Muskattia

käyttää myös muutamat muut esineiden kanssa työskentelevät henkilöt esinelogiikan kirjaamiseen ja fyysisten esineiden siirtojen tilaamiseen.

Muskattia käytetään myös siihen, että sieltä haetaan tietoa kokoelmapalveluiden asiakkaiden laskutusta varten. Muskatissa lasketaan asiakkaiden esineiden viemä säilytystila hyllykuutioina ja asiakkaita laskutetaan tämän kuutiomäärän mukaan. Tätä ominaisuutta tosin hyödyntää vain muutama käyttäjä.

7 Muskatin tilanne ennen kehitystä

Muskatin käyttöliittymä on käyttäjäkokemukseltaan todella hyvä, vaikka järjestelmä onkin keskeneräinen. Käyttäjät pitävät järjestelmästä ja kokevat sen helppokäyttöisenä. Muskatissa on kuitenkin puutteita ja bugeja, jotka aiheuttavat virheitä sekä hidastavat ja haittaavat työtä. Lisäksi järjestelmän käyttövarmuus aiheuttaa huolta.

Muskatti on hyvin yksinkertainen ja se ohjaa käyttäjää. Järjestelmään on tehty hyvin hallittavissa oleva määrä ominaisuuksia. Muskatti ohjaa käyttäjää ja muistuttaa useimmiten olennaisista asioista tarvittavissa kohdissa. Useimmat käyttäjät eivät edes tarvitse käyttöohjetta aloittaessaan sen käyttöä. Tosin on tapauksia, joissa käyttäjän olisi hyvä saada oikeanlainen virheilmoitus. Esimerkiksi sellaisessa tilanteessa, kun käyttäjä yrittää laittaa tyhjän laatikon hyllyyn, tulisi järjestelmän ilmoittaa, että tämä toiminto ei käy.

Muskatin käyttöliittymä on visuaalisesti selkeä ja sivuilla ei ole liikaa asiaa. Siinä olevat visuaaliset elementit erottuvat taustasta ja toisistaan. Tekstit ovat selkeitä ja niiden koko ja kontrasti on hyvä. Värikoodeilla ja ohjaavilla teksteillä on helpotettu tarvittavien toimintojen löytämistä. Järjestelmään on valittu loogisia ja yleisesti käytössä olevia symboleita. Tällöin sivulta on helppo hahmottaa, missä eri toiminnot ovat ja mitä ne tekevät.

Suurin osa järjestelmän toiminnoista on helppo löytää. Muskatissa on sivun vasemmassa reunassa navigointipalkki, josta pääsee eri sivuille. Sivuilta melkein kaikki käyttäjien eniten käyttämät toiminnot. Osa toiminnoista on piilotettu oikean yläkulman hampurilaisvalikon alle. Näiden toimintojen löytäminen on haastavampaa, mutta käyttäjiltä tulleen palautteen ei ilmene, että niitä ei olisi löydetty.

Muskatin kohdalla negatiivisesti sen käyttäjäkokemukseen vaikuttaa merkittävästi se, että järjestelmä on kesken. Siitä puuttuu monia käyttäjille tärkeitä ominaisuuksia, joita siihen on suunniteltu. Muskatissa on myös ohjelmistovirheitä ja epäintuitiivisia ominaisuuksia.

Järjestelmä ei aina toimi niin kuin käyttäjien prosessit vaativat. Muskatissa on esimerkiksi sellainen toiminto, jossa suuria määriä esineitä siirretään paikasta toiseen. Joskus esineet joutuvat odottamaan käsittelyä eri tilassa kuin missä ne käsitellään. Esinekokonaisuuksien käsittelyssä voi mennä paljon aikaa. Esineet pitäisi pystyä paikantamaan sekä ennen niiden käsittelyä että käsittelyn aikana. Nyt siirtoa ei voi siirtää edelleen vaan kaikki esineet olisi siirrettävä yksitellen tai tehtävä uusi siirto. Tämä on taas hyvin hidasta, jos siirrossa on satoja esineitä.

Järjestelmässä on myös mahdollista tehdä sellaisia toimintoja, jotka eivät ole tosielämässä mahdollisia. Nämä aiheuttavat virheitä objektien tietoihin ja lokitietoihin. Virheet taas aiheuttavat lisätyötä ja selvittelyä. Järjestelmässä on muun muassa sellainen virhe, että laatikossa oleva esine voidaan sijoittaa uuteen hyllyyn ottamatta sitä laatikosta. Jos laatikossa on useita esineitä, niin ne jäävät kuitenkin vanhaan hyllyyn. Uuden hyllyn ja siirretyn esineen lokitietoihin tulee näkyviin, että esineen sisältävä laatikko siirtyy uuteen hyllyyn. Laatikko tulee näkyviin myös uuteen hyllyyn. Laatikon omissa tiedoissa kuitenkin näkyy, että se on vanhassa hyllyssä. Tämä aiheuttaa selvitystyötä.

Muskattiin tehtiin muutoksia syksyllä 2019 ja se yhdistettiin kokoelmienhallintajärjestelmään. Tässä yhteydessä Muskattiin tuli virheitä, joita ei ehditty korjata. Useimmat virheet liittyvät näiden järjestelmien yhteistoimintaan, mutta virheet koskevat myös Muskattiin aiemmin tehtyjä toimintoja. Näihin virheisiin kuuluu esimerkiksi se, kun useita esineitä hyllyyn laitettaessa lokitiedot muodostuvat väärin. Ongelma on, että lokitietoihin tallennetaan jokaiselle esineelle sama merkkijono, johon on kirjoitettu vain ensimmäisen esineen numero. Lokitietojen esinenumerossa on linkki esineen sivulle, linkit kuitenkin osoittavat oikein eri esineisiin teksti on vain väärin. Lisäksi toiminnallisuus, jonka avulla kokoelmanhallintajärjestelmässä olevat kuvat siirtyvät Muskattiin, toimii väärin. Toiminnon yhteydessä pitäisi tarkistaa, onko esineelle jo tuotu kuvat toisesta järjestelmästä. Tämä ei kuitenkaan toimi ja nyt kokoelmanhallinnassa olevia kuvia tuodaan uudestaan ja uudestaan. Tämä tarkoittaa, että sama kuva tallennetaan moneen kertaan ja samalla siitä tehdään oletuskuva. Toisesta järjestelmästä tuodut kuvat eivät välttämättä kuvaa hyvin esinettä. Tämä vie turhaan palvelimelta tilaa ja kuvien käsin poistaminen, on työlästä.

Muskatissa ei myöskään ole huomioitu sitä, jos esineen numeroa tarvitsee muuttaa. Esinenumerot voivat olla esineissä virheellisesti tai esineet on voitu tuoda järjestelmään vahingossa väärällä numerolla. Kokoelmien siirtoprojektissa kokoelmien esineitä käydään läpi ja samalla kokoelmia inventoidaan. Tässä kohtaa tulee esille paljon ajan mittaan tehtyjä virheitä, joita korjataan. Jos esinenumeroa muutetaan, ei se kuitenkaan muutu mihinkään esineeseen liittyvään lokitietoon. Tämän lisäksi esineen linkitys kokoelmanhallintajärjestelmään ei muutu ja silloin esineen sijaintitiedot viedään kokoelmanhallintaohjelmassa väärälle esineelle. Tällä hetkellä korjaus on kuitenkin tehtävä niin, että vanha esine poistetaan järjestelmästä ja esine tuodaan oikealla numerolla uudestaan. Tällöin menetetään esineen lokitiedot.

Järjestelmän käyttäjille on tehty hyvin selkeät ja helppokäyttöiset käyttöohjeet. Käyttöohjeet on jaettu selkeisiin kokonaisuuksiin ja niistä on helppo löytää tarvitsemansa asiat. Käyttöohjeissa on painotettu sitä, miten järjestelmää voi käyttää. Ohjeissa on kuitenkin

hyvin vähän ohjeita siitä, järjestelmää pitäisi käyttää. Monet käyttäjät eivät myöskään tutustu käyttöohjeeseen. Kun yhteisiä sovittuja käytäntöjä ei ole tai käyttäjät eivät ole niistä tietoisia, niin he käyttävät järjestelmää oman mielensä mukaan. Eri toimijoilla on erilaisia tapoja käyttää järjestelmää ja eroavaisuuksia on jopa samassa yksikössä toimivien henkilöiden välillä. Tämä aiheuttaa usein ongelmia järjestelmän käytössä. Osa työtavoista on sellaisia, että niistä aiheutuu paljon virheitä esineiden tietoihin. Virheiden selvittämiseen menee paljon työaikaa ja se turhauttaa käyttäjiä. Tämä vaikuttaa negatiivisesti järjestelmän käyttäjäkokemukseen ja työyhteisön työilmapiiriin.

Järjestelmässä on paljon tietoa säilöttynä. Tämän tiedon menettäminen aiheuttaisi merkittäviä taloudellisia menetyksiä. Pelkästään esineiden tietojen uudelleen tuominen veisi yhdeltä työntekijältä useamman vuoden työn. Tämän lisäksi pitäisi tuoda uudestaan, paikat, tilat, laatikot ja kaikki muu tarpeellinen tieto. Paikkatietojen menettäminen myös hidastaisi työntekoa ja haittaisi useita museon toimintoja merkittävästi pitkän aikaa. Tämän lisäksi menetettäisiin se työ, mikä on mennyt järjestelmässä olevien kuvien ottamiseen. Järjestelmän tiedoista on tehty varmuuskopio, mutta sen palauttamista ei ole testattu.

Muskatin suunnittelussa ei ole mietitty sen elinkaarta tai ylläpitoa. Se on tehty vastaamaan tiettyä tarvetta ja se on muutaman ihmisen projekti. Muskatin kehityksen tavoitteena on ollut saada järjestelmä valmiiksi. Eikä sen suunnittelussa ei ole tehty suunnitelmaa sen ylläpidolle. Ohjelmistotuotannossa ei kuitenkaan voi ajatella, että ohjelmisto voitaisiin jättää koskematta, kun sen kehitys on valmis. Järjestelmien käyttöympäristöt ja tekniikka muuttuvat ajan mittaan. Jos muutoksiin ei reagoida, niin pitkällä aikavälillä tämä aiheuttaa ensin järjestelmän tietoturvan heikkenemisen ja lopulta järjestelmä ei enää toimi päivitettyissä laitteissa tai selaimissa. Muskatin kehityksessä ei ole myöskään huomioitu sitä, miten sen käyttö ja siihen liittyvät tarpeet muuttuvat ajan mittaan. Nyt on kuitenkin tiedossa, että tulevina vuosina on tulossa sellaisia projekteja, jotka vaativat merkittäviä muutoksia järjestelmään.

Järjestelmän tekijä on ollut yksin vastuussa kehityksestä. Hän ei ole tästä syystä juurikaan dokumentoinut järjestelmän teknistä toteutusta. Järjestelmästä ei ole virallista kuvausta tai dokumentaatiota. Käyttäjille on tehty muutamia dokumentteja, jotka selittävät heille järjestelmän toimintaa. Niissä ei kuitenkaan ole kuvattu järjestelmää sillä tasolla, että toinen kehittäjä pystyisi käyttämään niitä hyväksi järjestelmän jatkokehityksessä. Koodia on hieman dokumentoitu ja se helpottaa koodin ymmärtämistä. Mutta koodin kommentitkin tuntuvat olevan tarkoitettu järjestelmän alkuperäiselle tekijälle muistin tueksi.

Museovirastolla ei ole ollut järjestelmällistä tapaa viestiä tietojärjestelmiin tehtävistä muutoksista. Varsinkaan Muskattiin tehdyistä muutoksista ei ole viestitty käyttäjille. Usein kyseiseen järjestelmään tehtävät muutokset ovat tapahtuneet yllättäen. On ollut tilanteita, joissa käyttäjät ovat jopa luulleet tarkoituksen mukaisia muutoksia Muskatin bugeiksi. Tämä on aiheuttanut hämmennystä ja ylimääräisiä yhteydenottoja pääkäyttäjille sekä IT-tukeen.

