

Aalto-yliopisto
Teknillinen korkeakoulu
Informaatio- ja luonnontieteiden tiedekunta
Tietotekniikan tutkinto-ohjelma

Johannes Suanto

**Käytettävyyssongelmien synty
ohjelmistokehitysprosessissa,
Case: Liikenteen turvallisuusviraston VERO, REKI
ja PIIKO järjestelmät**

Diplomityö

Espoo 2. kesäkuuta 2010

Valvoja: Prof. TkT Marko Nieminen

Ohjaaja: Marko Myllyniemi, FM

Aalto-yliopisto Teknillinen korkeakoulu Informaatio- ja luonnontieteiden tiedekunta Tietotekniikan tutkinto-ohjelma		DIPLOMITYÖN TIIVISTELMÄ
Tekijä: Johannes Suanto		
Työn nimi: Käytettävyysohjelmien synty ohjelmistokehitysprosessissa, Case: Liikenteen turvallisuusviraston VERO, REKI ja PIIKO järjestelmät		
Sivumäärä: 8 + 90 + 4	Päiväys: 2.6.2010	Julkaisukieli: suomi
Professuuri: Käyttöliittymät ja käytettävyys		Professuurikoodi: T-121
Työn valvoja: Prof. TkT. Marko Nieminen		
Työn ohjaaja(t): FM Marko Myllyniemi		
<p>Tiivistelmä:</p> <p>Tietoisuus tietojärjestelmien käytettävyyden suomista eduista on viime aikoina levinnyt suurimpaan osaan järjestelmiä teettävistä tai tekevästä organisaatioista. Siksi onkin outoa miksi tiettyyn tehtävään ja tietylle käyttäjäryhmälle räätälöidyssä ohjelmistossa esiintyy edelleen suuria käytettävyysongelmia. Käytettävyyden huomioivia järjestelmäkehitysmalleja on saatavilla moneen eri tarkoitukseen, mutta tutkimuksia oikeiden projektien aikana syntyneistä käytettävyysongelmista ja näiden syistä ei juuri ole saatavilla.</p> <p>Tässä diplomityössä tutkin Liikenteen turvallisuusviraston VERO-, REKI- ja PIIKO- järjestelmiä löytääkseni niiden keskeiset käytettävyysongelmat sekä näiden taustalla olevat syyt. Tutustuin järjestelmiin ja näiden kehitykseen haastatteleamalla järjestelmien käyttäjiä sekä järjestelmien kehitykseen osallistuneita henkilöitä. Tutustuin myös järjestelmiin liittyvään dokumentaatioon. Kerätyn tiedon pohjalta muodostin mallin käytettävyysongelmien synnystä ja annoin Liikenteen turvallisuusvirastolle suosituksia joiden avulla käytettävyysongelmia voidaan tulevissa järjestelmissä vähentää. Malli on riittävän yleinen että sitä voidaan hyödyntää myös muissa järjestelmäkehitysohjelmilla.</p> <p>Tutkimuksen tuloksena voin todeta että suuri osa käytettävyysongelmista syntyy jo järjestelmän määrittelyvaiheessa. Järjestelmäkehitysprosessin aikana niiden havaitsemista ja korjaamista estää tai edistää useat eri tekijät. Näihin tekijöihin on mahdollista vaikuttaa organisaatio- ja työkuulttuurin, valitun järjestelmäkehitysmallin sekä käyttäjä- ja asiakaslähtöisyyden kautta.</p>		
Asiasanat: Käytettävyys, käytettävyysongelma , prosessikehitys, tapaustutkimus		

Aalto University School of Science and Technology Faculty of Information and Natural Sciences Degree programme of Computer Science and Engineering		ABSTRACT OF THE MASTER'S THESIS
Author: Johannes Suanto		
Title: The emergence of usability problems in the software development process, CASE The Finnish Transport Safety Agency's VERO, REKI and PIIKO systems		
Number of pages: 8 + 90 + 4	Date: 2.6.2010	Language: Finnish
Professorship: User interfaces and usability		Code: T-121
Supervisor: Prof. Marko Nieminen, Dr.Sc. (Tech)		
Instructor(s): Marko Myllyniemi, M.Sc.		
<p> Knowledge of the benefits gained from usable software has by now reached most organizations involved in commissioning or developing computer systems. Therefore it is somewhat surprising to find serious usability problems in computer systems that have been developed specifically for a known task and user group. There is an abundance of system development models, which cater to the development of usable computer systems. Unfortunately there is a lack of published results which report what usability problems resulted from a real-world development project, and what were the reasons behind these problems. </p> <p> In this Master's thesis I studied the Finnish Transport Safety Agency's VERO, REKI and PIIKO computer systems in order to find the systems' essential usability problems and their causes. I acquainted myself with the systems by interviewing the systems' users and people who took part in their development. I also used documentation related to the systems to gain information about them. From the assembled information I constructed a model of how usability problems emerge during a software development process. The model is general enough to be useful for other software development processes. I also gave the Finnish Transport Safety Agency some recommendations on how to avoid usability problems in future systems. </p> <p> As a result of this study I can state that a large portion of the essential usability problems encountered had their roots in the system specification stage of systems development. During the software development process there are several factors that hinder or promote the discovery and correction of usability problems. These factors can be influenced through organizational- and work culture, the chosen software development model, as well as through a user and customer centered attitude. </p>		
Keywords: Usability, usability problem, process development, case study		

Alkusanat

Tämä työ on ollut sekä vaativa että antoisa. Vaativa, koska kolmen järjestelmän käytettävyysongelmien tutkiminen on ollut ajoittain varsin haastavaa ja raskastakin salapoliisityötä. Antoisaa koska olen päässyt tutustumaan monen ihmisen työhön heitä haastatellessani ja koska olen vakaasti sitä mieltä että tämä työ voi omalta osaltaan auttaa eri organisaatioita tekemään ja teettämään käytettäviä järjestelmiä.

Haluaisinkin kiittää Liikenteen turvallisuusvirasto Trafia ja valvojaani Marko Myllyniemeä siitä että sain luvan tutkia heidän järjestelmiään täysin avoimesti. Kiitokset myös muille Trafilaisille työn aikaisista hyvistä vinkeistä, keskusteluista ja kommenteista.

Kiitokset myös ohjaajalleni Prof. Marko Niemiselle, joka tunnisti alustavista mietinnöistäni diplomityön potentiaalin ja rohkaisi minua tarttumaan tilaisuuteen. Hänelle myös syvä kiitos tämän työn aikaisesta ohjauksesta ja hyvien kysymysten esittämisestä.

Suuret kiitokset myös kaikille jotka ovat jaksaneet lukea ja kommentoida tätä työtä sen eri vaiheissa.

Lisäksi erittäin lämmin kiitos vaimolleni Harrietille, joka on jaksanut seurata, kannustaa ja tukea tätä prosessia. Tyttarellemme Susannalle myös lämmin kiitos, jaksat aina piristää päivääni.

Espoossa 2.6.2010

Johannes Suanto

Sisällysluettelo

1. Johdanto.....	1
2. Tutkimuksen kohde ja rajaus.....	3
2.1. Rajaukset.....	3
2.2. Ajoneuvoliikenteen tietojärjestelmä.....	4
2.3. Tutkimuksen kohteena olevat järjestelmät.....	4
2.3.1. VERO.....	5
2.3.2. REKI.....	5
2.3.3. PIIKO.....	6
2.4. Tutkimuksessa mukana olevat organisaatiot.....	6
2.4.1. Liikenteen Turvallisuusvirasto Trafi.....	7
2.4.2. Tieto Oyj.....	8
2.4.3. Logica Suomi Oy.....	8
3. Käytettävyys järjestelmäkehityksessä.....	9
3.1. Miksi käytettävyys on tärkeää.....	9
3.2. Käytettävyyden käsittely kirjallisuudessa.....	12
3.2.1. Perusteokset.....	12
3.2.2. Toimintaohjeet.....	13
3.2.3. Prosessikuvaukset.....	14
3.2.4. Käytettävyyden sisällyttäminen nykyiseen järjestelmäkehitykseen.....	15
3.3. Käytettävyyden huomioiminen tietojärjestelmien määrittelyissä.....	17
3.4. Kirjallisuuskatsauksen yhteenveto.....	18
4. Tutkimusmenetelmät ja tutkimuksen kulku.....	19
4.1. Tutkimuksessa käytetyt menetelmät.....	19
4.1.1. Artefakti-analyysi.....	19
4.1.2. Teemahaastattelu.....	20
4.1.3. Affiniteettidiagrammi.....	20
4.2. Tutkimuksen kohderyhmät.....	21
4.3. Tutkimuksen kulku.....	22
5. Järjestelmien keskeiset käytettävyysongelmat.....	26
5.1. VERO:n keskeiset käytettävyysongelmat.....	26
5.1.1. Listat.....	26
5.1.2. Tietojen synkronointi.....	28
5.1.3. Eräajot.....	29
5.1.4. Käyttöliittymän yleiset ongelmat.....	29
5.1.5. Muita VERO käyttäjien kokemia ongelmia.....	30
5.2. REKI:n keskeiset käytettävyysongelmat.....	31
5.2.1. Rekisteröintien korjaukset ja takautuvat muutokset.....	31
5.2.2. Tulostus.....	32
5.2.3. Käyttöliittymän yleiset käytettävyysongelmat.....	33
5.2.4. Muut REKI käyttäjien kokemat ongelmat.....	34
5.3. PIIKO:n keskeiset käytettävyysongelmat.....	35
5.3.1. Yleisen tason ongelmat.....	36
5.3.2. Listoille jäävät hakemukset.....	37
5.3.3. Korvaavat kortit.....	38
5.3.4. Yrityskortit.....	40
5.3.5. Laskut.....	41
5.3.6. Muut PIIKO- järjestelmän käytettävyysongelmat.....	42