Tietoturvan kannalta ohjelmiston suurimmat ongelmat ovat tiedon eheyden ja saatavuuden kanssa. Muskatissa on paljon tietoa, jonka eheyteen ei voi luottaa. Esimerkkinä on Muskatissa olevat lokitiedot, jotka eivät aina pidä paikkaansa. Muskatin toiminnot eivät ole myöskään aina kaikilta osiltaan saatavilla kaikille käyttäjille. Muskattia käytetään internetin kautta. Museovirastolla on kohteita, joissa on haasteita internetyhteyden laadun kanssa ja se on aiheuttanut paljon ongelmia järjestelmän käytössä. Tämä on saattanut näkyä siinä, että kaikissa kohteissa ei kaikki järjestelmän toiminnot toiminnassa. Esimerkiksi osa tulostuskärryistä on ollut välillä pitkiäkin aikoja pois käytöstä.

8 Kehitystarpeiden kartoitus

Muskatin keskeneräisyys haittaa usean henkilön työtä. Jo vuodesta 2019 käyttäjät ovat pyytäneet järjestelmän viimeistely- ja korjaustoimia. Tätä varten aloitettiin muutostarpeiden alustava kartoitus ja muutostöiden hankinta. Hankintaa varten tarvittiin ajankohtaista tietoa siitä, mitkä ovat merkittävimmät ongelmat.

Priorisoinnin avuksi selvitettiin kyselytutkimuksella, mitkä ovat Muskatin tärkeimmät toiminnot ja mitä käyttäjät käyttävät eniten. Tällöin korjaustoimenpiteet voidaan kohdistaa juuri niihin toimintoihin, joiden korjaamisella saadaan suurin hyöty. Kysely oli myös osa viestintää ja sillä oli tarkoitus viestiä, että Muskattia kehityksen kanssa tehdään töitä.

Kehitystoiveissa oli muutamia asioita, jotka oli kuvattu epäselvästi tai puutteellisesti. Näiden kohdalla tehtiin lisäselvityksiä, jotta saataisiin selville se, mitä nämä toiveet todella tarkoittavat. Tutkimuksilla pyrittiin löytämään se, miten toiminto tulisi korjata tai kehittää, jotta kehittäjälle saadaan selkeä kuva kehitystoiveista.

8.1 Kehitystarpeiden alustava kartoitus

Muutostarpeiden alustava kartoitus alkoi keväällä 2020. Aluksi käyttäjiä pyydettiin listamaan toiveitaan yhteiselle kommunikointialustalle. Tämän lisäksi kehityspyyntöjä läheteltiin myös suoraan Muskatin pääkäyttäjille. Näiden pyyntöjen pohjalta tehtiin taulukko, johon muutostoiveet kirjattiin. Korjaustoiveisiin oli tässä vaiheessa merkitty pääkäyttäjän arvioima tärkeysaste. (Liite 1.)

Toiveita kerättiin aktiivisesti ja niitä pyydettiin kaikilta Muskattia käyttäviltä henkilöiltä. Toiveita pyydettiin juuri käyttäjiltä, jotta korjaukset tukisivat sen kautta tehtävää työtä. Priorisointia varten toiveille oli alusta alkaen asetettu tärkeysaste, joka kertoi sen kuinka olennainen kyseinen muutos olisi. Taulukko oli kuitenkin vain kokoelmakeskuseräyksen näkökulmasta tehty. Siihen oli kerätty kaikki kehitystoiveet ilman kritiikkiä ja niiden tärkeysasteen oli määritellyt yksi henkilö omasta näkökulmastaan. Toiveiden koostamisen aikana ei mietitty sitä, mitkä ovat merkittävimmät käyttäjäryhmät ja mitkä ovat koko organisaation kannalta merkittävimmät muutoskohteet. Tämä johtuu siitä, että koostamista tehtiin yhden yksikön voimin ja tekijöillä ei ollut kokemusta eikä teorettista ohjelmistotuotannosta tai käyttäjäkokemuksen kehittämisestä.

Aluksi taulukossa olevat ongelmat oli välillä kuvattu niin, että oli vaikea ymmärtää, mistä on kyse. Jos taulukkoa katsoo henkilö, joka ei käytä säännöllisesti työssään Muskattia, on melkein mahdotonta ymmärtää osaa toiveista. Kun korjauksia kuvaillaan hyvin vajailla

tiedoilla kasvaa riski, että kehittäjä ymmärtää pyynnot väärin. Näin oli myös aiemmin käynytkin kyseisen järjestelmän kohdalla.

Keväällä 2020 siirryimme etätöihin hyvin nopealla aikataululla. Sain tehtäväkseni miettiä, miten voisin olla mukana Muskatin kehityksessä. Tällöin sain myös järjestelmän lähdekoodit ja pääsin tutustumaan sen toimintaan laajemmin. Tutkiessani käyttäjätoiveet sisältävää taulukkoa huomasin, että en oikein ymmärtänyt osaa toiveista. Toin esille huomioimani mahdolliset riskit ja sain tehtäväkseni täydentää taulukkoa ja auttaa määrittelemään muutostyömenpiteitä. Aloitin taulukon selventämisen lisäämällä sinne sarakkeen, johon pyrin kuvailemaan tehtävän muutoksen selkeästi (Liite 2). Pyrin siihen, että kehittäjän olisi helppo ymmärtää, mitä halutaan. Moneen kohtaan lisäsin myös selvittäviä kysymyksiä, koska minullekin oli epäselvää, mitä haluttiin. Työn aikana huomasin, että olin itsekin todellakin käsittänyt osan muutostoiveista väärin, vaikka käytän ohjelmaa päivittäin.

Taulukon pohjalta oli koostettu PowerPoint-esitys (Liite 3). Tämän oli tarkoitus olla tarjouspyynnön pohjana ja sen avulla ohjelmiston kehitystarpeet esiteltiin johtoryhmälle. Tämän esityksen perusteella pyydettiin tarjousta kehittäjältä. Esitykseen oli kuitenkin kirjattu hyvin moniselitteisiä korjaustoiveita. Kehittäjä ei pystynyt arvioimaan hintaa esityksen perusteella. Ei olisi ollut mahdollista, että korjauksia olisi lähdetty toteuttamaan pelkästään koostetun PowerPointin pohjalta ilman merkittävää lisäselvitystyötä. Kehittäjän olisi pitänyt lisätä töihin ensin kustannus sille, että hän selvittää korjauskohteet ja vasta sen jälkeen olisi ollut mahdollista tehdä virallinen tarjous.

Normaalistikin kehittäjät käyttävät reilusti alle puolet työajastaan varsinaiseen koodin kirjoittamiseen (Crispin 2015; Dasdan 2018). Muu aika menee muun muassa suunnitteluun ja vaatimusten selvittämiseen. Jo yksinkertaisen ja selkeänkin pyynnön kohdalla kehittäjän on mietittävä, miten sen toteuttaa. Mutta jos toive on epäselvä ja vaatii lisäselvitystä tarkoittaa se vielä enemmän työtä. Jos korjauksia olisi lähdetty toteuttamaan tämän listan perusteella, se olisi saattanut nostaa hankinnan hintaa todella merkittävästi. Lisäksi oli todella iso riski, että korjauksien toteutus ei olisi ollut se, mitä haluttiin.

Tässä kohtaa alkoi tämän työn tekeminen. Jotta kehittäjä voisi antaa tarjouksen ja hankinnassa voitaisiin priorisoida tärkeimpiä asioita, niin korjauskohteista tarvittiin selkeämpi priorisoitu lista. Kun tiedossa on korjausten tärkeysaste, niin tiedetään mihin kohteisiin tulisi budjetissa olevat määrä rahat käyttää. Eli mitkä kokonaisuudet tulisi korjata ensin.

8.2 Kysely

Alkuperäinen ajatus oli kysyä käyttäjiltä mielipidettä korjattavista toiminnoista. Suunnitelma oli kysyä heiltä, että mitä he haluaisivat Muskatissa korjattavan ja mitkä ongelmat hidastavat heidän työtään eniten? Huomattiin kuitenkin, että tämä oli huono tapa kartoittaa korjauskohtia. Ensinnäkin korjaustoimenpiteitä ei ollut vielä määritelty ja osa niistä oli hyvin epäselviä. Olisi ollut hankalaa muotoilla niitä kyselyyn niin, että kaikki käyttäjät olisivat ymmärtäneet ne samalla tavalla. Toinen ongelma koski sitä, että minkälaista tietoa kyselystä näin saisi. Todennäköistä olisi, että kaikki olisivat valinneet kaikki kohtaamansa ongelmat. Tällöin tulos olisi saattanut näyttää siltä, että kaikki korjaukset ovat tärkeitä ja mitään ei voi priorisoida. Lisäksi tällä hetkellä muutostoiveita on melkein 20, joten kyselystä olisi tullut myös aika pitkä. Tultiin siihen lopputulokseen, että kysymysten asettelut on mietittävä uudestaan.

Jotta sopivat kysymyksien muotoilussa pohdittiin, mitä tietoa kyselyllä olisi tarkoitus saada, ja millä kysymyksillä saisi esille tietoa, jonka perusteella korjattavia ominaisuuksia voisi priorisoida. Kun alkuperäisiä muutostoiveita oli tarkasteltu, niin huomattiin, että ne on helpointa ryhmitellä toimintojen mukaan. Tästä heräsi ajatus, että olennaisin tieto, mitä tarvitaan ei ole se, mitä käyttäjät haluavat korjattavan, vaan se, miten he ohjelmistoa käyttävät ja mitkä toiminnot ovat heille tärkeimpiä.

Kysymyksillä lähdettiin selvittämään sitä, miten ohjelmaa käytetään ja mikä sen käytössä vaikeuttaa työtä eniten. Tämän lisäksi taustatietona kysyttiin vastaajien käyttäjäryhmää. Näin voidaan priorisoida niiden käyttäjien mielipiteitä, kenen työ riippuu eniten järjestelmästä. Tämän lisäksi kyselyyn laitettiin yksi avoin kysymys, johon käyttäjät pystyivät kirjoittamaan lisätoiveita. Taustakysymys ja monivalintakysymykset määriteltiin pakollisiksi, mutta avoimelle palautteelle tarkoitettu kenttä oli vapaaehtoinen.

Kysymykset:

- Taustakysymys: Mihin käyttäjäryhmään kuulut? Vastausvaihtoehdot: Kokoelmakeskusyksikkö: kokoelmakeskuksessa työskentelevä, Kokoelmakeskusyksikkö: keskusvarastossa työskentelevä, Tutkija, Konservointiyksikön konservaattori tai muu.
- Monivalintakysymys, jossa sai valita useita vaihtoehtoja: Mitä ohjelmiston toimintoja käytät? Vastausvaihtoehdot: Ohjelmiston merkittävimmät toiminnot.
- Monivalintakysymys, jossa sai valita useita vaihtoehtoja: Mitkä toiminnot vaikuttavat työskentelysi tehokkuuteen eniten? Vastausvaihtoehdot: Ohjelmiston merkittävimmät toiminnot.

- Kysymys, johon sai vastata vapaalla tekstillä: Minkälaisia muutoksia tai korjauksia toivot Muskattiin?