5.3.7. Järjestelmän ulkopuoliset ongelmat.....	43
5.4. Järjestelmien yhteiset ongelmat.....	44
6. Käytettävyysohjelmien syyt.....	46
6.1. VEROn käytettävyysohjelmien synty.....	46
6.1.1. Pitkien listojen toiminta.....	49
6.1.2. Tietojen synkronointi-ongelmat.....	50
6.1.3. Eräajojen hitaus ja kaatuilu.....	52
6.1.4. VEROn käyttöliittymän käytettävyysohjelmien.....	53
6.1.5. Maksujärjestely- toiminnallisuus.....	53
6.1.6. Työtehtävien käyttäjäroolit.....	55
6.1.7. Yhteenveto VEROn ongelmien synnystä.....	56
6.2. REKIn käytettävyysohjelmien synty.....	57
6.2.1. Rekisteröintien korjaukset ja takautuvat muutokset.....	58
6.2.2. Näyttöjen välinen navigointi.....	59
6.2.3. Tulostuksen ja raporttien ongelmien synty.....	60
6.2.4. Yhteenveto REKIn ongelmien synnystä.....	61
6.3. PIIKOn käytettävyysohjelmien synty.....	62
6.3.1. Yleisen tason ongelmat.....	63
6.3.2. Listoilta jäävät hakemukset.....	63
6.3.3. Korvaavat kortit.....	64
6.3.4. Yrityskortit.....	65
6.3.5. Laskut.....	66
6.3.6. PIIKOn muut käytettävyysohjelmien.....	67
6.3.7. Yhteenveto PIIKOn käytettävyysohjelmien syistä.....	69
6.4. Yhteenveto järjestelmien ongelmien syistä.....	70
7. Johtopäätökset.....	73
7.1. Ongelmien syntymekanismi.....	73
7.2. Ongelmien havaitsemista ja korjaamista estävät tekijät.....	74
7.3. Ongelmien havaitsemista ja korjaamista edistävät tekijät.....	78
7.4. Ongelmien lähtökohdat.....	80
8. Suositukset.....	82
8.1. Lähtökohtien kehittäminen.....	82
8.2. Kilpailutuksen kehittäminen.....	84
8.3. Järjestelmäkehitysprosessin kehittäminen.....	84
8.4. Elinkaariajattelun kehittäminen.....	86
8.5. Organisaation oppimisen kehittäminen.....	87
9. Yhteenveto ja pohdintaa.....	88
9.1. Yhteenveto tutkimuksen tuloksista.....	88
9.2. Pohdintaa.....	90
Lähteet.....	i
LIITE 1 - haastattelurunko.....	iii

Käytetyt termit ja lyhenteet

Ake	Ajoneuvohallintokeskus, nykyään Liikenteen turvallisuusviraston tieliikennetoimiala
Ammattipätevyyskortti	Ammattiliikenteessä käytetty kortti josta käy ilmi kortin haltijan pätevyys erilaisten ajoneuvojen ja ajoneuvoyhdistelmien kuljettamiseen
Asiakas	Henkilö joka asioi jossain toimipisteessä siten että hänen asiansa käsittelyyn käytetään jotain tutkimuksen alla olevaa järjestelmää
ATJ	Ajoneuvoliikenteen tietojärjestelmä, Liikenteen turvallisuusviraston omistama järjestelmäkokonaisuus joka koostuu useasta tietojärjestelmästä
ENNI	Tietojärjestelmä ajoneuvojen ennakkorekisteröintiä varten, osa ATJtä
KATSA	Tietojärjestelmä ajoneuvojen katsastustietojen hallintaan, osa ATJtä
Kortti	Ammattipätevyys- tai piirturikortti
Käyttäjä	Henkilö jonka työhön kuuluu tutkittavana olevan järjestelmän käyttö
Navigointipolku	ATJ järjestelmien yläreunassa oleva hyperlinkkipolku, joka kuvaa mitä reittiä käyttäjä on päässyt nykyiseen tilanteeseen. Voidaan käyttää navigoimaan aiemmille näytöille
PALKO	Palvelujen kokonaisuudistushanke jonka alaisuudessa ATJ järjestelmät toteutettiin
PIIKO	Piirturikorttijärjestelmä. Osa ATJtä jossa hallinnoidaan piirturikortteja sekä ammattipätevyyskortteja
Piirturikortti	Ammattiliikenteen kuljettajien työaikojen valvontaan käytetty älykortti
REKI	Tietojärjestelmä ajoneuvojen rekisteritietojen hallintaan. Osa ATJtä
Sanomajono	Tiedonsiirtotekniikka, kuljettaa tietoa järjestelmästä toiseen ilman että järjestelmät ovat suorassa yhteydessä toisiinsa
Sanomajonon kuuntelija	Ohjelmistokomponentti joka tarkkailee tiettyä sanomajonoa ja lukee sieltä tietoa järjestelmään. On mahdollista kiinnittää useampi kuuntelija tarkkailemaan yhtä sanomajonoa esimerkiksi sinne tulevan suuren sanomamäärän vuoksi
Takautuva muutos	Muutos joka kohdistuu aiemmin tallennettuun tietoon. Voi aiheuttaa muutoksia myöhemmin käsiteltyihin tapahtumiin, usein tämän vuoksi ongelmallisia
Trafi	Liikenteen turvallisuusvirasto
VERO	Tietojärjestelmä ajoneuvoverojen hallintaan, käyttää ATJn järjestelmiä hyväkseen ajoneuvojen ja henkilöiden tietojen osalta

Kuvat ja taulukot

Kuva 1 VEROn ongelmat - listat.....	27
Kuva 2 VEROn ongelmat - yhteislasku.....	28
Kuva 3 VEROn ongelmat - eräajot.....	29
Kuva 4 VEROn ongelmat - käyttöliittymä	30
Kuva 5 REKIn ongelmat - korjaukset ja takautuvat muutokset	32
Kuva 6 REKIn ongelmat - tulostus.....	33
Kuva 7 REKIn ongelmat – käyttöliittymän yleiset ongelmat.....	33
Kuva 8 REKIn ongelmat - käyttäjäryhmiin liittyvät ongelmat	34
Kuva 9 REKIn ongelmat - muut.....	35
Kuva 10 PIIKOn yleisen tason ongelmat	36
Kuva 11 PIIKOn listoille jäävät hakemukset - ongelma	37
Kuva 12 PIIKOn korvaavien korttien ongelmat.....	39
Kuva 13 PIIKOn yrityskorttien ongelmat.....	40
Kuva 14 PIIKOn laskutuksen ongelmat	41
Kuva 15 PIIKOn muut käytettävyysongelmat.....	42
Kuva 16 PIIKOn järjestelmän ulkopuoliset ongelmat.....	43
Kuva 17 PIIKOn listoille jäävien syyt.....	63
Kuva 18 PIIKOn korvaavien korttien syyt	65
Kuva 19 PIIKOn yrityskortti ongelmien syyt.....	66
Kuva 20 PIIKOn laskujen ongelmien syyt	67
Kuva 21 PIIKOn muiden ongelmien syyt – osa 1	67
Kuva 22 PIIKOn muiden ongelmien syyt - osa 2.....	68
Kuva 23 Malli käytettävyysongelmien syntyprosessista.....	73
Taulukko 1 Hakutermien antamia tuloksia.....	9
Taulukko 2 Kohderyhmien haastattelut.....	21

1. Johdanto

Tietojärjestelmien kankeudesta ja sopimattomuudesta siihen työhön johon ne alun perin kehitettiin, on luultavasti ollut puhetta niin kauan kun tietojärjestelmiä on aktiivisesti käytetty. Aikojen mittaan tietojärjestelmien kehittäjiä on, vaihtelevalla menestyksellä, pyydetty parantamaan tuottamiensa järjestelmien käytettävyyttä. Käytettäessä valmiiksi tuotteistettuja järjestelmiä, voidaan ajatella että huono käytettävyys johtuu käyttäjäorganisaation rajallisesta mahdollisuudesta vaikuttaa järjestelmän kehitykseen. Mielenkiintoisempi kysymys on kuitenkin miksi myös tietyille organisaatiolle räätälöityjen järjestelmien käytettävyys on usein heikolla tasolla.

Työskennellessäni järjestelmävastaavana Liikenteen turvallisuusviraston edeltäjälle, Ajoneuvohallintokeskukselle, havaitsin että vaikka järjestelmät on tehty räätälityönä, eivät ne olleet kovin käyttäjäystävällisiä. Samoihin aikoihin satuin opinnoissani tutustumaan John R. Landryn artikkeliin ”*Can computing professionals be the unintentional architects of evil information systems?*” (Landry 2008), jossa hän pohtii eettisiä kysymyksiä joita nousee esiin siirrettäessä päätäntävaltaa käyttäjiltä tietojärjestelmille. Tämä on ongelmallista koska käyttäjillä ei yleensä ole mahdollisuutta valita mitä tietojärjestelmää he työssään käyttävät, vaan joutuvat useimmiten vain hyväksymään työvälineensä (Troutt 2007). Näissä tilanteissa järjestelmän sopivuus käyttäjien tekemään työhön on riippuvainen järjestelmien kehittäjistä, jotka usein luovat järjestelmät omien toiminta- ja ajatusmalliansa pohjalta (Robertson 2006). Näistä herätteistä lähtikin liikkeelle ajatus kartoittaa miten käytettävyysongelmat oikein syntyvät järjestelmäkehitysprosessissa. Tässä työssä pyrin selvittämään tätä kysymystä Liikenteen turvallisuusvirasto Trafi:lle räätälöityjen VERO-, REKI- ja PIIKO-järjestelmien osalta.