Kysely toteutettiin Google Forms –lomakkeella. Alun perin oli suunniteltu, että kysely tehtäisiin Windowsin Teams-alustan kyselytoiminnolla. Googlen kyselyssä oli kuitenkin paremmat ominaisuudet, joten se valikoitui kyselyn toteuttamislustaksi. Ensinnäkin Googlen lomakkeessa on mahdollista asettaa monivalintakysymysten vastauksille vaihtuva järjestys. Tämä vähentää riskiä, että vastaajat valitsevat vastauksia sen mukaan, missä kohtaa vastausvaihtoehto sijaitsee. Teamsissa tämä ei ollut mahdollista niissä kysymysrakenteissa, mitä halusin käyttää. Meillä on ollut myös välillä ongelmia, että Teamsissa kaikki käyttäjät eivät ole päässeet haluamaansa sisältöön käsiksi. Googlen lomakkeeseen vastaaminen ei vaadi sisäänkirjautumista, joten ei käy niin, että joku jättää vastaamatta, kun ei pääse lomakkeeseen.

Linkki kyselyyn lähetettiin sähköpostilla kaikille ohjelmiston käyttäjille. Kyselystä muistutettiin vielä toisella viestillä. Lomakausi oli valitettavasti pahasti päällä, kun kysely tehtiin. Kyselylinkki lähetettiin 66 henkilölle, joista 18 ei ollut kyselyn toteutusaikaan saavutettavissa. Paikalla olevat käyttäjät vastasivat kuitenkin kyselyyn aktiivisesti ja siihen tuli 22 vastausta. Eli kolmasosa käyttäjistä vastasi kyselyyn.

Kyselyyn vastanneissa oli määrällisesti eniten niitä, jotka myös käyttävät ohjelmistoa paljon. He ovat henkilöitä, joille ohjelmisto on työn kannalta tärkeä. Tämä oli erittäin positiivista, koska se kertoo mielestäni siitä, että he haluavat auttaa ohjelmiston kehityksessä. Lisäksi tämä myös kertoo siitä, että käyttäjille muutostarpeiden toteuttaminen on tärkeää.

Kyselyn tuloksista näkyi se, että eri käyttäjäryhmille eri toiminnot ovat tärkeitä (Liite 4). Tässä ei ollut yllättäviä tuloksia, mutta vastaukset vahvistivat alkuperäistä oletusta ohjelmiston eniten käytetyistä toiminnoista ja siitä, mitkä asiat hidastavat työskentelyä eniten.

Kyselystä tuli esille vahvasti se, että järjestelmässä olevien kuvien toivotaan siirtyvän koelmienhallintajärjestelmään. Tätä ominaisuutta on toivottu ja sen kehittämistä on tavoiteltu voimakkaasti muutamien henkilöiden toimesta. Se näkyi myös kyselyssä vahvasti.

Vaikka yritin muotoilla kysymykset niin, että ihmiset eivät listaisi ongelmakohtiksi kaikkia käyttämiään toimintoja, tekivät vastaajat kuitenkin osittain niin. Minun olisi pitänyt vielä miettiä toista kysymystä tarkemmin ja se olisi kannattanut muotoilla niin, että huomio olisi kohdistunut niihin toimintoihin, joissa ovat ongelmat vaikuttavat työn tehokkuuteen eniten. Lisäksi olisi voinut myös kartoittaa, mitkä ovat kullekin ryhmälle työn kannalta

tärkeimmät toiminnot. Tässä olisi voinut vielä rajata vastausvaihtoehtojen määrää, jolloin ihmiset olisivat joutuneet priorisoimaan.

Kyselyssä näkyi hyvin selkeästi Muskattiin rajapinnan kautta liitetyn toisen kokoelmanhallintajärjestelmän ongelmat. Kokoelmanhallintajärjestelmässä on havaittu hitautta. Tämä hitaus heijastuu myös API-rajapinnan kautta muihin rinnalla käytettäviin ohjelmistoihin. Ja hitaus vaikuttaa myös Muskatin käyttöön. Kyselyn aikaan kokoelmanhallintajärjestelmän käyttäjiltä kerättiin tietoa kyseisen hitauden ilmenemisestä ja mielestäni tämä oli yksi esimerkki siitä, miten hitaus työssämme näkyy. Tämän takia kyselyn tiedot luovutettiin myös yksikölle, joka työskentelee kokoelmanhallintajärjestelmän kanssa, jotta he voivat välittää ne palvelun tarjoajalle.

8.3 Käyttäjätestaus

Muskatissa on ominaisuus, jonka avulla sinne voi tuoda ison määrän esineitä CSV-tiedoston avulla. Tässä toiminnossa oli ongelma, josta käyttäjillä oli erimielisyyttä. Osa käyttäjistä sanoi, että Muskatti lukee vain tietyn määrän tiedoston riveistä ja lopettaa. Osa käyttäjistä sanoi, että tällaista ongelmaa ei ole. Lisäksi kukaan ei osannut sanoa, mikä kyseinen raja on. Itselleni tämä ominaisuus oli vieras, joten päädyin siihen, että sitä olisi hyvä testata. Tämä ei ollut toiminnon ainoa ongelma. Testauksella pyrittiin kartoittamaan kaikkia tähän liittyviä ongelmia. Ennakkoon oli tiedossa, että CSV-tiedoston lukemisen yhteydessä tiedostossa esinenumero kentässä olevien esinenumeroitten kirjaimet muutetaan isoiksi kirjaimiksi. Tämä ei ollut haluttu toiminto, koska esinenumeroissa osa kirjaimista on tarkoituksella pieniä kirjaimia.

Testaus toteutettiin niin, että etäyhteyden avulla seurattiin henkilöä, joka toteutti ennalta määritellyn tehtävän liittyen tähän toiminnallisuuteen. Samalla hän kertoi "Think aloud" -menetelmän mukaisesti toiminnastaan. Testissä käyttäjä näytti, miten hän laittaa esinenumerot CSV-taulukkoon ja tallentaa taulukon. Tämän jälkeen käyttäjä latsi taulukon järjestelmään. Taulukolla tuotiin testin aikana vain esinenumerot, vaikka sen avulla voisi tuoda myös esineen nimen ja kuvailun. Esineen nimi ja kuvailu haettiin kokoelmanhallintajärjestelmästä rajapinnan kautta taulukon tuomisen jälkeen erikseen.

Tässä testissä ei havaittu, että tiedoston lukeminen olisi katkennut väärästä kohdasta, mutta käytetyssä tiedostossa oli vain alle sata riviä. Testauksen aikana tuli esiin useita muita ongelmia. Osalle ongelmista löydettiin aiheuttaja tai korjausta helpottava tekijä tämän testauksen tuloksien avulla.

Käyttäjä kertoi testauksen aikana, että ohjelma tekee tyhjiä kenttiä, jos taulukkoon on jäänyt muotoiluja. Tätä ominaisuutta testattiin vielä käyttäjätestauksen jälkeen ja havaittiin, että tyhjiä rivejä tulee aina, jos taulukon solussa on yli 1000 merkkiä. Järjestelmään oli ilmeisesti laitettu raja, että näin monta merkkiä voidaan ottaa lomakkeen solun sisään. Usein esineiden kuvailuissa tämä raja tulee vastaan ja aiheuttaa ongelmia. Ongelmia ei aiheudu, jos tiedot tuodaan kenttiin kokoelmanhallintajärjestelmästä, koska nämä tiedot haetaan vasta lomakkeen latauksen jälkeen ja niissä tulevat merkkijonot käsitellään ajoissa.

Testauksen aikana tuli esiin myös lisää käyttäjäkokemukseen negatiivisesti vaikuttavia tekijöitä. Kun taulukko tuodaan sisään ja siinä on jo järjestelmässä olevia esineitä tai vääriä muotoiluja, niin taulukko joudutaan muokkaamaan uudestaan ja tuomaan järjestelmään uusiksi. Tähän menee paljon aikaa ja se on työlästä. Olisi huomattavasti nopeampaa, jos taulukon latauksen jälkeen voisi käyttöliittymästä poistaa lomakkeelta turhat rivit ja jo järjestelmässä olevat esineet.

Kun esineiden tiedot haettiin kokoelmanhallintajärjestelmästä, niin osa esineiden nimistä meni esineiden kuvailukenttään. Tämä jouduttiin korjaamaan käsin ja vei paljon aikaa. Tämän selvittämiseksi kokoelmanhallintajärjestelmästä tutkittiin, minkälaisia nimiä esineillä oli. Tässä havaittiin, että jos esineellä on vain yleisnimi -tyyppiä olevia nimiä, niin nimet eivät mene nimikenttään, vaan kuvailukenttään. Mutta jos esineellä on yleisnimen lisäksi muun tyyppisiä nimiä, niin nimet menivät oikeisiin kenttiin. Muut tiedot menivät oikeisiin kenttiin.

Testauksen aikana ei tulostettu esineiden viivakoodilappuja. Testattava totesi kuitenkin, että olisi toivottavaa, että esineiden viivakoodilaput tulostuisivat numerojärjestyksessä. Tässä kokeiltiin, että mitä tapahtuu, jos tuotava tiedosto järjestellään ennakkoon. Tällä toimenpiteellä saatiin viivakoodit tulostumaan halutussa järjestyksessä.

Tiedoston lukemisen katkeamista varten tehtiin vielä toinen testaus sellaisella listalla, jonka kanssa tämä ilmiö oli havaittu. Tiedoston katkeamista ei kuitenkaan pystytty toteuttamaan. Virhe tapahtui siinä vaiheessa, kun esineet tuodaan järjestelmään (kyseiset esineet olivat järjestelmässä, joten niitä ei voitu tuoda sinne uudestaan). Ongelma kuulosti siltä, että ohjelma lopettaa toiminnon tuntemattomasta syystä kesken. Tässä kohtaa päädyttiin jättämään syyn selvittäminen ohjelmistokehittäjälle.

Tiedoston tuomiseen liittyviä ongelmia testattiin vielä tuomalla muutama väärin muotoiltu tiedosto. Tiedostoissa oli jätetty pois otsikko kenttiä ja näiden tiedostojen kohdalla lataus

ei toiminut. Tässä kohtaa havaittiin, että tiedoston muotoilussa pitää olla tarkka ja tiedoston tuomisessa havaittuja ongelmia voidaan korjata tiedoston muokkaamisella. Tässä kohtaa päädyttiin siihen, että käyttäjiä ohjeistamalla voidaan myös korjata ongelmia.

8.4 Haastattelu tilastotoiminnosta

Muskattiin toivottiin muutamaa uutta toimintoa. Nämä uudet toiminnot nopeuttavat ihmisten työtä ja automatisoivat sisäisiä prosesseja. Uusina toimintoina toivottiin tilastotoimintoja, joilla ohjelmasta saisi suoraan tietyllä aikavälillä tehtyjen tapahtumien määriä. Tämän lisäksi toivottiin lisätoiminnallisuutta, jonka avulla järjestelmään lisätyt kuvat siirtyisivät toiseen järjestelmään automaattisesti. Järjestelmässä on myös laskennallisia tietoja, jotka vaikuttavat siihen, miten asiakkaita laskutetaan. Nyt näitä tietoja joudutaan laskemaan käsin ja se vie työaikaa usean yksikön työntekijöiltä turhaan, kun tämän voisi automatisoida.

Näitä toimintoja ei voi luonnollisesti testata, mutta olisi kuitenkin hyvä saada ohjelmiston kehittäjälle selkeät raamit siitä, miten kyseiset toiminnallisuudet olisi hyvä toteuttaa. Jotta näistä ominaisuuksista saadaan sellaiset, että ne palvelevat käyttäjiä.

Tilastot ovat esimiehille tärkeitä ja niillä raportoidaan tehdystä työstä ylemmille tahoille. Tämä auttaa resurssien suunnittelussa ja kertoo sen, mihin meidän työaikamme menee. Museoviraston toimintakertomukseen on myös liitetty yksikkömme keräämiä tilastotietoja. Tähän mennessä tilastoja on laskettu käsin tai Excel-taulukoihin merkityistä luvuista. Tämä on aikaa vievää ja välillä muun työn ohessa lukujen kirjaus helposti unohtuu. Osa tilastoista on myös vain tiettyjen henkilöiden saavutettavissa ja silloin tilastotietoja ei saada, jos kyseinen henkilö ei ole enää töissä tai hänen arkistonsa häviävät.

Ohjelmistoon haluttiin toiminto, josta saa helposti tilastotietoja. Nyt ohjelmistosta saa vain muutamia tietoja ja niistäkään ei voi valita tiettyä aikaväliä. Eniten tietoja saa ihmisten henkilökohtaisesta toiminnasta, mutta niissäkin näkyy vain kokonaismäärät eikä tietoja saa tietyltä aikaväliltä. On kuitenkin tarvetta saada tietoa siitä, miten esineet liikkuvat ja mitä määriä niitä pyydetään milloinkin. Ohjelmistoon merkitään aikaleimattuja lokitietoja, jolloin toiminnallisuus pitäisi olla mahdollista toteuttaa.

Jotta ominaisuuteen saataisiin kaikki olennaiset toiminnot, tuli asiaa tutkia. Tätä asiaa tutkittiin haastatteleamalla henkilöä, joka on pitänyt kirjaa yksikkömme tilastoinnista. Hän on henkilö, joka laskee, säilyttää ja toimittaa tilastotiedot eteenpäin. Haastattelussa kysyttiin, että mitä tilastoja edellisvuosina on pyydetty ja mitä tilastoja olisi tarpeen saada ohjelmistosta ulos.

Haastateltava lähti miettimään asiaa ensin sen kautta, että ei viitsi pyytää kovin monimutkaista muutosta. Hän myös mietti, että onko toimintoa mahdollista tehdä. Lisäksi hän totesi, että voimme kerätä tätä tietoa lisäksi itsekin. Haastateltavalle kerrottiin, että tällä hetkellä tärkeintä on kartoittaa, mitä tarvitsemme ja myöhemmin selviää, mitä voidaan toteuttaa. Tässä kyselyssä oli kuitenkin tarkoituksena kartoittaa tilastotoiminnon ihanteelliset ominaisuudet.

Haastattelussa selvisi, että edellisvuosina on laskettu, paljonko esineitä on saapunut pääasialliseen säilytystilaamme ja paljonko esineitä on laitettu hyllyyn. Haastattelun aikana selvisi, että eri vuosina on saatettu haluta eri lukuja. Ei ole ollut selkeää linjaa siitä, mitä tietoja milloinkin halutaan. Lukuja tarkastellaan usein vuoden vaihteessa, kun tutkitaan vuoden työsuorituksia. Niitä voidaan haluta myös silloin, jos pitää tehdä päätöksiä koskien resursseja tai työnjärjestelyä. Tulimme tämän takia siihen lopputulokseen, että parempi miettiä laajemmin, mitä tilastoja voitaisiin tulevaisuudessa ohjelmasta haluta ulos.

Mietimme tilastoja sen kannalta, että mitkä toiminnot ovat merkittävä osa työtämme ja mistä toiminnoista on tärkeä saada tietoa. Mietimme, mitkä ovat ne työt, joihin menee paljon aikaa ja joihin liittyvät luvut kertovat siitä, mihin yksikössämme työskentelevien työaika menee.

Lopulta päädyimme siihen, että ohjelmistosta olisi hyvä saada tietoa käsiteltyjen esinemäärien muodossa rajatuilla aikaväleillä. Tietoja haluttiin sekä kaikkien toimijoiden esinemääristä, että henkilöiden käsitlemistä.

8.5 Selvitys kuvien siirtymisestä kahden järjestelmän välillä

Museovirastolla on käytössä kokoelmanhallintajärjestelmä, johon on talletettu esineiden tiedot ja kuvat. Suurimmalta osalta esineiltä kuitenkin puuttuu kuva. Nyt meneillään olevassa siirtoprojektissa kaikki esineet pyritään valokuvaamaan. Kun esineitä joka tapauksessa käsitellään, niin ne ovat samalla helppo kuvata. Kuvat liitetään Muskatissa esineiden tietoihin. Kuvien avulla näkee, minkälainen esine on ja se on helpompi löytää. Nämä kuvat haluttaisiin siirtää automaattisesti kokoelmanhallintajärjestelmään, jonka kautta ne saataisiin näkymään myös Finna-hakutuloksissa.

Kuvien siirron voi tehdä käsin, mutta se ei ole käytännössä kannattavaa. Prosessi veisi näin tehtynä todella paljon työaika. Kuvat olisi ladattava yksitellen toisesta järjestelmästä ja liitettävä käsin metatietoineen toiseen järjestelmään.

Kokoelmanhallintajärjestelmässä on kuitenkin tarkat käytännöt, joilla kuvat on sinne tuotava. Kuvista on lisättävä oikeat tiedot oikeisiin kenttiin. Automaatiotoiminto tulisi suunnitella tarkasti, jotta kaikki tarvittavat tiedot saadaan siirtymään oikein. Selvitystyö aloitettiin keskustelemalla sellaisen työntekijän kanssa, joka on läheisesti tekemisissä kokoelmanhallintajärjestelmän kanssa. Hän oli nähnyt Muskattiin ottamiamme kuvia ja todennut ne laadultaan niin hyviksi, että ne voitaisiin viedä kokoelmanhallintajärjestelmään. Hän oli myös yksi niistä henkilöistä, jotka toivoivat kyseistä ominaisuutta.

Aluksi oli tarkoitus, että tutkimusyksikkö ja kokoelmanhallintajärjestelmän tuntevat henkilöt koostaisivat taulukon, jossa olisi säännöt järjestelmän kenttien täyttämiseen. Heillä ei kuitenkaan ollut syvällistä käsitystä Muskatin toiminnasta. Lopulta päädyttiin siihen, että molempia järjestelmiä ymmärtävien henkilöiden kesken pidettiin kokous, jossa käytiin läpi järjestelmien yhteistoiminta. Kokouksessa suunniteltiin kuvien siirtymisen toteuttaminen ja pohdittiin mahdolliset riskit siihen liittyen.

Yksi havaitsemamme ongelma oli, että logistiikkajärjestelmän kaikkiin kuviin ei ole merkitty kuvanottajaa. Tämä oli tieto, joka haluttiin kokoelmanhallintajärjestelmään. Muskattissa ei ole sen alkuaikana lisätty tietoa siitä, kuka lisäsi kuvan. Ei voida aukottomasti sanoa, että ohjelmistoon sisäänkirjautunut käyttäjä olisi aina se henkilö, joka on ottanut kuvan. Tämä johtuu siitä, että työntekijät lainaavat työvälineitään ja osa kuvista on ladattu muista lähteistä. Järjestelmään kirjataan ainoastaan tieto, kenen tunnuksilla kirjautunut käyttäjä on lisännyt kuvan. Tässä kohtaa päädyttiin siihen, että kuviin lisätään vain tieto, että ne on tuotu toisesta järjestelmästä. Muuten kaikki kuvaajatiedot eivät pitäisi paikkaansa. Kuvien oikeudet ovat käytännössä aina kuvaajalla, vaikka kuvat olisi otettu työsuhteessa. Tässä yhteydessä voidaan kuitenkin katsoa, että kuvien ottaminen kuuluu osana käsittelyprosessia ja kuvien oikeudet kuuluvat työnantajalle ja työnantaja saa esitellä kuvia haluamallaan tavalla. Kuviin voidaan liittää lisenssi, joka rajoittaa kuvien käyttöä kolmansien osapuolien toimesta. Tällöin voidaan rajoittaa kuvien vapaata käyttöä, jos ne julkaistaan avoimissa portaaleissa kuten Finnassa.

Logistiikkajärjestelmään tuodaan jatkuvasti kuvia kokoelmanhallintajärjestelmästä. Ohjelmiston pitäisi tarkistaa, että onko kuva tuotu aikaisemmin, jotta samoja kuvia ei tulisi moneen kertaan. Tämä ei kuitenkaan toimi ja nyt kuvat siirtyvät aina uudestaan. Muskatin tietokantaa ja dokumentointia tutkimalla syy löytyi. Kuvatoiminnon suunnittelussa määriteltiin nyt myös tämän toiminnon muutos, jotta tämä vika saataisiin korjattua.

Vaikka aluksi toive oli, että kaikki kuvat siirtyisivät automaattisesti, niin tässä kuitenkin todettiin riski. Kaikki kuvat eivät ole välttämättä tasalaatuisia ja

kokoelmanhallintajärjestelmään halutaan siirtää hyvälaatuisia kuvia. On myös riski, että automaatio ei toimi oikein ja aiheuttaa ongelmia järjestelmien välillä. Kuvien suuri koko ja iso määrä aiheuttaisi myös suurta liikennemäärää järjestelmien palvelinten välillä ja voisi hidastaa niiden muuta käyttöä. Näiden syiden takia päädyttiin siihen, että kuvien siirto tulisi tapahtua käyttäjän harkinnan mukaan. Tähän kohtaan määriteltiin, että Muskatin esi-
neen sivulle tulisi tehdä painike, jota painamalla saa siirrettyä esineen kuvat kokoelman-
hallintajärjestelmään. Lisäksi määriteltiin, että tarvitaan myös massatyökalu isompien
määrien siirtämiseen esimerkiksi niin, että korissa olevien esineiden kuvat siirtyvät painik-
keen avulla.

Kokouksen lopputuloksena saatiin aikaan toiminnon määrittely. Tutkimusyksikkö toimitti
myös tarkan taulukon kuvien siirrettävistä tiedoista ja siitä, mihin kenttiin ne pitää toisessa
järjestelmässä asettaa.

8.6 Selvitys asiakaslaskutukseen liittyvän toiminnon korjauksesta.

Muskattiin on tallennettu kokoelmakeskuksen säilytyspalveluiden asiakkaiden esineiden
tiedot. Näiden tietojen perusteella asiakkaita laskutetaan. Laskutus tapahtuu sen perus-
teella paljonko tilaa esineet vievät.

Meillä on kuitenkin kahta erihintaista säilytystilaa. Hyllyille laitettavien esineiden säilytyk-
sen hinta lasketaan sen mukaan montako kuutiota ne vievät tilaa. Verkkoseinälle ripustet-
tavien taideteosten säilytyksen hinta lasketaan sen mukaan paljonko pinta-alaa ne vievät
verkkoseinältä. Nyt ohjelmistosta ei kuitenkaan saa ulos tietoa siitä, minkälaisessa tilassa
tai hyllyssä esineet ovat. Tällöin verkkoseinälle sijoitettujen teosten vievä tila pitää laskea
käsien. Ohjelmistoon oli toivottu muutosta, jolla nämä tiedot saataisiin helposti ja erillisinä
tietoina.



Kuva 3. Konservattori Ilona Jaaranen asettaa taideteosta verkkoseinälle.

Tämän ominaisuuden kohdalla kuitenkin päädyttiin siihen, että tätä ominaisuutta ei kehitetä tällä kertaa. Nykyisten asiakkaiden tilan tarve on tarkasti laskettu ja siihen ei oleteta suuria muutoksia. Tämän ominaisuuden tärkeimmiltä käyttäjiltä ei myöskään saatu tarvittavia yksityiskohtaisia tietoja tämän työn tekemisen aikana.

9 Muskatin kehityssuunnitelma

Muskatin kohdalla kehitettävien kohtien priorisointiin vaikutti eniten sen perimmäinen merkitys esineiden logistiikkajärjestelmänä. Jos logistiikka ohjelmalla ei voida hallinnoida logistiikkaa tehokkaasti ja luotettavasti, sen merkitys katoaa. Tämän takia korjaustoimenpiteissä priorisoitiin tiedon saatavuuteen ja eheyteen liittyviä toimenpiteitä.