Tämän työn 2. luvussa esittelen tutkimuksen kohteena olevat tietojärjestelmät, järjestelmien käyttäjäryhmät ja tutkimukseen osallistuneet organisaatiot sekä käyn läpi tämän tutkimuksen rajaukset. Kolmannessa luvussa tarkastelen miten kirjallisuudessa käsitellään käytettävyyttä, varsinkin miten sitä käsitellään osana järjestelmäkehitysprosessia. Tässä luvussa tarkastelen myös miten kirjallisuudessa on käsitelty käytettävyyden huomioimista järjestelmäkehityksen määrittelyvaiheessa. Neljännessä luvussa esittelen varsinaisen tutkimuksen kulun ja perustelen käytetyt

tutkimusmenetelmät. Viidennessä luvussa esittelen tutkimuksen aikana VERO-, REKI- ja PIIKO- järjestelmistä löytämäni keskeiset käytettävyysongelmat. Luvussa 6 selvitän järjestelmistä löydettyjen käytettävyysongelmiin syntyä. Seitsemännessä luvussa tuon esiin mahdollisen mallin käytettävyysongelmiin syntyä kuvaamiseen edellisten lukujen tulosten perusteella. Kahdeksannessa luvussa annan Liikenteen turvallisuusvirastolle suosituksia, joiden avulla on mahdollista vähentää käytettävyysongelmiin esiintymistä tulevissa järjestelmäkehityksissä. Viimeisessä luvussa teen yhteenvedon tutkimuksesta sekä pohdin tutkimuksen kulkua ja sen aikana opittuja asioita. Luvussa arvioin myös tutkimuksen luotettavuutta ja tulosten käyttökelpoisuutta muissa yhteyksissä.

2. Tutkimuksen kohde ja rajaus

Tässä luvussa esittelen tutkimuksen kohteena olevat tietojärjestelmät ja niiden kehittämisessä mukana olleet organisaatiot. Luvun lopussa perustelen valitut tutkimusmenetelmät ja esittelen niiden toiminnot lyhyesti.

Tässä tutkimuksessa minulla on tarkoituksena selvittää mitä keskeisiä käytettävyysoongelmia tutkittavista järjestelmistä löytyy ja miten ne ovat syntyneet. Lisäksi tarkoitukseni on kehittää malli siitä miten käytettävyysongelmat pääsevät syntymään räätälöityihin tietojärjestelmiin.

Tutkimuskysymykset joihin etsin vastausta ovat siis:

1. Mitkä ovat tutkittavien järjestelmien keskeiset käytettävyysongelmat?
2. Miten nämä keskeiset käytettävyysongelmat ovat syntyneet järjestelmäkehityksen aikana?
3. Voidaanko ongelmien synnylle kehittää sopivaa mallia?

2.1. Rajaukset

Tässä työtä tehdessä tiedostan että Liikenteen turvallisuusvirastolla on tutkittavien järjestelmien lisäksi myös muita räätälöityjä järjestelmiä, mutta jätän ne tämän tutkimuksen ulkopuolelle. En myöskään tarkastele Liikenteen turvallisuusviraston tai järjestelmätoimittajien käyttämiä ohjelmistokehitysprosesseja syvällisesti, vaan tarkastelen niitä vain siltä osin kun ne liittyvät tutkittavien järjestelmien käytettävyysongelmien syntyyn.

Sellaiset organisaatiot jotka käyttävät tutkimuksen kohteena olevia tietojärjestelmiä, ovat mukana tässä tutkimuksessa pelkästään tietojärjestelmien käytön osalta. Näiden organisaatioiden prosesseja ja toimintatapoja ei tutkita muilta kuin tutkimuksena olevien tietojärjestelmien käytön osalta.

2.2. Ajoneuvoliikenteen tietojärjestelmä

Ajoneuvoliikenteen tietojärjestelmä (ATJ) on Liikenteen turvallisuusviraston ”Palvelujen kokonaisuudistus” (PALKO) hankkeen piirissä teettämä ja omistama järjestelmäkokonaisuus johon kuuluu yhdeksän operatiivista tietojärjestelmää sekä näiden lisäksi kokonaisuutta tukevia järjestelmiä. Kaikki ATJn alaisuudessa toimivat järjestelmät on pyritty luomaan saman tietojärjestelmäarkkitehtuurin mukaisesti.

Tämän tutkimuksen kannalta on tärkeää tuntea muutama koko ATJ kokonaisuuteen vaikuttava arkkitehtuuriratkaisu. Ensimmäisenä nostan esille arkkitehtuuriin kuuluvan tavan tietojen välittämiseen järjestelmien välillä sanomajonoja pitkin kulkevinä määrämuotoisina sanomina. Tämä tietojen siirtotapa eroaa monesta muusta järjestelmien välisestä tiedon siirtotavasta siinä että järjestelmät luovuttavat sanoman sanomajonojen kuljetettaviksi, eivätkä siten ole suorassa yhteydessä keskenään.

Toisena arkkitehtuuriratkaisuna nostan esiin järjestelmien verkkopohjaisuuden. Kaikkia ATJ järjestelmiä käytetään www-selaimen yli. Tämä tarkoittaa että käyttöliittymiä koskee samat rajoitukset kuin www-sivujakin. Näihin voidaan laskea rajoitettu mahdollisuus interaktion toteuttamiseen, sekä arkkitehtuurin iästä kertova lataus- ja prosessointiaikojen näkymättömyys käyttäjälle. Tätä viimeisintä ongelmaa on pyritty korvaamaan asettamalla oletuksena kaikkiin sovelluksiin kahden minuutin raja, jonka ylittyessä käyttäjän ja varsinaisen sovelluksen välissä oleva www-palvelin tuo käyttäjän näkyviin ”*timeout*” virheilmoituksen. Käyttäjäorganisaatioihin vaikuttava arkkitehtuuriratkaisu on ATJn rajoitettu selaintuki. Tällä hetkellä ATJ tukee vain Internet Explorer selaimen versioita 6 ja 7.

2.3. Tutkimuksen kohteena olevat järjestelmät

Tässä osiossa esittelen tutkimuksen kohteena olevat järjestelmät. Tutkimuksessa tarkastellaan kolmea Liikenteen Turvallisuusvirasto Trafín tietojärjestelmää. Kaikki kolme järjestelmää ovat osa Ajoneuvoliikenteen tietojärjestelmä- (ATJ) kokonaisuutta. Järjestelmät on toteutettu Trafín tilauksesta siten että joko Logica Suomi Oy tai Tieto Oy on toiminut järjestelmätoimittajana. Järjestelmät ovat kaikissa tapauksissa täysin räätälöityjä.

2.3.1. VERO

VERO järjestelmä on ajoneuvoverotuksen tietojärjestelmä jonka avulla hoidetaan ajoneuvoihin liittyvien veropäätösten tekeminen ja veron maksuunpaneminen, verovarojen kerääminen ja maksamattomien verojen perintä. Vero-järjestelmässä hoidetaan n. 2,5 miljoonan auton verotus ja niihin kohdistuvien maksujen perintä. Vero-käsittelyn aiheuttavia muutostapahtumia rekisteröinnin ja katsastuksen järjestelmistä tulee päivittäin noin 6000. VERO- järjestelmän lähdekoodi on kooltaan 18,4 megatavua, ja järjestelmällä on erilaisia näytöjä 78 kpl.

VERO järjestelmään sisältyy kuusi toiminnallista kokonaisuutta (alijärjestelmää). Nämä ovat Veropäätös-, Reskontra-, Sopimus-, Ajoneuvo- ja asiakas-alijärjestelmä, Asiakastulosteet-, Autovero- ja polttoainemaksu- sekä Raportointi—alijärjestelmät. Veropäätös-alijärjestelmässä syntyy vuosittain n. 6,7 miljoonaa maksuunpanoa, joista eräajot hoitavat automaattisesti n. 99 %. VERO- järjestelmän käyttäjät käsittelevät tapaukset, joiden maksuunpanoja järjestelmä ei pysty käsittelemään automaattisesti. Tapaukset joita järjestelmä ei pysty käsittelemään automaattisesti viedään järjestelmän sisäiseen tarkistuspiinon, josta käyttäjät ottavat tapaukset manuaaliseen käsittelyyn.

Vero-järjestelmän käyttäjät ovat pääasiassa Liikenteen turvallisuusviraston palveluksessa olevia verotuksen virkailijoita (n. 30) ja ulkoistetun puhelinpalvelun henkilöitä (n. 40). Tämän tutkimuksen piirissä keskityn VERO järjestelmän osalta tutkimaan Liikenteen turvallisuusviraston palveluksessa olevia VERO käyttäjiä.

2.3.2. REKI

REKI- järjestelmässä hallinnoidaan ajoneuvojen liikennekäyttöön liittyviä tietoja, kuten omistaja- ja haltijatietoja sekä rekisteritunnuksia. Järjestelmää käytetään pääasiassa suorakäyttöisesti, mutta myös taustalla toimivia eräajoja käytetään hyväksi. REKI- järjestelmä toteuttaa palveluna järjestelmän oman tulostuksen lisäksi Ajoneuvoliikenteen tietojärjestelmään (ATJ) kuuluville ENNI- ja KATSA-järjestelmille todistusten tulostuspalvelun. Tulostuksen ylläpito tehdään REKI- järjestelmän ylläpitoryhmässä. REKI- järjestelmän lähdekoodi on kooltaan noin 27 megatavua ja järjestelmällä on 97 erilaista näyttöä.

REKI- järjestelmän käyttäjäorganisaatioina toimivat Liikenteen turvallisuusviraston lisäksi muun muassa katsastustoimipaikat, vakuutusyhtiöt, autoliikkeet ja rahoitusyhtiöt, jotka ovat tehneet sopimuksen Liikenteen turvallisuusviraston kanssa asiakkaidensa ajoneuvojen rekisteröimisestä. REKI järjestelmän määrällisesti suurin käyttäjäryhmä on vakuutusyhtiöiden asiakaspalvelijat, jotka vastaavat ajoneuvojen rekisteröinnistä asiakkaidensa puolesta. Tässä tutkimuksessa tutkin lähemmin Liikenteen turvallisuusviraston sekä yhden vakuutusyhtiön REKI järjestelmän käyttäjiä.

2.3.3. PIIKO

Piirturikorttijärjestelmä PIIKO on tietojärjestelmä, jonka avulla Liikenteen turvallisuusviraston ulkoistamassa asiointiverkossa käsitellään ammattiliikenteen käyttämien piirturi- ja ammattipätevyyskorttien hakemuksia, tehdään näiden korttien tilauksia, seurataan korttien toimituksen etenemistä ja ylläpidetään asiakkaiden- sekä korttien tietoja. Lisäksi järjestelmän avulla vaihdetaan kortinhakijoiden sekä korttien tietoja muiden yhteistyössä toimivien maiden kanssa. Valvontaviranomaiset (poliisi ja työsuojeluviranomaiset) hyödyntävät lisäksi järjestelmän tietovarastoa omassa toiminnassaan. PIIKO- järjestelmän lähdekoodi on kooltaan noin 24 megatavua ja järjestelmällä on 69 eri näyttöä.