Seuraavaksi priorisoinnissa painotettiin niitä toimenpiteitä, jotka koskevat järjestelmän eniten käytettyjä toimintoja. Tämä tieto tuli tehdystä kyselytutkimuksesta. Tässä korostettiin vielä niiden käyttäjäryhmien mielipiteitä, joiden työhön liittyvät toiminnot ovat eniten riippuvaisia järjestelmästä. Tähän vaikutti se, miten paljon tietyt käyttäjät käyttävät järjestelmää sekä miten olennaista järjestelmän toiminta on heidän työnsä kannalta.

Lisäksi priorisoitiin toimintoja, jotka vaikuttavat suoraan rahallisesti merkittävien projektien onnistumiseen. Museossa oleva kokoelmien siirtoprojekti ja kokoelmien saavutettavuuden lisääminen olivat näitä projekteja.

Kehityssuunnitelmasta tehtiin dokumentti, joka sisälsi tämänhetkiset Muskatin kehitystarpeet priorisoituna listana. Kehitystarpeiden lisäksi dokumentissa suositeltiin, että kehityksen ennakkoimenpiteenä järjestelmälle tehdään kehitysympäristö, testiautomaatio ja versionhallinta.

Kun alustava lista oli saatu valmiiksi, käytiin se läpi vielä kokouksessa, jossa oli Museoviraston ICT-päällikkö ja Muskattia eniten käyttävän yksikön esimiehet sekä pääkäyttäjät. Kokouksessa käytiin läpi yksittäiset korjauskohdat ja korjausten prioriteetti. Näihin tehtiin kokouksessa tarkennuksia ja muutoksia.

Kokouksessa todettiin, että voimme jättää pois muutamia muutoksia. Pois jätettävät muutokset koskivat toimintoja, jotka on mahdollista tehdä kokoelmanhallintajärjestelmän avulla tai joihin voimme vaikuttaa omilla työtavoilla. Pois jätettiin mm. toive, että ohjelmasta saisi tietoja ulos taulukkomuodossa, koska tämä onnistuu jo toisesta järjestelmästä. Ei ole taroituksen mukaista tehdä kahteen järjestelmään päällekkäisiä toimintoja. Kokouksessa myös tähdennettiin Muskatin merkitystä ja tehtiin tietoisia valintoja sen kehittämistä ennen kaikkea logistiikkajärjestelmänä.

Kokouksessa tuotiin esille myös sellaisia asioita, jotka liittyvät järjestelmän käyttöön ja vaikuttavat sen käyttäjäkokemukseen, mutta jotka eivät ole kehittäjän vaikutuksen alla. Varsinaisten kehitystoimien lisäksi kehoitettiin myös parantamaan käyttäjille tapahtuvaa

viestintää ja miettimään järjestelmän laajentamista muihin logistiikkaan liittyviin asioihin. Järjestelmän muutoksista ja sen käyttökatkoista ei aiemmin ole juuri kerrottu käyttäjille. Parannuksena kehoitettiin tekemään muutokset järjestelmällisesti ja viestimään niistä käyttäjille. Jos muutoksista ei viestitä, niin käyttäjät voivat olettaa niiden olevan järjestelmässä olevia häiriöitä. Tämä aiheuttaa turhia yhteydenottoja ohjelmiston pääkäyttäjille ja tekniseen tukeen.

Nyt Muskatti halutaan muuttaa sellaiseksi, että sinne saisi tuoda vain esineitä, jotka löytyvät kokoelmanhallintaohjelmasta ja ovat virallisia museoesineitä. Järjestelmän tulevaisuuden kehittämisen kannalta ehdotettiin, että järjestelmään tuotaisiin myös muita kuin virallisia kokoelmaesineitä. Kyseessä ovat sellaiset esineet, jotka eivät ole museoesineitä, mutta joiden sijainnilla on merkitystä. Nämä esineet voivat olla esimerkiksi esineiden kopioita, rekvisiittaesineitä tai esinetukia. Rekvisiittaesineet saattavat olla hyvin museoesineen kaltaisia, mutta niitä ei ole virallisesti liitetty kokoelmiin. Ne voivat olla silti osa näyttelyä ja ainakin Seurasaaressa näitä esineitä pidetään näyttelyissä. Niiden sijainnista olisi hyvä olla tieto. Esinetuet on usein tehty tietyille esineille. Niitä ei voida aina säilyttää esineiden yhteydessä, mutta ne olisi hyvä voida paikallistaa ja yhdistää esineeseen, jotta niitä ei tarvitse tehdä uudestaan.

Muutostarpeet koottiin PowerPoint-dokumenttiin tärkeysjärjestyksessä. Ne oli kuitenkin koottu kokonaisuuksittain, että kehittäjällä olisi mahdollisuus antaa erillinen tarjous eri kokonaisuuksista. Tämän listan lisäksi oli koostettu tarkentavia liitteitä. Liitteissä oli lokiongelmien kuvaukseen liittyviä bugiraportteja, jotta kyseisen toiminnon ongelmat olisi helppompaa paikallistaa ja korjata. Liitteissä oli myös tarkka kuvaus siitä, miten kuvien tulisi siirtyä logistiikkajärjestelmän ja kokoelmanhallintajärjestelmän välillä. Näiden lisäksi oli myös alustava testaussuunnitelma. Testaussuunnitelma oli antamassa kuvaa siitä, miten laaja ja työläs se olisi. Se jouduttiin kuitenkin jättämään vielä vajaaksi, koska sen täydentäminen vaati tiedon siitä, mitkä ominaisuudet korjataan.

9.1 Priorisointi

Ensimmäiset ja isoimmat muutokset koskivat kehityksen taustatöitä ja niiden tarkoitus on edesauttaa järjestelmän ylläpitoa ja jatkokehitystä. Ennen varsinaisia kehitystöitä järjestelmälle suositeltiin tehtävän kehitysympäristö, jossa on automatisoitu testaus ja versionhallinta.

Varsinaisen Muskatin käyttöön liittyvät korjaukset oli jaettu prioriteetiltaan kokonaisuuksiin:

- Lokiongelmien & tiedon oikeellisuus
- Muskatin ja kokoelmienhallintajärjestelmien yhteistoiminta.

- CSV-tiedostolla tietojen tuonti ja massatoiminnot
- Siirtotoiminto
- Pikkukorjaukset prioriteetti 1.
- Tilastotoiminnot
- Kori toiminnon kehittäminen
- Pikkukorjaukset prioriteetti 2.
- Pikkukorjaukset prioriteetti 3.

Varsinaisista kehityskohteista tärkeimpiä olivat tietojen eheyttä koskevat korjaukset. Kyseessä on logistiikkajärjestelmä, jonka ensisijainen tehtävä on säilyttää tietoa esineiden sijainnista ja jonka kautta hallitaan niiden liikkeitä. Järjestelmän tärkein käyttötarkoitus on, että sen avulla museoesineet voidaan paikallistaa helposti. Paikkatietoihin vaikuttavat tiedot ovat sen takia oltava oikein, ja niihin liittyvät toiminnot pitää toimia niin, että ne tukevat tietojen oikeellisuutta. Jos Muskatista ei saa oikeaa tietoa esineiden liikkeistä tai sijainneista, niin sillä ei ole käyttöä ja se hidastaa sen avulla tehtävää työtä.

Muskatista tehdyssä kyselyssä nousi voimakkaimmin esille järjestelmän toiminta kokoelmanhallintajärjestelmän kanssa. Näiden järjestelmien yhteistoimintaan liittyviä toimintoja tekivät kaikki järjestelmää aktiivisesti työssään käyttävät. Näiden järjestelmien yhteistoiminnassa olevat ongelmat sekä puutteet otettiin seuraavaksi prioriteetiksi. Tärkeän osan tässä osuudessa sai toiminto, jonka avulla ohjelmistossa olevat kuvat saadaan siirrettyä kokoelmienhallintajärjestelmään. Tätä työtä ei ole järkevää tehdä käsin, joten sen automatisointi on loogista. Sille on myös tarve, koska se parantaa merkittävästi kokoelmien saavutettavuutta sen kautta, että esineiden kuvat ovat saavutettavissa.

Seuraaviksi tärkeimmiksi kehityskohteiksi nousi sellaiset toiminnot, joiden avulla käsitellään isoja esinemääriä. Nämä toiminnot olivat käytössä useilla käyttäjillä. Käyttäjät olivat raportoineet useista puutteista ja ongelmista näiden toimintojen yhteydessä. Nämä toiminnot ovat sellaisia, että niillä on iso vaikutus työn sujuvuuteen ja nopeuteen. Kyseisten toimintojen kehittäminen nopeuttaa ennen kaikkea esineiden siirtoprojektin työtä. Tässä projektissa on tiukka aikaraja ja suurien esinemäärien nopea käsittely on tärkeää projektin onnistumisen kannalta. Projektin myöhästyminen aiheuttaa taloudellisia menetyksiä, joten siihen liittyvien toimintojen korjausta on priorisoitava.

Tämän jälkeen prioriteetissa tuli sarja pieniä korjauksia. Näillä korjauksilla oli tarkoitus parantaa käyttäjäkokemusta ja nopeuttaa tiettyjen toimintojen tekemistä. Tämän jälkeen tuli koritoiminnon kehittäminen. Koritoimintoa käytetään massatoimintojen yhteydessä ja sen kehittäminen auttaisi nopeuttamaan työtä. Korjausten ja isoimpien kehitystoimien jälkeen

tärkeimmäksi koettiin saada tilastotoimintoja. Näillä on merkitystä, kun seurataan yksikön ja yksilöiden työsuoritusta sekä suunnitellaan tulevia resursseja.

Lopuksi listattiin pieniä käyttäjäkokemusta parantavia muutoksia ja korjauksia. Nämä oli jaettu kahteen kokonaisuuteen. Nämä olivat lähinnä pieniä käyttäjien negatiivisesti koke-
mia asioita. Asiat olivat merkitykseltään melko pieniä ja koskivat usein vain pientä osaa käyttäjistä. Lisäksi ne olivat monesti sellaisia, jotka on mahdollista kiertää järjestelmässä.

9.2 Muutoksien toteutus

Muutokset määriteltiin aloitettavan järjestelmän perustan turvaavilla toimenpiteillä. Aiem-
min muutokset oli tehty suoraan Muskatin kooditiedostoihin palvelimelle. Järjestelmältä
puuttui versionhallinta. Tämä tarkoittaa, että aiempiin versioihin ei voi palata. Tehdyt muu-
tokset on myös aiemmin testattu tuotantojärjestelmässä. Nyt tähän haluttiin muutos ja en-
simmäinen kehityskohde on, että muutokset tulisi tehdä ja testata kehitysympäristössä,
jossa on testiautomaatio. Tämä sen takia, että käytössä olevan järjestelmän tiedot eivät
korruptoidu, jos muutos rikkoo jotain pahasti.

Samalla tulisi tehdä versionhallinta ja muutokset tulisi toteuttaa versioina. Tällöin pysytään
paremmin kärryillä siitä, mitä muutoksia on tehty ja milloin sekä voidaan tarvittaessa pa-
lata edelliseen versioon. Samalla muutokset voidaan tehdä selkeinä kokonaisuuksina ja
niistä voidaan viestiä käyttäjille ajoissa. Näiden muutoksien avulla on myös helpompi
hankkia ylläpitopalveluita järjestelmää tuntemattomilta tahoilta ja pysyä kärryillä siitä, mikä
versio ohjelmointikielestä on käytössä ja vaatiiko se mahdollisesti päivityksiä?

Näiden korjausten lisäksi tulisi varmistua järjestelmän tietojen säilymisestä. Varmuuskopi-
ointi tapa tulisi varmistaa ja tarvittaessa muuttaa niin, että varmuuskopioita on ainakin
kaksi kappaletta fyysisesti eri sijainnissa. Lisäksi varmuuskopion palautus tulisi testata
joko tietohallinnon tai kehittäjän toimesta. Jos palautusta ei testata, niin tällöin ei voida
varmasti sanoa, että palautus onnistuu.

Testausta varten tehtiin alustava testaussuunnitelma. Siihen sisällytettiin muutamia moni-
mutkaisia toimintaketjuja. Testauksessa haluttiin tarkistaa monta eri asiaa ja muutamalla
monimutkaisella testillä saadaan paljon tietoa. Testien määrä pyrittiin pitämään melko vä-
häisenä, jotta testiautomaatio ei muodostu taakaksi ylläpidon kannalta. Testauksen ulko-
puolella jouduttiin jättämään kokoelmanhallintajärjestelmään liittyvät toiminnot, koska testi-
järjestelmää ei ole mahdollista yhdistää kokoelmanhallintajärjestelmään.

Tärkeimmiksi kehityskohdiksi koettiin ne kohdat, jotka vaikuttavat tiedon eheyteen eli järjestelmän tietoturvaan. Tiedon eheys vaikuttaa siihen, miten järjestelmästä saatavaan tietoon voi luottaa ja miten oikein se näyttää esimerkiksi esineiden sijainnit. Korjausehdotuksissa pyydettiin estämään joitain toimintoja, jotka aiheuttavat usein virheitä. Nämä tilanteet olivat usein myös sellaisia, että oikeassa elämässä ne eivät olisi mahdollisia. Pyyntönä oli, että esineitä ei voisi siirtää tai sijoittaa muualle, jos ne ovat hyllyssä. Lisäksi pyydettiin, että yksittäistä esinettä ei voi sijoittaa hyllyyn tai tilaan ennen kuin se on poistettu laatikosta. Lisäksi pyydettiin, että ohjelma kertoo tai varoittaa käyttäjää kyseisen toiminnon aikana.

Muskatissa olevat lokitiedot ovat tärkeitä, kun jäljitetään kateissa olevia esineitä. Tämä koskee esineiden, tilojen, paikkojen ja hyllyjen tietoja. Ohjelmistossa on useita kohtia, joissa näihin liittyvät luvut tai lokitiedot eivät näy oikein. Osa näiden vikojen aiheuttajista, oli tiedossa, osa vioista kuvailtiin liitteessä olevilla bugiraporteille. Jo tiedossa olevat ongelmat pyydettiin korjaamaan niin, että järjestelmässä olevat lokitiedot muodostetaan eritavalla. Nyt ne kirjoitetaan merkkijonona, jossa on linkkejä. Nyt on tapauksia, joissa linkki on oikein, mutta teksti on väärin. Ehdotus olisi, että merkkijono muodostetaan niin, että laatikko, hylly ja esinenumerot laitetaan niiden id numeron avulla merkkijonoon. Tällöin niiden teksti vastaisi linkkiä. Esinenumero on viittaus tiettyyn fyysiseen objektiin ja lokitiedoissa on tässä järjestelmässä tarkoitus kuvailla fyysisen esineen liikkeitä. Silloin esineen numero on muututtava lokitiedossa, jos se muuttuu esineen tietueessa.

Muskatin ja kokoelmanhallintajärjestelmän välisessä toiminnassa muutoskohteet liittyivät tiedonsiirtoon niiden välillä. Ensinnäkin olisi tärkeää korjata viat, jotka näiden järjestelmien yhteistoiminnasta johtuu. Tärkein korjauskohde olisi estää ylimääräisten kuvien tuleminen kokoelmanhallintajärjestelmästä. Lisäksi järjestelmien välinen toiminta tapahtuu osittain ajastettuina tehtävinä ja ajastetut tehtävät ajettiin 5 minuutin välein. Tämä hidastaa osaa toiminnoista, kun tehtävien valmistumista joutuu odottamaan. Pyyntönä oli, että kehittäjä selvittäisi onnistuuko ajastettujen tehtävien ajo useammin kuin 5 minuutissa. Järjestelmään haluttiin lisäominaisuus, jonka avulla kuvat siirtyvät logistiikkajärjestelmästä kokoelmanhallintajärjestelmään. Tämä ominaisuus määriteltiin tarkasti ja kehitystarpeiden liitteenä toimitettiin taulukko, jossa on selvitys siitä, mitkä tiedot kuvista on siirrettävä.

Massatoiminnoissa ongelmia aiheutti muun muassa isojen esinemäärien tuominen CSV-tiedoston avulla. Tämän toiminnon kautta on havaittu ongelmia, jotka riippuvat siitä, kuinka monta esinettä tuodaan ja paljonko tekstiä on taulukon soluissa. Nyt on useita tilanteita, joissa taulukon lataamisen jälkeen ei voida mennä eteenpäin vaan taulukko pitää muokata ja ladata uudestaan. Tämä on aikaa vievää. Kehityspyynnöissä on pyyntö, että

vialliset ja tyhjäät lomakkeen solut voisi poistaa järjestelmässä ennen niiden lataamista tietokantaan. Tämä nopeuttaisi toimintaa huomattavasti. Samassa yhteydessä haetaan tietoja kokoelmanhallintajärjestelmästä ja nämä tiedot menevät välillä lomakkeessa väärin kenttiin. Tämä on korjattava käsin ja toiminnon korjaaminen säästäisi paljon aikaa ja vaivaa.

Siirron ja tilausten kohdalla pyydettiin korjauksia, jotka helpottaisivat työtä ja toimisivat paremmin nykyisten työtapojen kanssa yksiin. Siirtojen kohdalla toivottiin, että ne voisi vastaan ottaa väliaikaiseen tilaan, jos niitä ei voida käsitellä heti. Näin esineet olisivat helpommin paikannettavissa ja toiminto vastaisi oikeaa elämää. Tilauksien kohdalla pyydettiin, että tilaukset jakautuisivat automaattisesti paikan suhteen. Nyt tilauksissa voi olla sekaisin eri paikkojen esineitä. Tilauksien tullessa joutuu niitä keräilevä poimimaan oman kohteensa esineet ja pitämään huolta, että huomaa kaikki oman kohteensa tilaukset. Olisi parempi, jos tilaus olisi vain yhdestä paikasta. Tilaaja ei taas välttämättä tiedä, missä esineet ovat ja on helpompi heille tehdä yksi tilaus.

Tilastotoiminnoissa määriteltiin ne asiat, joista halutaan tietoa käsiteltyjen esinemäärien kautta. Tilastotoiminnon kehitykseen valitut esinemäärät olivat toiminnoista, joihin menee paljon aikaa ja joiden tilastoja on tärkeää esitellä ylemmille tahoille. Kehitystoiveissa määritellyt tilastotarpeet:

- Siirroissa siirtyneet esineet tietyllä aikavälillä tiettyjen sijaintien kesken.
- Tietylle asiakkaalle noudetut tai asiakkaan tilaamat esineet tietyllä aikavälillä.
- Tietyn asiakkaan hyllytetyt esineet tietyllä aikavälillä.
- Tietyissä paikassa tietyllä aikavälillä hyllytetyt ja hyllystä otetut esineet.
- Sijainneissa olevat esinemäärät reaaliaikaisen tiedon mukaan sekä aikaisempina ajankohtina (sekä hyllyssä olevat että tiloihin sijoitetut).
- Henkilökohtaiset tiedot tietyllä aikavälillä.

Näiden korjaustoimenpiteiden lisäksi toivottiin useita pieniä korjauksia. Muskatissa oli muutamia toimintoja, jotka olivat joko jääneet kesken tai eivät toimineet. Etusivulla oleva vapaahakukenttä antaa usein monisivuisen hakutulossivun, kyseisen sivun navigointipalkki ei toimi ja sivulla ei pääse siirtymään sivujen välillä. Tämän voi kiertää tekemällä hakuja esine- tai laatikkosivulla, mutta tämä on käyttäjistä ärsyttävää.

Muskatissa on mahdollista lisätä uusia tulostuskärryjä, mutta uudet kärryt eivät lisäännä automaattisesti kaikkiin valikkoihin. Tämä hankaloittaa toimintaa, koska uusien kärryjen kanssa toimivat henkilöt eivät voi valita niitä oletustulostuskärryyn. Tämä tarkoittaa,

että kärry on muistettava vaihtaa aina, kun kirjautuu järjestelmään sisään. Tämä toiminto aiheuttaa hukkatulosteita ja kärryjen lisääminen kaikkiin valikkoihin listattiin muutoksiin.

Järjestelmässä olevien kuvien kohdalla toivottiin mahdollisuutta, että niiden suuntaa voisi kääntää. Kuvat eivät tallennu järjestelmään samassa suunnassa kuin ne on otettu, vaan ne näytetään aina vaakasuuntaisina. Tämä vaikeuttaa hyvien kuvien ottamista, koska osa esineistä istuu paremmin pystykuvaan. Kun kuvan avaa järjestelmässä, niin tulee esille painikkeet, joista kuvan voi kääntää ja kääntösuunnan tallentaa. Nämä painikkeet eivät kuitenkaan tallenna kuvan suuntaa. Kuvan suunta on mahdollista kääntää käsittelemällä kuva kuvankäsittely ohjelmassa, mutta se on hyvin hidasta, jos kuvia otetaan paljon.

9.3 Ylläpito

Ylläpidolla tässä yhteydessä tarkoitetaan niitä toimia, joiden avulla tietojärjestelmä pidetään käyttäjien käytettävissä. Ylläpitoon kuuluu usein palvelinten päivittämistä, pieniä korjauksia järjestelmän koodiin tai kirjastoihin sekä laitteistoon liittyviä toimenpiteitä. Lisäksi ylläpitoon olisi hyvä sisällyttää varmuuskopiointi ja sen säännöllinen varmistus.

Ylläpito voidaan suunnitella hyvin monella eri tavalla. Jokaisessa tavassa on omat riskinsä ja niiden kustannuksissa on myös isoja eroja. Suurimmat asiat, jotka ylläpitoa hankkiessa tulisi määritellä ovat ylläpidon laajuus ja työnjako. Eli mitä ohjelmiston ylläpitoon kuuluu ja se kuka sen tekee? Tässä työssä tarkastellaan sitä, mitä riskejä tai etuja liittyy eri vaihtoehtoihin. Lisäksi pohditaan hieman sitä, mikä vaikutus on ylläpidon kokonaisvaltaisessa tai osittaisessa ulkoistamisessa.

Järjestelmien ylläpito voi tarkoittaa sitä, että tehdään minimi ohjelmiston käytölle tarvittavat toimenpiteet eli pidetään tarvittavat palvelimet käynnissä ja mahdollistetaan sen käyttö. Usein ylläpitoon kuitenkin liitetään myös järjestelmän tietoturvan ja toiminnan kanalta kriittisiä korjaustoimenpiteitä. Ylläpito voi olla hyvinkin kokonaisvaltaista, jolloin siihen voi kuulua järjestelmän kehitys ja sen käyttöön tarvittava laitteisto.

On mahdollista ulkoistaa sekä järjestelmän hallinta, kehitys ja siihen liittyvien laitteiden ylläpito. Tämä olisi hyvin kokonaisvaltaista ja vapauttaisi hankkijan omia resursseja järjestelmän ja laitteiston ylläpidosta. Tämä on kuitenkin kallista, koska kyseessä on kustomoitu ohjelmisto ja siihen on liitetty monenlaisia erilaisia laitteita. Fyysisiä laitteita on monessa eri paikassa ja on haastavaa, että ulkopuolinen toimija tulisi niitä huoltamaan tai asentamaan. Tämä vaatisi, että ulkopuolisille pitää antaa vapaa pääsy tiloihin, joissa ei normaalisti toimi ulkopuolisia.