Korttihakemusten käsittely kattaa uudet kortit, korttien uusimisen, kadonneiden tai vioittuneiden korttien korvaamisen sekä ulkomaisten korttien vaihtamisen suomalaisiksi korteiksi. Järjestelmä auttaa hakemuksen rekisteröintivaiheessa käyttäjää tekemällä automaattisesti tarkistuksia toisista järjestelmistä ja hakemalla niistä tietoja päätöksenteon tueksi. Liikenteen turvallisuusvirasto on ulkoistanut piirturi- ja ammattipätevyyskorttien käsittelyn. Tällä hetkellä käsittely on Ajovarma Oy:n asiointiverkoston vastuulla. Tässä tutkimuksessa tutkin Liikenteen turvallisuusviraston asiantuntijoiden ja Ajovarman asiakaspalvelijoiden PIIKO- järjestelmän käyttöä.

2.4. Tutkimuksessa mukana olevat organisaatiot

Tässä osiossa kuvaan tutkittavien järjestelmien kehityksessä mukana olleet organisaatiot. Järjestelmien käyttäjäorganisaatioita ei tutkita muuten kuin niiltä osin kun heidän edustajansa käyttävät tutkimuksen kohteena olevia järjestelmiä. Liikenteen

turvallisuusviraston VERO järjestelmän järjestelmätoimittaja on Tieto Oyj. REKI ja PIIKO järjestelmienjärjestelmätoimittaja on Logica Suomi Oy.

2.4.1. Liikenteen Turvallisuusvirasto Trafi

Ajoneuvohallintokeskus (AKE) yhdistyi Ilmailuhallinnon, Merenkulkulaitoksen meriturvallisuustoimialan sekä Rautatieviraston kanssa Liikenteen turvallisuusvirasto Trafiksi (www.trafi.fi) 1.1.2010. Liikenteen turvallisuusvirasto Trafin tieliikennetoimiala vastaa pitkälti Ajoneuvohallintokeskukselle aikaisemmin kuuluneista tehtävistä. Tämän tutkimuksen kohteena olevat järjestelmät on räätälöity Liikenteen turvallisuusvirasto Trafin tieliikennetoimialan tarpeisiin ja ovat sen omistuksessa.

Liikenteen turvallisuusvirasto kuuluu liikenne- ja viestintäministeriön hallinnonalaan. Virasto toimii liikenteen turvallisuuteen liittyvien asioiden hallinto-, palvelu- ja informaatiokeskuksena Liikenteen Turvallisuusvirasto Trafin tieliikennetoimialan tehtäviin kuuluu muun muassa

- ajoneuvojen ja niiden osien hyväksyminen
- ajoneuvojen rekisteröinti ja vuotuinen verotus
- katsastustoiminnan valvonta
- kuljettajantutkintojen järjestäminen
- ajokorttien rekisteröinti
- ajoneuvoliikenteen tietopalvelu

Liikenteen turvallisuusviraston tieliikennetoimialan tärkeimmät päämäärät ovat ajoneuvoliikenteen turvallisuuden ja ympäristöystävällisyyden edistäminen sekä luotettavan tieliikenneajoneuvojen rekisteritiedon hallinta ja tarjoaminen erilaisiin tarpeisiin.

Liikenteen turvallisuusviraston toimintaan tarvittavat varat kerätään palvelujen käyttäjiltä. Palvelut on pääsääntöisesti hinnoiteltu kattamaan palveluiden tuottamisesta aiheutuneet kustannukset. Ainoastaan tietojen myynti on liiketaloudellisesti hinnoiteltua.

2.4.2. Tieto Oyj

Tieto Oyj on palveluyhtiö, joka tarjoaa tietotekniikka-, tuotekehitys- ja konsultointipalveluja (www.tieto.fi). Tieto Oyj tarjoaa tietotekniikka-, tuotekehitys- ja konsultointipalveluja. Tieto Oyj:n palveluksessa on maailmanlaajuisesti noin 17 000 henkilöä, ollen siten yksi suurimmista tietotekniikan palveluyrityksistä Pohjoismaissa. Yritys pyrkii vahvaan asiakaskeksisyyteen ja sähköisten palveluiden asiantuntemukseen, verrattuna kilpailijoihinsa.

Tieto Oyj:n päämarkkina-alue on Pohjois-Eurooppa, Saksa ja Venäjä, missä yritys keskittyy keskisuurten ja suurten organisaatioiden palvelemiseen. Yritys palvelee asiakkaitaan maailmanlaajuisesti tietoliikennealalla, metsä-, öljy- ja kaasuteollisuudessa sekä sähköisissä palveluissa. Tieto Oyj tekee yhteistyötä maailman johtavien yhtiöiden ja organisaatioiden kanssa.

2.4.3. Logica Suomi Oy

Logica on IT- alan palveluyritys, jonka palveluksessa on maailmanlaajuisesti 39 000 henkilöä, joista Suomessa Logica Suomi Oyn palveluksessa on noin 3 000.

Logica Suomi Oy tarjoaa konsultointipalvelua asiakkaiden toiminnan ja palveluiden kehittämiseen, kuten tietojärjestelmien suunnitteluun ja toteuttamiseen. Yritys myös integroi asiakkaiden eri tietojärjestelmiä ja toimii asiakkaiden ulkoistuskumppanina. Liikenteen turvallisuusvirasto onkin ulkoistanut Ajoneuvoliikenteen tietojärjestelmän ”helpdesk” toiminnon Logicalle. Yrityksen asiakkaina on eurooppalaisia yrityksiä ja julkishallintoa.

3. Käytettävyys järjestelmäkehityksessä

Tässä luvussa tutkin miten käytettävyys on huomioitu tietojärjestelmien kehitykseen liittyvässä kirjallisuudessa. Aluksi tarkastelen miksi tietojärjestelmien käytettävyyteen tulisi kiinnittää huomiota järjestelmäkehityksessä. Tämän jälkeen esittelen muutaman kirjallisuudessa käytetyn lähestymistavan käytettävyyden käsittelyyn. Lopuksi esittelen miten kirjallisuudessa käsitellään käytettävyyden huomioiminen jo järjestelmäkehityksen määrittelyvaiheessa.

Tiedonhaussa olen käyttänyt Aalto-yliopiston tarjoamaa Nelliportaali (www.nelliportaali.fi) palvelua, joka mahdollistaa haut useaan sähköiseen kirjastoon ja artikkelitietokantaan. Tehdessäni hakuja tätä kautta rajasin lähes välittömästi haut tieteellisiin artikkeleihin huomattavasti riippuen tarkoista hakusanoista (ks. Taulukko 1, alla). Tulosten manuaalinen läpikäynti sopivia artikkeleita etsiessä oli ajoittain varsin työlästä.

Taulukko 1 Hakutermien antamia tuloksia

Hakutermi	IEEE	EBSCO	ABI/INFORM
"source of usability problems"	0	0	12
"usability problems"	72	47	45
"usability TCO"	0	5811	0
"usability cost"	5	6	5
"user experience"	855	578	2

3.1. Miksi käytettävyys on tärkeää

Tietojärjestelmien käytön lisääntyessä, ja niiden ilmaantuessa yhä uusiin käyttötarkoituksiin, tulisi järjestelmien kehittäjien huomioida yhä laajeneva käyttäjäkunta. Väestön ikääntyessä olisi lisäksi suotavaa että työpaikoilla ja yhteiskunnassa muuten käytettävät tietojärjestelmät tukisivat järjestelmien helppoa oppimista ja käyttöä. Useilla aloilla onkin jo olemassa ohjeet tai suositukset

tietojärjestelmien käytöstä ja niiden käytettävyydestä. Esimerkiksi julkishallinnon alaa koskettaa Julkisen hallinnon tietohallinnon neuvottelukunnan (JUHTA) JHS-jaoston antamat suositukset. Näistä esimerkiksi ”JHS 129 Julkishallinnon verkkopalvelun suunnittelun ja toteuttamisen periaatteet” koskettaa viranomaisten verkkopalvelujen suunnittelua. Suosituksessa todetaan että ”*Palvelu täytyy suunnitella tukemaan käyttäjien tarpeita ja heidän lähtökohdistaan. Palvelua on kehitettävä perustuen todellisten käyttäjien kanssa toteutettuihin tutkimuksiin, esim. käytettävyydesteihin*” (JHS suositukset 2006). Tämän lisäksi Valtiovarainministeriön tiedotteesta (Valtiovarainministeriö 2010a) käy ilmi että Valtioneuvosto on 4.2.2010 linjannut tapoja, joilla julkishallinnon organisaatiot selviävät nykyisessä taloudellisessa tilanteessa. Näihin toimenpiteisiin kuuluu muun muassa ”*Prosessien yhtenäistäminen, yhtenäinen arkkitehtuuri, helppokäyttöiset järjestelmät ja yhteiset rajapinnat*” (Valtiovarainministeriö 2010a).

Käyttäjien kokeman käytön helppouden lisäksi tietojärjestelmien käytettävyydellä voidaan saavuttaa suoria kustannussäästöjä. Esimerkiksi George Donahue esittää että huomioimalla käytettävyys järjestelmäkehityksessä voidaan saada aikaan säästöjä ”*järjestelmäkehityksessä, järjestelmätuessa ja koulutuksessa, dokumentoinnissa ja ylläpidossa*” (Donahue 2001). Donahuen mukaan hyvä käytettävyys myös parantaa käyttäjien tehokkuutta, alentaen siten järjestelmällä tehtävän työn tapahtumakohtaisia kuluja. Tällä on suuria vaikutuksia Suomen julkishallinnolle, koska julkishallinnon organisaatioiden tulee seurata valtion tuottavuusohjelmaa (Valtiovarainministeriö 2010b). Valtion tuottavuusohjelman tavoitteena on että ”*Tuottavuutta on kyettävä parantamaan selvästi ja kattavasti kaikilla toimialoilla, mukaan lukien julkinen sektori.*” (Valtiovarainministeriö 2010b). Tuottavuuden parantaminen tarkoittaa että nykyinen työmäärä pyritään tuottamaan pienemmällä henkilöstömäärällä. Tavoitteen saavuttaminen on helpompaa mikäli työn suorituksessa käytetyt tietojärjestelmät tukevat tätä mahdollistamalla työn helpon oppimisen ja tehokkaan suorittamisen. Näihin tavoitteisiin pääsemiseksi tulee tietojärjestelmien käytettävyyteen kiinnittää huomiota.