Yksi vaihtoehto toteuttaa ylläpito on luovuttaa järjestelmä kokonaan ulkopuolisen yrityksen hallintaan ja kehitettäväksi, mutta pitää laitteisiin liittyvät resurssit itsellä. Tämä tarkoittaa, että ohjelmisto sijaitsee fyysisesti ulkopuolisen tahon palvelimella ja he hoitavat siihen liittyviä koodaus ja päivitys muutoksia. Tässä on se ongelma, että ohjelmisto on kuitenkin kehitetty Museovirastossa ja siihen liittyvät tarpeet liittyvät vahvasti museomaailmaan. Jos sen kehitys luovutetaan ulkopuolella voi se tarkoittaa sitä, että ulkopuolinen taho ei ymmärrä tarpeitamme oikein ja ei osaa huomioida niitä kehityksessä.

Kaikissa tapauksissa, joissa osa ylläpitoa ulkoistetaan, toistuu sama riski. Palveluun todennäköisesti sovitaan perusraamit ja usein siihen kuuluu lisäpalveluita, joille on lisähinta. Käyttäjien palvelupyynnöt voivat olla tällaisia. Muskatin käyttäjät eivät osaa päätellä sitä, mistä ohjelmiston ongelmat voisivat johtua. He saattavat luulla, että järjestelmä on rikki, jos heillä ei toimi internetyhteys. He eivät myöskään usein kuvaile ongelmiaan tarkasti ja tämä johtaa usein siihen, että ongelman syy on vaikea hahmottaa. Tällöin on riski, että käyttäjät tekevät turhia palvelupyynnöitä väärälle taholle. Tämä aiheuttaa lisätyötä ja voi nostaa ylläpidon hintaa.

Yksi kannalta otettava vaihtoehto on, että ylläpitoa ei ulkoisteta, vaan siihen tarvittavat resurssit pyritään hankkimaan talon sisälle. Tämä tarkoittaa käytännössä sitä, että palkataan henkilö, joka osaa tehdä Muskatin pienkehitystä sekä ymmärtää siihen liittyviä laitteita. Tai määritellään nämä henkilöt jo olemassa olevista resursseista ja heille annetaan työaikaa hoitaa näitä tehtäviä. Tässä on etuna se, että ulkopuoliselle ei tarvitse antaa pääsyä toimijan palvelimelle, järjestelmiin tai tiloihin. Lisäksi näille työntekijöille muodostuu hyvä käsitys järjestelmän käyttöympäristöistä ja muista järjestelmään liittyvistä ohjelmistoista. Tämä ei kuitenkaan ole todennäköisesti kokonaiskustannuksiltaan halvin vaihtoehto. Kuitenkin tähän vaihtoehtoon liittyviä resursseja on helpompi hallinnoida ja resursseista on mahdollista saada paras hyöty kuluihin nähden. Tämä johtuu siitä, että työntekijöillä on hyvä ymmärrys käyttöympäristöstä, jolloin he todennäköisesti ymmärtävät siellä olevia haasteita ja osaavat ennakoida tulevia kehitystarpeita. Tämä vähentää riskiä siihen, että tulee yllättäviä lisäkustannuksia sellaisista kehitystoimista, joita ei osattu ennakoida. Suurin riski on, että jos työhön hankitaan yksi henkilö, niin järjestelmän ymmärrys ja kehitys on riippuvainen tämän henkilön työmotivaatiosta ja henkilökohtaisesta osaamisesta. (Bocij, Greasley & Hickie 2015, 587-591.)

Ainoa asia, joka ohjelmistossa olisi hyvä ulkoistaa joka tapauksessa, on järjestelmän palvelinalusta. Nykyisillä pilvipalveluina palvelintilaa myyvillä tahoilla on hyviä, edullisia ja turvallisia ratkaisuja. Useimmat palveluntarjoajat tarjoavat skaalautuvaa palvelintilaa ja

hoitavat automaattisesti palvelinten tietoturvaa ja päivityksiä. Nykyaikaiset palvelinsalit ovat myös sähkönkulutuksen kannalta tehokkaita ja siten ympäristöystävällisiä.

Mihin vaihtoehtoon sitten päädytäänkään, niin on varmaa, että Muskatin kehitykseen joudutaan aina käyttämään rahan lisäksi Museoviraston sisäisiä resursseja, kuten työaika. Vaikka palvelu hankittaisiin ostopalveluna, niin sen hankinnan teko ja valvominen tulee viemään työaika. Hankinnassa on haasteita ja mikään vaihtoehto ei ratkaise kaikkia ongelmia tai riskejä. Tärkeintä olisi kuitenkin se, että ylläpidon hankinnan suunnittelijalla olisi hyvä käsitys järjestelmään liittyvistä tarpeista. Lisäksi on tärkeää, että toimittaja ymmärtää järjestelmän erikoisluonteet ja heidän kanssaan on mahdollista viestiä tulevista tarpeista. (Talentum 2005, 21, 42.)

10 Johtopäätökset

Tämän työn aikana tuli esille se, että on tärkeä ymmärtää, mitä tietojärjestelmällä halutaan tehdä ja miten? Minkälaisia tehtäviä käyttäjät pyrkivät tekemään ja miten he järjestelmää käyttävät? Kun näihin kysymyksiin saa vastauksen, on käyttäjäkokemuksen kehittäminen huomattavasti helpompaa. Nämä asiat auttavat myös järjestelmien muutoksien lisäksi siinä vaiheessa, kun järjestelmää ollaan suunnittelemassa.

Parannuksia voidaan löytää tarkastelemalla käyttäjäkokemusta. Työssä käytettävien järjestelmien käyttäjäkokemuksen parantuminen parantaa henkilökunnan työmotivaatiota, turvallisuutta ja vaikuttaa positiivisesti työn sujuvuuteen. Käyttäjäkokemusta tulisi kuitenkin tarkastella kokonaisvaltaisesti. Jos vain yhden ryhmän tarpeet huomioidaan, ei välttämättä löydetä niitä muutoskohtia, jotka ovat olennaisimpia. Jos asiaa lähestytään vain yhden käyttäjän tai käyttäjäryhmän tarpeista, voidaan pahimmillaan saada aikaan isoa haittaa toisille käyttäjille.

Tietojärjestelmien hankinnassa ei tulisi miettiä pelkästään nykyhetkeä. Tietojärjestelmät eivät voi pysyä täysin samanlaisina. Niiden käyttöympäristöt muuttuvat ja kehittyvät jatkuvasti. Jos järjestelmiä ei päivitetä, niin ensin niiden tietoturva alkaa heikentyä ja lopulta järjestelmät eivät enää toimi uusissa selaimissa tai laitteissa. Tämän lisäksi järjestelmien käyttö ja niiden käyttäjien tarpeet voivat vaihdella eri aikoina. Kun järjestelmien kehityksen resurssit kohdistetaan juuri oikeisiin järjestelmiin ja niissä oikeisiin kohtiin, niin saavutetaan usein merkittäviä taloudellisia säästöjä.

Tässä työssä tutkittiin melko yksinkertaisen järjestelmän kehitystä. Tuloksena saatiin kuitenkin pitkä lista parannustarpeista. Työn tekemisen jälkeen löytyi kuitenkin vielä parannuskohtia ja virheellisiä toimintoja. Lopulta täysin kaiken kattavaa ja täydellisesti kokonaisuuteen sopivaa listausta on melko mahdoton tehdä. Tärkeintä on kuitenkin löytää isoimmat korjausta vaativat kohteet (Budi & Nielsen 2013, 7). Ne korjaamalla saadaan jo huomattavia etuja aikaan.

Tässä työssä tuli esille se, että käyttäjäkokemuksen tutkimisessa on hyvä käyttää useampia erilaisia tutkimustapoja. Jokaisella tavalla voidaan löytää erilaisia epäkohtia. Tutkimustavat on myös valittava kehitettävän asian mukaan. Kyselytutkimusta tehdessä tuli selvästi esille se, että siinä äänensä saa kaikista aktiivisin ja motivoitunein joukko. Heille kehitettävä asia on tärkeä ja he viitsivät käyttää aikaansa vastaamiseen. Kyselytutkimusten tuloksia tulisi aina tutkia kriittisesti ja miettiä kysymysten asettelut oikein. Muuten saatetaan saada sellaisia tuloksia, joista on vaikea tehdä johtopäätöksiä. Kyselyissä on hyvä myös

antaa tilaa avoimille vastauksille, koska sieltä voi löytyä yllättäviä huomioita, jotka muuten jäisivät sanomatta.

Huomattiin myös se, että vaikka käyttäjien kuunteleminen on tärkeää, niin vaaditaan asiantuntemusta löytää kaikista tärkeimmät kehityskohdat. Käyttäjät usein priorisoivat omasta näkökulmastaan tärkeitä asioita, eikä esimerkiksi sellaisia kohteita, jotka ovatkin kokonaisuuden kannalta olennaisimpia. Kehityskohteita priorisoitaessa pitää aina pitää mielessä kokonaisuus eikä kuunnella pelkästään äänekkäimpiä tahoja. Myös sen hiljaisimman puurtajan pitää saada työnsä hoidettua.

Käyttäjät eivät myöskään ole aina parhaita kuvailemaan tarpeitaan. He puhuvat usein omasta näkökulmastaan ja unohtavat, että heidän työssään on paljon sellaista, joka ei ole ulkopuoliselle itsestään selvää. He eivät myöskään usein ymmärrä tietotekniikkaa. Monesti tietoteknisesti perustaidot osaavan on vaikea antaa kehittäjälle selkeää kuvaa siitä, miten hän haluaa tietyt korjaukset. Tässä on hyvä olla mukana henkilöitä, joilla on kokomusta järjestelmäkehityksestä sekä käyttäjien ymmärtämisestä.

Tämän työn tekemisen aikana tuli esille se, miten tärkeää on järjestelmien dokumentointi. Dokumentointi auttaa järjestelmien ymmärtämisessä ja on hyödyllistä varsinkin silloin, kun järjestelmän hallinta siirtyy henkilöltä tai taholta toiselle. Tämän järjestelmän kehityksen kanssa suurimmat haasteet olivat siinä, että järjestelmää ei ole dokumentoitu juuri ollenkaan. Tosin dokumentointi on tehty hyvin pintapuolisesti ja dokumentointia tulisi jatkaa ja syventää, jotta järjestelmän kehitystä on helpompi hallita tulevaisuudessa. Jos ulkopuolinen kehittäjä tulee tekemään kehittämistä, niin tulisi palveluun sisällyttää koodin dokumentaatio. Tämän lisäksi talon sisällä tulisi tehdä käyttäjätapauksia, joilla kehittäjille selvennetään järjestelmän avulla tehtävää työtä. Käyttäjätapaukset olisivat olleet hyödyksi myös tämän työn tekemisessä. Tosin nämä kaikki vaativat aina päivityksiä silloin, kun järjestelmään tehdään muutoksia.

Tämän työn kautta tuli hyvin esille se, miten paljon esityötä ulkoistettujen toimintojen hankinnat teettävät. Ulkoistamisessa ei aina muisteta huomioida niitä kustannuksia, joita tulee hankkijataholle hankinnan suunnittelusta (Haverinen 2016, 38). Ensin on tiedettävä, mitä hankitaan ja sen jälkeen määriteltävä, mitä hankintaan tulee kuulua ja mitä voidaan jättää pois? Tämä työ on kuitenkin pakko tehdä hyvin, jotta hankinnasta saadaan hyötyä. On myös tärkeää, että hankinnassa on mukana ihmisiä, joita hankinta koskee ja tämän lisäksi sellaisia ihmisiä, joilla on ymmärrystä hankinnan kohteesta. Tällä kertaa kehittäjä ei edes pystynyt antamaan tarjousta ennen kuin sai tarkennuksia muutostarpeisiin.