Tietojärjestelmien kehittäjien on lisäksi hyvä tarkastella omaa toimintaansa suhteessa luomiensa järjestelmien käyttäjien toimintaan. Toni Robertson esittää (Robertson 2006) että järjestelmän suunnittelijat ja toteuttajat upottavat järjestelmään omat näkemyksensä ja tulkintansa järjestelmän ja sen taustalla olevien työprosessien toiminnasta. Koska

järjestelmän loppukäyttäjät lähestyvät järjestelmän käyttöä oman kokemuspohjansa kautta, on heillä myös omat näkemykset ja odotuksen järjestelmän ja siihen liittyvien työprosessien toiminnasta. Useimmiten havaitaan että suunnittelijoiden ja loppukäyttäjien näkemysten välillä on kuilu. Robertsonin mukaan tämä asettaa järjestelmän suunnittelijalle eettistä vastuuta, koska tämä muotoilee järjestelmää oman näkemyksensä mukaan. Mikäli loppukäyttäjää ei huomioida järjestelmän kehityksessä, voi lopputuloksena olla suunnittelijan ehdoilla tehty järjestelmä joka pahimmassa tapauksessa rajoittaa loppukäyttäjän toimintaa ja tehokkuutta.

Liikenteen turvallisuusvirastolle loppukäyttäjien huomioiminen on tärkeää julkishallintoa ohjaavien lakien ja suositusten lisäksi myös kustannussyistä. Suuri osa viraston toiminnasta on ulkoistettu sopimuskumppaneille, joiden toiminnasta virasto maksaa kumppaneille korvausta. Kumppaneilla on intressiä neuvotella korvaukset sellaiselle tasolle että ne vastaavat viraston ulkoistamien tehtävien toteutuskustannuksia. Mikäli kumppaneiden käyttämät järjestelmät ovat vaikeasti käytettäviä, on kumppaneilla ymmärrettävä taipumus huomioida tämä palvelunsa hinnoittelussa.

Kilpailutettaessa kumppaneita keskenään, on mahdollista että nykyiselle kumppanille koituu kohtuutonta etua saadusta kokemuksesta järjestelmän käyttämisessä, mikäli järjestelmä on vaikeasti käytettävä. Erityisen hankalaksi tilanne muodostuu, mikäli järjestelmän tehokas käyttö vaatii dokumentoimattomien oikopolkujen tai kiertoteiden käyttöä. Tällöin tiedossa on asiakkaiden kokeman palvelutason lasku, mikäli palvelu siirtyy toiselle kumppanille. Vaihtoehtona on nykyisen kumppanin sopimuksen automaattinen jatkaminen, mikä puolestaan sotii kilpailutuksen ideaa vasten.

Tietojärjestelmien käytettävyys on siis tärkeää käyttäjien tyytyväisyyden ja tehokkuuden kannalta. Lisäksi tietojärjestelmien hyvällä käytettävyydellä voidaan saavuttaa suoria kustannussäästöjä. Jotta tietojärjestelmäkehityksessä mukana olevat organisaatiot voisivat hyödyntää näitä käytettävyyden tuomia etuja, on niiden ensin ymmärrettävä mitä käytettävyydellä tarkoitetaan ja miten käytettävyys voidaan huomioida järjestelmäkehityksessä. Tässä voidaan käyttää apuna jo tehtyä tutkimusta julkaistun kirjallisuuden muodossa. Kirjallisuuteen tutustuttaessa on hyödyllistä ymmärtää minkälaisista näkökulmista eri kirjoittajat lähestyvät aihetta.

3.2. Käytettävyyden käsittely kirjallisuudessa

Kirjallisuudessa tietojärjestelmien käytettävyyttä käsitellään eri näkökulmasta. Tässä nostan lyhyesti esiin käytettävyyden perusteokset, käytettävyyteen liittyvät toimintaohjeet, käytettävyyden huomioivat järjestelmäkehitysprosessit sekä ohjeet käytettävyyden huomioimiseksi organisaation nykyisessä järjestelmäkehitysprosessissa.

3.2.1. Perusteokset

Käytettävyyden perusteoksissa esitellään useimmiten käytettävyyttä yleisellä tasolla. Nämä teokset toimivat joko oppikirjoina tai aiheeseen tutustuttavina opuksina. Yleensä niissä esitellään muutama yleinen malli käytettävyydestä sekä mahdollisesti myös jokin kirjoittajien suosima järjestelmäkehitysprosessimalli.

Esimerkkinä käytettävyyteen tutustuttavasta oppikirjasta voidaan pitää Leventhalin ja Barnesin ”Usability Engineering : Process, Products and Examples” kirjaa (Leventhal 2007). Kirja painottaa käytettävyyssuunnittelun prosessin oppimista, joten kirjassa esitellään muutama yleinen malli käytettävyydestä kirjoittajien oman mallin lisäksi. Kirjassa tuodaan myös esiin pari versiota ohjelmistokehitysprosessista (vesiputous, iteratiivinen). Kirjan fokus on siten, varsin ymmärrettävästi, käytettävyyden perustietojen ja –mallien esittelyssä suhteellisen yleisellä tasolla.

Käytettävyyden perusteoksiin voidaan myös lukea alan pioneerien kirjoittamat, käytettävyyteen tutustuttavat teokset. Tällainen on esimerkiksi yleisesti siteerattu Jakob Nielsenin kirja ”*Usability Engineering*” (Nielsen 1993.), jossa hän esittelee käytettävyyden perusteet ja esittelee miten käytettävyyttä voidaan mitata. Lisäksi perusteoksiin tulee lukea erilaiset viralliset standardit ja ohjeet, kuten esimerkiksi jotka määrittelevät mitä käytettävyydellä tarkoitetaan eri tilanteissa. Esimerkkinä standardista voidaan mainita jo pitkään voimassa ollut ”SFS-EN ISO 9241-11 Näyttöpäätteillä tehtävän toimistotyön ergonomiset vaatimukset. Osa 11: Käytettävyyden määrittely ja arviointi” (IS09241-11 2000).

3.2.2. Toimintaohjeet

Varsin monet julkaisut ja artikkelit esittelevät erilaisia tapoja suorittaa käytettävyyden huomioimiseen liittyviä yksittäisiä järjestelmäkehityksen tehtäviä. Nämä tehtävät voivat olla käytettävyydestä tai sen osa-alueet, käyttäjätietojen hankkiminen vaatimusmäärittelyä varten, tai muut käytettävyyden huomioimiseen liittyvät osa-alueet. Yhteistä näille julkaisuille on niiden tiukka fokus tarkastelun kohteena olevaan osatehtävään.

Ohjeita käytettävyyden testaukseen löytyy yleensä jo perusteoksista, kuten Leventhalin ja Barnesin (Leventhal 2007) sekä Nielsenin (Nielsen 1993.) kirjoista, mutta myös artikkeleista ja muista julkaisuista. Testausohjeet keskittyvät yleensä kirjoittajan kiinnostuksen kohteeseen, ja ovat siten näkökulmaltaan hieman rajoitettuja. Tästä huolimatta ne ovat erittäin hyödyllisiä omalla kohdealueellaan.

Esimerkkinä tällaisesta erikoistilanteeseen kehitetystä testausmenetelmästä voidaan pitää Leen ja Gricen kehittämää menetelmää (Kwang, Grice 2004) mobiilien PDA-laitteiden käytettävyyden tutkimiseen. Siinä pyritään huomioimaan mobiilin laitteen käytön erikoisuudet verrattuna normaaliin tietokonepohjaisen järjestelmän käyttöön. Menetelmässä käytetään kyselyiden, skenaariopohjaisten tehtävien ja heuristisen arvioinnin yhdistelmää.

Verkkosivujen käytettävyydestä on kehitetty useita eri variaatioita käytettävyydestä. Hyvänä esimerkkinä voidaan mainita Anandhan *et al* (Anandhan et al. 2006) kehittämä ”CARE”- metodi verkkosivujen käytettävyyden testaukseen. Kyseinen metodi pyrkii edulliseen, tarkkaan, luotettavaan ja tehokkaaseen (Cheap, Accurate, Reliable, Efficient) käyttäjätestaukseen valikoimalla kirjoittajien mielestä sopivimmat menetelmät verkkosivujen käytettävyyden arvioimiseen.

Sampsa Hyysalo puolestaan esittelee kirjassaan ”*Käyttäjätieto*” (Hyysalo 2006) miten käyttäjätietoa voidaan hankkia eri menetelmin ja miten kyseistä tietoa voidaan hyödyntää tuotekehityksessä. Hyysalo käsittelee käyttäjätiedon hankkimisen menetelmiä siten että menetelmien tehokas käyttö on käytännössäkin mahdollista. Lisäksi Hyysalo tiivistää menetelmien vahvuudet ja heikkoudet siten että sopivan menetelmän valinta on lukijalle helppoa.

3.2.3. Prosessikuvaukset

Prosessikuvauksilla tarkoitan tässä yhteydessä kirjoittajan kehittämää muunnosta olemassa olevaan järjestelmäkehitysmalliin, tai kirjoittajan ehdotusta täysin uudeksi järjestelmäkehitysmalliksi. Usein tällaiset prosessikuvaukset ovat muunnelmia olemassa olevista järjestelmäkehitysmalleista, tai sitten ne on kehitetty johonkin suppeampaan erityiskohteeseen, kuten www sivujen tai mobiililaitteiden käytettävyyden kehittämiseen.

Hyvänä esimerkkinä tiettyyn kontekstiin suunnitellusta käytettävyyden huomioivasta prosessimallista voidaan pitää Petra Pietiläisen diplomityötä vuodelta 2001 (Pietiläinen 2001). Pietiläinen esittelee tutkimuksessaan prosessimallin joka on tarkoitettu nimenomaan käytettävien verkkosovellusten kehittämiseen. Hänen mallinsa yhdistelee olemassa olevia prosessimalleja ja on suunniteltu huomioimaan teollisuudessa esiintyvät rajoitukset ajankäytön ja resurssien suhteen.

Toinen tapa käsitellä löytyy Eva Olssonin väitöskirjasta (Olsson 2004), jossa hän lähestyy käytettävyyden huomioimista järjestelmäkehityksessä varsin kokonaisvaltaisella tavalla. Hän esittää että käytettävyyden huomioimisessa tulisi ensisijaisesti huomioida mitä käyttäjät oikeasti tarvitsevat työssään, ja vasta tämän jälkeen tutkia erilaisia teknisiä ratkaisuja osana työn kehittämistä. Olssonin ajatusmaailman lähtökohtana on oletus siitä että käyttäjät ovat ammattitaitoisia ja keksivät jatkuvasti tapoja ohittaa ongelmia järjestelmissä voidakseen toimia tehokkaasti. Tämä tulisi Olssonin mukaan huomioida järjestelmäkehityksessä siten, että järjestelmäkehittäjät tutustuvat käyttäjien työhön ja työnkuvaan niin että ymmärtävät sen. Käyttäjien tutustuttaminen järjestelmäkehitykseen vaarantaa hänen mukaansa käyttäjien oman näkökulman, koska heillä on näissä tilanteissa taipumusta omaksua järjestelmäkehittäjien sanasto ja ajatusmallit. Olsson toteaa myös että *”työsuunnittelu, käyttöliittymät ja käytettävyys ovat punotuneet yhteen, eikä ole mahdollista ensin analysoida työtä, sitten suunnitella käyttöliittymää ja lopuksi lisätä käytettävyyttä”* (Olsson 2004). Hän yhdistää nämä ideat ja asenteet osallistavaksi kehitysprosessiksi joka todennetaan eri projekteissa, joissa Olsson on ollut mukana väitöskirjan ja siihen liittyvien artikkeleiden kautta.

3.2.4. Käytettävyyden sisällyttäminen nykyiseen järjestelmäkehitykseen

Yllä käsiteltyjen valmiiden prosessikuvausten lisäksi kirjallisuudessa esiintyy teoksia, joissa pyritään kuvaamaan miten käytettävyyden huomiointi saadaan sisällytettyä järjestelmiä kehittävän organisaation olemassa olevaan järjestelmäkehitysprosessiin. Näissä teoksissa pyritään tutustuttamaan organisaatio, sen päättävät tahot sekä varsinaisesta järjestelmäkehityksestä vastaavat henkilöt käytettävyyden huomioimisen etuihin. Lisäksi näissä pyritään kuvaamaan millä eri toimenpiteillä käytettävyyden huomiointi on mahdollista integroida organisaation nykyiseen kehitysprosessin eri vaiheisiin.

Ferre *et al* artikkeli ”*Usability Basics for Software Developers*” (Ferre et al. 2001) on hyvä lähtökohta kun halutaan tutustua käytettävyyden integroimiseen organisaation järjestelmäkehitykseen. Artikkelin on lyhyt mutta tiivis, ja se sisältää viittauksia moniin yleisesti hyvinä pidettyihin lähteisiin. Huomionarvoista on että artikkelissa todetaan eksplisiittisesti että ”*puhtaan vesiputouksmallin noudattaminen tekee käytettävyystekniikoiden sisällyttämisestä järjestelmäkehitykseen melko mahdotonta*” (Ferre et al. 2001).

Heiskari *et al* artikkeli ”*Bridging the Gap Between Usability and Requirements Engineering*” (Heiskari et al. 2009) esittelee kaksi organisaatiota joissa käytettävyys on otettu mukaan osaksi järjestelmäkehitysprosessia. Tutkijat havaitsivat että käytettävyys ymmärretään pelkästään käyttöliittymäsuunnitteluna, vaikka käytettävyys on huomattavasti laajempi käsite. Esimerkiksi ISO 9241-11 standardissa käytettävyys määritellään seuraavasti ”*Mitta, miten hyvin määrätyt käyttäjät voivat käyttää tuotetta määrätyssä käyttötilanteessa saavuttaakseen määrätetyt tavoitteet tuloksellisesti, tehokkaasti ja miellyttävästi.*” (ISO9241-11 2000). Huomionarvoista on että käyttöliittymää ei tässä määrittelyssä mainita ollenkaan. Heiskari *et al* artikkelissa (Heiskari et al. 2009) todetaan myös että käytettävyysongelmien havaitseminen on hankalaa pelkästään vaatimuksia katselmoimalla, koska silloin käytettävyysasiantuntijoilta puuttuu laajempi kokonaiskuva järjestelmän halutusta toiminnasta ja sen käyttäjien tehtävistä. Lopuksi kirjoittajat toteavat että käytettävyyden integroiminen osaksi vaatimusmäärittelyprosessia on haastavaa.

Bosch mainitsee artikkelissaan ”*Designing Software Architectures for Usability*” (Bosch 2003) että käytettävyyden lisääminen järjestelmään järjestelmäkehityksen loppuvaiheessa tai sen valmistumisen jälkeen on kallista, eivätkä arkkitehtuuriin vaikuttavat muutokset ole enää käytännössä mahdollisia. Tämän vuoksi käytettävyyttä tulisi käsitellä yhtenä järjestelmän laatukriteerinä, eikä pelkästään keskittyä järjestelmän kehittävän organisaation toivomiin ominaisuuksiin. Tämä tarkoittaa että käytettävyys tulisi huomioida jo järjestelmäkehitykseen käytetyssä ohjelmisto-arkkitehtuurissa. Käytettävyys laatukriteerinä on jo huomioitu tietyissä ohjelmisto-arkkitehtuureissa, ohjeissa ja standardeissa. Esimerkiksi IEEE:n standardi 1061 (IEEE standards 1998) huomioi käyttäjänäkökulman järjestelmän laatukriteereiden määrittelyssä. Bosch nostaa artikkelissaan (Bosch 2003) myös esiin tarpeen käsitellä käytettävyyttä laajemmin kuin pelkästään käyttöliittymän osalta. Hän muistuttaa että käytettävyys koskettaa myös kaikkea käyttäjän ja järjestelmän välillä tapahtuvaa tiedon vaihtoa.

Hanhisalonen tutki pro-gradu työssään (Hanhisalo 2008) kahta, samalla tehtävänannolla liikkeelle lähtenyttä, ohjelmistoprojektia ja tarkasteli miten määrittelyvaiheen vaatimukset vaikuttavat valmiiden järjestelmien käyttöliittymään. Hanhisalonen mukaan ”*lähes kaikki tutkittujen järjestelmien määrittelyn aikana syntyneet käyttötapaukset sitoivat toiminnan ja sen toteutuksen käyttöliittymässä. Suurin osa käyttäjä- ja järjestelmävaatimuksista sitoivat kuitenkin vain sen että jokin toiminnallisuus on oltava järjestelmässä*” (Hanhisalo 2008). Järjestelmiin syntyneiden käytettävyyso Ongelmien vakavuusasteilla oli Hanhisalonen tutkimissa järjestelmissä yhteys ongelmien syntymissyhyyn. Vakavammat käytettävyyso ngelmat syntyivät hänen tutkimissa järjestelmissä jo vaatimusmäärittelyssä. Pienemmät ongelmat, kuten esimerkiksi ongelmat käyttöliittymän komponenttien sijoittelussa ja muotoilussa, syntyivät itse käyttöliittymän toteutuksen aikana.

Hanhisalo toteaaakin yhteenvedossaan että: ”*Yhteenvetona keskeisin havainto on, että perinteisellä tavalla toteutettu vaatimusmäärittely ilman todellisen käyttäjän työnkulkujen selvittämistä näyttäisi johtavan käyttöliittymäratkaisuihin, joissa toiminnallisuus ei vastaa todellisia työnkuluja. Lisäksi vaikuttaa vahvasti siltä, että käyttötapaukset sitovat hyvin vahvasti käyttöliittymäratkaisuja ja aiheuttava käyttöliittymään vakavia tehokkuusongelmia*” (Hanhisalo 2008).

3.3. Käytettävyyden huomioiminen tietojärjestelmien määrittelyissä

Kuten yllä olevasta voidaan todeta, on käytettävyyden huomioiminen tärkeää jo tietojärjestelmän vaatimusmäärittelyitä tehtäessä. Valitettavan usein vaatimusmäärittelijät eivät kuitenkaan osaa, tai heillä ei ole mahdollisuutta, hankkia tarvittavaa käytettävyydestä vaatimuksia laadittaessa. Juristo ehdottaa artikkelissaan ”*Guidelines for Eliciting Usability Functionalities*” (Juristo 2007) erillisen tietovaraston perustamista järjestelmää kehittävälle organisaatiolla, josta vaatimusmäärittelijät voivat hakea ohjeita käytettävyyden huomioimisessa vaatimusmäärittelyitä tehtäessä. Ohjeet ovat Juriston ehdottamassa tietovarastossa malleja, joita käyttämällä voidaan kerätä tarpeellinen tieto käytettävyyden huomioimisesta vaatimusmäärittelyissä. Periaate näiden mallien takana on sama kuin joissakin järjestelmän toiminnallisten osien määrittelyissä käytettävissä malleissa tai käyttötapauksissa. Ne kuvaavat halutun toiminnallisuuden standardoidulla tavalla ja esittävät mahdollisia ratkaisumalleja tunnettuihin ongelmiin. Käsittelemällä käytettävyysoiminnallisuuksia samoin kuin järjestelmän toiminnallisia ominaisuuksia, voidaan käytettävyysoiminnallisuuksien vaatimukset selvittää ja kirjata tarkemmin, käyttäen samankaltaista prosessia kuin toiminnallisten ominaisuuksien kanssa. Juriston tekemien testien mukaan mallit auttoivat selkeästi käytettävyysoiminnallisuuksien huomioimisessa järjestelmäkehityksessä, erityisesti kun käytettävyyssiantuntijoita ei ollut riittävän hyvin käytettävissä (Juristo 2007).