Loppujen lopuksi tärkein, mitä tästä työstä tuli esille oli, että tietojärjestelmien järjestelmällisellä suunnitelmallisella kehittämisellä ja ylläpidolla saavutetaan useita hyötyjä. Tämä alkaa jo järjestelmän suunnitteluvaiheessa. Ennen kaikkea on olennaista ymmärtää, mitä järjestelmällä halutaan tehdä ja miten? Missä järjestelmää käytetään ja milloin se tulisi olla käytössä. Näiden lisäksi on hyvä myös huomioida käyttäjäryhmät ja heidän erityistarpeensa. Tämän lisäksi tulisi huomioida järjestelmien elinkaari ja miettiä, mitä tehdään siinä vaiheessa, kun järjestelmästä joudutaan luopumaan? Ja elinkaaren aikana on hyvä olla tietoinen, että järjestelmät vaativat jatkuvaa ylläpitoa ja huoltoa.

11 Pohdinta

Työn aloituksen yhteydessä määritellyistä riskeistä monikin toteutui työn aikana. Työn aloitus venyi, koska toimeksiantajalta kesti useita viikkoja tehdä päätös työn luvan antamisesta. Työtä ei ehkä koettu tarpeelliseksi tai sitten ihmisten työkiireiden takia asiaan liittyvien päätösten teko oli hidasta. Opinnäytetyötä ei voi kuitenkaan toteuttaa ilman toimeksiantajan hyväksyntää ja lupaa. Organisaatiossa oli kuitenkin muutamia aktiivisia toimijoita, joiden avulla työn merkitys ymmärrettiin ja työ pääsi alkamaan.

Työssä oli olennaista, että sen ympärillä tehtiin yhteistyötä useiden toimijoiden kesken. Tämän työnteon aikana siitä, keskusteltiin monien yksiköiden välillä. Vaikka se ei ollut työntarkoitus tai osa-alue, niin tämän työn kautta useat toisistaan erillään olevat toimijat tekivät yhteistyötä ja se edisti käyttöjärjestelmän ympärillä toimivien ihmisten välistä kommunikaatiota.

Kun työ aloitettiin, oli sillä jo kiire. Muutoksien teko haluttiin aloittaa mahdollisimman pian ja sitä varten tarvittiin kehittäjältä tarjous. Muutosten määrittely ja priorisointi haluttiin tätä varten saada mahdollisimman pian tehtyä. Tämä tarkoitti sitä, että työn teoriapohjaa ei ehditty viimeistellä ennen käytännön työn aloittamista. Tosin tämä aiheutti sen, että niiden työstämisen kautta tuli esiin myös mahdollisia asioita, joita työssä olisi voinut tehdä toisin. Työssä olisi voitu esimerkiksi käyttää automaattisia käyttäjäkokemuksen tutkimustyökaluja. Ne olisivat voineet paljastaa myös teknisestä ympäristöstä kehitettäviä asioita. Nyt työ keskittyi eniten käyttäjiin, heidän tarpeittensa kartoittamiseen, sanoittamiseen ja tarpeiden priorisointiin.

Järjestelmään tehtäviä muutoksia pohtiessa mietittiin myös ohjelman tietoturvaa. Suurimmat riskit tietoturvan kannalta havaittiin tiedon eheydessä ja saatavuudessa. Näihin liittyviä parannuksia kirjattiin kehityskohteisiin ja ne priorisoitiin tärkeiksi. Ohjelman todettiin olevan muilta osilta tarpeeksi tietoturvallinen. Tosin, jos järjestelmän käyttö muuttuu merkittävästi, tulisi tilanne harkita uudestaan. Järjestelmän käyttöä on joskus harkittu laajennettavan museon ulkopuolisiin tahoihin ja tässä tilanteessa tulisi ehdottomasti tutkia tietoturvatarpeet tarkasti.

Työssä tuli loppuvaiheessa selväksi se, että olisi ollut hyvä, jos työn alkuvaiheessa olisi tehty tavallisista ohjelmiston käyttötapauksista kuvaukset. Se olisi helpottanut korjausten prioriteetin havaitsemista. Niiden kautta olisi voinut lisätä ymmärrystä siitä, miten järjestelmään käytetään. Käyttötapaukset olisi hyvä tehdä, jos järjestelmän kehitys ja ylläpito

ulkoistetaan. Varsinkin, jos kyseessä on taho, joka ei tunne järjestelmän käyttöympäristöjä ja siihen liittyvää työtä.

Tässä työssä käsiteltiin hyvin vähän saavutettavuuteen liittyviä asioita. Käyttäjien suurin saavutettavuuteen liittyvä haaste on tietotekninen osaamisen puuttuminen. Viestinnän kehittämällä pyrittiin huomioimaan näitä haasteita. Ohjelmiston käyttäjillä on hyvin vähän sellaisia haasteita, joihin tarvittaisiin esimerkiksi ruudunlukuohjelmaa, joten tätä ei tarvinnut huomioida. Ohjelmisto ei ole myöskään yleisölle saatavilla, joten saavutettavuuteen liittyvä lainsäädäntö ei vaikuta siihen. Tosin tämä ei tarkoita, etteikö saavutettavuutta olisi olennaista miettiä myös organisaatioiden sisäisiä järjestelmiä kehitettäessä.

Tässä työssä jätettiin huomioimatta hankintaan liittyvät kestävän kehityksen seikat. Nykyään julkishallinnossa tehtävissä hankinnoissa pyritään yhä enemmän huomioimaan kestävän kehityksen elementtejä (Valtiokonttori 2020). Tässä hankinnassa kokoelmien saavutettavuus paranee ja sen kautta esineistä saadaan enemmän tietoa useammille ihmisille. Tämä jo itsessään edistää kestävää kehitystä. Yhtenä kriteerinä olisi voinut olla toimittajan työoloihin liittyvät vaatimukset. Näiden lisäksi myös ohjelmistokehityksen ympäristövaikutuksia voitaisiin pitää yhtenä hankinnan kriteerinä. Tällä hetkellä on tiedossa, että hyvin tehty ja tehokas koodi sekä optimoitu palvelinympäristö auttavat vähentämään ainakin virran kulutusta. Mutta tämän osa-alueen huomiointi on toistaiseksi saatavan tiedon valossa haastavaa ja vaatii vielä lisää tutkimusta, kuten Pirjo Hamari totesi Digitaalinen kulttuuri-perintömme – Kestävästi digitaaliseen tulevaisuuteen – webinaarissa.

Tämän työn valmistumisen aikaan kehitysprojekti on hankintavaiheessa. Järjestelmän ympärillä tapahtuva työ jatkuu ja hankinnan yksityiskohtia hiotaan. Jää nähtäväksi, miten hankinta lopulta onnistuu ja saavutetaanko sillä haluttuja parannuksia.

Lähteet

- Aalto, L. & Paavilainen, K. 14.05.2019. Saavutettavuus on suunnittelijan supervoima. Frantic. Luettavissa: <https://www.frantic.com/fi/blogi/saavutettavuus-on-suunnittelijan-supervoima>. Luettu: 19.1.2021
- Babich, N. 22.2.2019. 5 Simple Rules For Using Images More Effectively. UX planet. Luettavissa: <https://uxplanet.org/5-simple-rules-for-using-images-more-effectively-4dfd9352d3cb>. Luettu: 20.1.2021
- Bocij, P., Greasley, A. & Hickey, S. 2015. Business Information Systems: Technology, Development and Management for the E-Business. Pearson Education Limited. Harlow, United Kingdom.
- Budiu, R. & Nielsen, J. 2013. Mobile Usability. New Riders. Berkley, USA.
- Crispin, L. & Gregory, J. 2015. More Agile Testing. Pearson Education Inc. New Jersey, USA.
- Dasdan, A. 2018. How much time is left for coding? or How does a software engineer spend time at work? LinkedIn. Luettavissa: <https://www.linkedin.com/pulse/how-much-time-left-coding-does-software-engineer-spend-ali-dasdan/>. Luettu: 1.11.2020.
- Dufva, M. 29.7.2020. Teknologia tuo koronanälkeiseen aikaan paljon mahdollisuuksia – ja muutamia uhkia. Sitra. Luettavissa: <https://www.sitra.fi/artikkelit/teknologia-tuo-koronanälkeiseen-aikaan-paljon-mahdollisuuksia-ja-muutamia-uhkia/>. Luettu: 16.1.2020
- Grams, C. 2019. How Much Time Do Developers Spend Actually Writing Code? The New Stack. Luettavissa: <https://thenewstack.io/how-much-time-do-developers-spend-actually-writing-code/>. Luettu: 1.11.2020.
- Grant, W. 2018. 101 UX Principles. Packt Publishing. Birmingham, UK.
- Green, M. 2007. Change management masterclass: A step by step guide to successful change management. Kogan Page Limited. London, United Kingdom and Philadelphia, USA.

Groom, V. & Nomm, N. 2015. The Impact of Images: Research on Users Processing and Perception. User Experience Magazine. Luettavissa: <https://uxpamagazine.org/the-impact-of-images/>. Luettu 20.1.2021

Hamari, P. 2020. Museoviraston arkisto ja tietopalvelujen osastonjohtaja. Yhdessä kestävään tulevaisuuteen –paneeli. Digitaalinen kulttuuriperintömme – Kestävästi digitaaliseen tulevaisuuteen –webinaari 21.10.2020. Webinaarin esitykset luettavissa ja katsottavissa: <https://www.digime.fi/event/webinaari-digitaalinen-kulttuuriperintomme-kestavasti-digitaaliseen-tulevaisuuteen/>.

Haverinen, E. 2016. Hankintojen kilpailuttamisen kustannukset ja hankevalmistelu kunta-organisaatiossa. Oulun ammattikorkeakoulu. Luettavissa: https://www.theseus.fi/bitstream/handle/10024/116486/Haverinen_Elina.pdf?sequence=1. Luettu: 11.11.2020.

Hokkanen, J. 2018. Tietojärjestelmän kehittäminen palvelumuotoilun menetelmiä soveltaen. Luettavissa: https://www.theseus.fi/bitstream/handle/10024/144532/Hokkanen_Janne.pdf?sequence=1&isAllowed=y. Luettu: 15.6.2020.

Iny, C. s.a. The Best UX Tools For User Research And User Testing. Usabilitygeek. Luettavissa: <https://usabilitygeek.com/best-ux-tools-user-research-user-testing-2018/> Luettu: 19.1.2021

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

Kügelgen, M. & Laukkonen, V. 2021. Kaikki koodaa. Into Kustannus Oy. Helsinki.

Nielsen, J. 1993. Usability Engineering. Academic Press. San Diego, USA.

Nielsen, J. 24.12.2017. A 100-Year View of User Experience. Nielsen Norman Group. Luettavissa <https://www.nngroup.com/articles/100-years-ux/>. Luettu: 11.11.2020.

Piha, K. & Sutinen, M. 2020. Muutosvoimaa – Tutkimusmatka moderniin muutosjohtamiseen. Alma Talent Oy. Helsinki

Plugge, A. 2012. Managing change in outsourcing. Palgrave Macmillan. New York, USA.

Valtiokonttori 2020. Vastuullisuusraportointi yhtenäistyy valtiolla vuonna 2020. Luettavissa: <https://www.valtiokonttori.fi/uutinen/vastuullisuusraportointi-yhtenaistyy-valtiolla-vuonna-2020/#37c2852b>. Luettu 11.11.2020.

Liitteet

Liite 1. Luottamuksellinen.

Liite 2. Luottamuksellinen.

Liite 3. Luottamuksellinen.

Liite 4. Luottamuksellinen.

Liite 5. Luottamuksellinen.

Liite 6. Luottamuksellinen.

Liite 7. Luottamuksellinen.

Liite 8. Luottamuksellinen.