Vesa-Matti Mäkisen pro-gradu tutkimuksessa (Mäkinen 2005) kuvataan miten ohjelmistoprojektissa käyttöliittymä suunnitellaan varsin pitkälle jo järjestelmän määrittelyvaiheessa. Suunnittelun yhteydessä muodostetut näyttökuvat, näyttökuvasarjat ja toimintalogiikan kuvaukset toimivat Mäkisen mukaan kyseisessä projektissa hyvin järjestelmän kuvaamiseen määrittelyvaiheessa. Mäkisen mukaan tällainen toimintatapa on toimiva, kun käyttöliittymäsuunnittelijat eivät ole ohjelmoijien käytettävissä koko projektin läpi. Mikäli käyttöliittymäsuunnittelijoita on käytettävissä koko projektin ajan, selvittää Mäkisen mukaan kevyemmällä käyttöliittymän dokumentoinnilla.

3.4. Kirjallisuuskatsauksen yhteenveto

Käytettävyyttä käsitellään kirjallisuudessa useasta eri näkökulmasta. Perusteokset auttavat eri koulutustaustan ja kokemuksen omaavia ihmisiä sisäistämään käytettävyyden peruskäsitteet ja jonkin yleisen toimintatavan käytettävyyden huomioivassa järjestelmäkehityksessä. Erilaiset toimintaohjeet syventävät osaamista yksittäisten tehtävien osalta. Täydelliset prosessikuvaukset toimivat vaihtoehtoina kun halutaan korvata nykyinen järjestelmäkehitysprosessi kokonaan uudella, tai on tarve kehittää järjestelmää kuvauksen erikoisalalla. Useammin organisaatiot eivät kuitenkin ole valmiita luopumaan käytössä olevista prosesseistaan, vaan lähinnä haluavat täydentää omaa järjestelmäkehitystään käytettävyyden huomioivilla toimenpiteillä. Tällöin ohjeet käytettävyyden sisällyttämisestä olemassa olevaan järjestelmäkehitysprosessiin voivat olla varsin hyödyllisiä.

Näkökulmasta riippumatta, monet julkaisut korostivat selkeästi että on tärkeää huomioida käytettävyys jo järjestelmäkehityksen alkumetreillä, mieluiten jo vaatimusmäärittelyjen aikana. Tutkimastani aineistosta käy selkeästi ilmi että järjestelmän vaatimusmäärittelyillä ja koko määrittelyvaiheella on erittäin suuri merkitys järjestelmän lopullisen käytettävyyden suhteen. Aineiston perusteella voi olettaa että ainakin osa järjestelmien käytettävyysongelmista on jäljitettävissä järjestelmän määrittelyvaiheen puutteisiin ja ongelmiin.

Kirjallisuuteen tutustumisen aikana huomasin että omakohtaisista kokemuksista kertovia kuvauksia toteutuneista projekteista niihin liittyvine ongelmineen, ei tunnu olevan julkaistu samassa määrin kuin muita, yllä käsiteltyjen näkökulmien mukaisia julkaisuja. Esimerkkeinä kokemusperäisistä havainnoista voidaan kuitenkin pitää Mäkisen (Mäkinen 2005) ja Hanhisalon (Hanhisalo 2008) tekemiä pro-gradu tutkimuksia. Näissä tekijät ovat kuitenkin keskittyneet tietyn ongelma-asettelun tutkimiseen tarkastelun kohteena olevissa järjestelmissä niiden kehityksen aikana. Tutkivampaa näkökulmaa, jossa pyritään kartoittamaan järjestelmän kehityksen aikana tehdyt virheet ja löytämään käytettävyysongelmien lähteet järjestelmäkehityksessä, en ole onnistunut löytämään.

4. Tutkimusmenetelmät ja tutkimuksen kulku

Tässä luvussa kuvaan miten tutkimus eteni ja miksi valitsin työssä käytetyt menetelmät. Työn aloituskokouksessa päätimme yhdessä Liikenteen turvallisuusviraston edustajien kanssa mitkä järjestelmät olisivat tämän tutkimuksen kohteena. Tarkasteltuani eri tutkimusmetodeja päätin käyttää kolmea pääasiallista tutkimusmetodia tässä tutkimuksessa. Nämä ovat artefakti-analyysi, teemahaastattelut sekä affiniteettidiagrammien käyttö.

4.1. Tutkimuksessa käytetyt menetelmät

Artefakti-analyysiin päädyin käytettävissä olevan dokumentaation määrän vuoksi, sekä koska se antaa kuvan dokumentit tuottaneesta organisaatiosta. Teemahaastattelut olivat looginen valinta hankkia tietoa järjestelmien kanssa tekemisissä olevilta henkilöiltä kun tiesin järjestelmistä jo jotain ja halusin laajentaa tätä tietoa. Affiniteettidiagrammi oli minulle tuttu metodi, jonka olin todennut itselleni sopivaksi tavaksi yhdistää eri lähteistä hankkimaani tietoa kokonaiskuvaksi. Tutkimuksessa käytetyt menetelmät on kuvattu tarkemmin alla.

4.1.1. Artefakti-analyysi

Artefakti-analyysillä tarkoitetaan tässä yhteydessä järjestelmiin ja organisaatioihin liittyvän dokumentaation läpikäyntiä sekä varsinaisiin järjestelmiin tutustumista. Käytettävissä oli Liikenteen turvallisuusviraston kuvaustenhallintajärjestelmään KUHAan tallennetut järjestelmien määrittely- ja suunnitteludokumentit. Lisäksi pystyin tutustumaan järjestelmiin haastattelujen aikana.

Artefakti-analyysillä on mahdollista selvittää järjestelmän *käyttöominaisuuksia*, sen *käyttöliittymää* ja *informaationsisältöä*. Lisäksi on mahdollista havaita miten järjestelmää *täydennetään* erilaisin muistilistojen ja ohjeiden avulla (Hyysalo 2006). Dokumentaatiota tarkastelemalla on mahdollista tutustua dokumentaation tuottaneen organisaation yrityskulttuuriin varsinaisen dokumentaation sisällön lisäksi. Tämä mahdollistaa haastateltavien kertoman peilaamisen organisaation yrityskulttuuria vasten,

siten että havainnoidaan helpommin mahdolliset ristiriidat ohjeistuksen ja varsinaisen toiminnan välillä.

4.1.2. Teemahaastattelu

Teemahaastattelulla tarkoitetaan haastattelua jossa haastattelijalla on kysymysrunko, mutta sitä käydään läpi haastateltavan vastauksiin mukautuen (Hyysalo 2006). Hyysalon mukaan teemahaastattelu on hyvä vaihtoehto tilanteeseen, jossa haastattelija tietää jo jotain, muttei ole varma tietääkö hän riittävästi. Tämä vastasi hyvin tilannetta tutkimuksen aikana. Artefakti-analyysin, ja myöhemmin muiden haastattelujen, pohjalta minulla oli kuva järjestelmän käytettävyysongelmista ja niiden synnystä, mutta varmuus oletuksiin tuli vasta kun uusien haastateltavien kommentit alkoivat toistaa aiempia haastatteluja.

Haastatteluiden kohteina oli ongelmien kartoitusvaiheessa järjestelmästä nykyään vastaavia henkilöitä ja järjestelmien nykyisiä käyttäjiä. Ongelmien syntyhistoriaa selvitetessä haastattelin järjestelmästä vastaavia henkilöitä ja kehitysprojektissa mukana olleita. Haastattelujen kysymysrunko räätälöitiin haastateltavien tehtävän mukaan, niin että niiden avulla saataisiin mahdollisimman laaja käsitys järjestelmään ja sen kehitysprojektiin liittyvistä ongelmista. Haastattelun loppupuolella kysyin haastateltavalta sellaisista asioista jotka muut olivat ottaneet esiin, mutta nykyinen haastateltava ei ollut. Näin sain uuden näkökulman myös haastatellulta ensin unohtuneisiin asioihin. Esimerkki haastattelurungosta löytyy liitteestä 1.

4.1.3. Affiniteettidiagrammi

Affiniteettidiagrammi on menetelmä, jonka avulla voidaan yhdistää monista eri lähteistä saatu tieto yhdeksi tietokokonaisuudeksi. Tiedon osat kirjataan post-it lapuille tai vastaaville ja ne ryhmitellään loogisiksi kokonaisuuksiksi. Tällä tavoin on mahdollista löytää tiedoille yhteiset ryhmät sekä hahmottaa näiden välisiä yhteyksiä. Benyon *et al* antavat hyvän esimerkin affiniteettidiagrammin käytöstä kirjassaan ”Designing interactive systems” (Benyon, Turner & Turner 2005.).

Tässä tutkimuksessa toteutin affiniteettidiagrammien muodostamisen ”*Institute for Human and Machine Cognition*” organisaation toteuttamalla CmapTools konseptikarttaohjelmistolla (<http://cmap.ihmc.us>). Piirsin erilliset affiniteettidiagrammit järjestelmissä havaituista käytettävyysongelmista ja niiden syistä. Tutkimalla syntyvää karttaa pystyin selvittämään vaikuttavatko jotkin syyt useamman käytettävyysongelman syntyyn.

4.2. Tutkimuksen kohderyhmät

Tutkimuksen aikana selvitin VERO-, REKI- ja PIIKO- järjestelmien käytettävyysongelmia ja niiden syitä. Tätä varten haastattelin ensin järjestelmien ylläpidosta vastaavia ihmisiä Liikenteen turvallisuusviraston tietohallinnosta sekä järjestelmät omistavilta toimialoilta. Käyttäjiä haastattelin sekä Liikenteen turvallisuusvirastosta, jolloin kohteena olivat järjestelmän omistavan toimialan asiantuntijat. Tämän lisäksi haastattelin asiakaspalvelijoita organisaatioista jotka käyttävät Liikenteen turvallisuusviraston kehittämiä REKI- ja PIIKO- järjestelmiä. VERO- järjestelmän ulkoistettua puhelinpalvelua en haastatellut koska tämä olisi vaatinut pitempää matkustusta. REKI- järjestelmän Trafín ulkopuolisia käyttäjiä haastattelin Vakuutusyhtiö If:n eri pääkaupunkiseudun konttoreissa. PIIKO- järjestelmän Trafín ulkopuolisia käyttäjiä haastattelin Ajovarma Oy:n eri toimipisteissä pääkaupunkiseudulla lähialueineen. Näiden lisäksi haastattelin ihmisiä järjestelmät toimittaneista organisaatioista, eli Tieto Oyj:stä (VERO) sekä Logica Suomesta (REKI ja PIIKO). Haastatellut jakaantuivat neljään kohderyhmään oheisen taulukon mukaisesti.

Taulukko 2 Kohderyhmien haastattelut

Tehtävä	Yhteensä	VERO	REKI	PIIKO
Ylläpito	9	3	2	4
Käyttö/Trafi	8	2	5	1
Käyttö/kumppani	8	0	3	5
Kehitys	9	3	2	4
Yhteensä	34	8	12	14

Kuten Taulukko 2 käy ilmi, oli haastateltuja usein varsin vähän segmenttiä kohden. Tämä on haastavaa haastattelujen luottamuksellisuuden kannalta, erityisesti koska

ylläpidosta ja kehityksestä vastaavat henkilöt ovat edelleen tiiviissä yhteistyössä. Tämän vuoksi päätin yhdistää haastattelujen tulokset, jottei yksittäisiä mielipiteitä olisi mahdollista yhdistää tiettyyn haastateltavaan. Haastateltavien yhdenvertaisuuden nimissä ulotin tämän periaatteen kaikkiin kohderyhmiin.

4.3. Tutkimuksen kulku

Varsinainen tutkimuksen kulku oli seuraava. Aloituskokouksen ja tutkimuksessa käytettävien metodien valitsemisen jälkeen aloitin kirjallisuustutkimuksen. Keskityin kirjallisuuteen joka käsittelee käytettävyyttä osana järjestelmäkehitystä. Kirjallisuustutkimuksen ohessa aloitin tutustumisen VERO ja REKI järjestelmien dokumentaatioon. Erityisesti minua kiinnosti järjestelmätoimituksen kilpailutukseen liittyvä määrittelydokumentaatio, jonka perusteella järjestelmien suunnittelu ja toteutus on aloitettu. Saatuaani yleiskuvan VERO ja REKI järjestelmien tilasta niiden määrittelyvaiheessa, aloitin haastattelut. Pyysin kaikkiin haastatteluihin kirjallisen äänitysluvan, jonka myös sain. Kaikki haastattelut olivat luottamuksellisia. VERO- ja REKI järjestelmien ylläpidosta vastaavien haastattelut ajoittuivat syys- lokakuuhun 2009, käyttäjien haastattelut loka- marraskuuhun 2009 ja kehityksessä mukana olleiden vuorostaan joulukuuhun 2009 sekä tammikuuhun 2010. PIIKO- järjestelmän haastattelut ajoittuivat ylläpidosta vastaavien osalta helmi- maaliskuuhun 2010, käyttäjien osalta maaliskuuhun 2010 sekä kehityksessä mukana olleiden osalta maaliskuu- huhtikuuhun 2010.

Kaikkien haastattelujen kulku oli pääpiirteissään pitkälti seuraavankaltainen. Ensin alustin tilanteen kertomalla mistä oikein on kyse ja mitä asioita pyrin selvittämään. Haastattelun aluksi, ja tilanteeseen totuttautumiseksi, pyysin yleensä haastateltavaa kertomaan itsestään ja työtehtävistään sekä siitä, miten nämä liittyvät tutkimuksen kohteena olevaan järjestelmään.

Käyttäjiltä kyselin seuraavaksi heidän yleistä mielipidettään järjestelmästä ja miten se soveltuu heidän työhönsä. Tässä vaiheessa alkoi yleensä esiintyä kuvauksia ongelmallisista ratkaisuista järjestelmässä. Purin näitä auki kyselemällä niistä lisää. Spontaanisti kerrottujen ongelmien jälkeen kävin haastateltavien kanssa läpi järjestelmään liittyvää työprosessia askel askeleelta, niiltä osin kun se liittyi

järjestelmään, jolloin yleensä nousi esiin lisää ongelmia tai haastateltavia ihmetyttäviä ratkaisuja.

Järjestelmän kehityksessä ja ylläpidossa mukana olleilta kyselin järjestelmän käytön sijaan järjestelmässä esiintyneistä ongelmista. Lisäksi kyselin kehitysprojektin aikaisista asioista kuten ilmapiiristä, esiin tulleista ongelmista sekä erityisen hyvin tai huonosti sujuneista asioista. Esiin nousseita ongelmia tai muuten mielenkiintoisia seikkoja pyrin purkamaan auki esittämällä niistä lisäkysymyksiä. Kaikkien haastattelujen lopuksi pyysin haastateltavia kertomaan kaksi asiaa, jotka he korjaisivat järjestelmässä välittömästi jos tämä olisi mahdollista.

Ensimmäisenä haastattelin VERO- ja REKI- järjestelmien tämän hetkisestä ylläpidon kanssa tekemisissä olevia henkilöitä. Näistä haastatteluista sain kuvan järjestelmien tämän hetken toiminnasta, käyttäjien raportoimista ongelmista sekä järjestelmien kehityslistalla olevista asioista. Näiden tietojen pohjalta tiesin suunnilleen minkä tyyppisiä ongelmia järjestelmissä todennäköisesti esiintyy. Näiden ongelmien vaikutusta käyttäjiin tarkistin seuraavassa vaiheessa.

Jatkoin järjestelmän nykytilanteen kartoitusta haastattelemalla järjestelmän käyttäjiä niin Liikenteen turvallisuusvirastosta kuin sidosryhmistäkin. Haastattelujen perusteella muodostin affiniteettidiagrammin järjestelmien käytettävyysongelmista ja niihin liittyvistä asioista. Käyttäjien haastatteluista sain hyvän kuvan järjestelmien ongelmista jotka vaikuttavat loppukäyttäjien työn suorittamiseen. Piirsin ongelmista affiniteettidiagrammit, joiden avulla ongelmia oli mahdollista koota laajemmiksi kokonaisuuksiksi. Saadun affiniteettidiagrammin pohjalta listasin VERO- ja REKI-järjestelmissä esiintyneet, käyttäjien kannalta keskeiset, käytettävyysongelmat. Löydetyt keskeiset käytettävyysongelmat on kuvattu luvussa ”5 Järjestelmien keskeiset käytettävyysongelmat”.

Kun keskeiset käytettävyysongelmat olivat tiedossa, siirryin VERO- ja REKI-järjestelmissä havaittujen käytettävyysongelmien synnyn kartoittamiseen. Tämä tapahtui haastattelemalla järjestelmän kehitysprojektissa mukana olleita henkilöitä. Henkilöt toimivat mm. järjestelmävastaavina, suunnittelijoina, projektipäällikköinä tai toteuttajina. Haastattelujen kulku oli samankaltainen aiempien haastattelujen kanssa. Ensin pyysin haastateltavia kertomaan omasta työstään projektissa ja miten projekti oli yleisesti

toiminut. Tämän jälkeen tartuin kiinni mahdollisesti esiin tulleisiin ongelmiin tai käytin affiniteettidiagrammien osoittamia ongelmakohtia lähtöpisteinä haastattelun jatkamisessa. Haastattelussa pyrin selvittämään missä vaiheessa ongelmat oli ensin havaittu, mikä oli ne aiheuttanut sekä miten ongelmiin oli reagoitu. Yritin myös selvittää reaktioiden taustalla olevat syyt. Haastatteluissa esiin tulleista syistä piirsin uudet affiniteettikaaviot, joiden perusteella pystyin kartoittamaan mitkä syyt olivat järjestelmien käytettävyysongelmien taustalla. Käytettävyysongelmien syyt ja syntyhistoriat on kuvattu luvussa ”6 Käytettävyysongelmien syyt”.

Kun VERO- ja REKI- järjestelmät oli tällä tavoin tutkittu, siirryin PIIKO- järjestelmän tutkimiseen. Suoritin tutkimuksen samalla tavalla kuin VERO- ja REKI- järjestelmien kohdalla. Myös PIIKO- järjestelmän keskeiset käytettävyysongelmat löytyvät luvusta ”5 Järjestelmien keskeiset käytettävyysongelmat”. PIIKO- järjestelmän käytettävyysongelmien syyt on samoin kuvattu luvussa ”6 Käytettävyysongelmien syyt”. Haastatteluita kertyi tutkimuksen aikana kaiken kaikkiaan 34 kappaletta, joista 16 oli käyttäjien haastatteluja ja 18 oli järjestelmän kehitykseen tai ylläpitoon liittyvän henkilön haastattelua. PIIKO- järjestelmän tiimoilta haastattelin kuusi käyttäjää ja kahdeksan kehitykseen tai ylläpitoon liittyvää henkilöä. REKI- järjestelmän kohdalla haastattelin neljäätoista käyttäjää ja neljä kehitykseen tai ylläpitoon liittyvää henkilöä. VERO- järjestelmän osalta haastattelin kahta käyttäjää sekä kuutta kehitykseen tai ylläpitoon osallistunutta henkilöä. REKI- järjestelmän käyttäjiä haastateltiin suhteessa eniten, koska myös REKI- järjestelmän käyttäjiä on maan laajuisesti eniten (tuhansia). PIIKO- järjestelmän käyttäjiä on myös ympäri maata mutta selkeästi vähemmän kuin REKI- järjestelmän käyttäjiä. VERO- käyttäjät ovat suureksi osaksi Liikenteen turvallisuusviraston virkamiehiä, ja heitä on määrällisesti varsin vähän (kymmeniä). Tämän vuoksi VERO- järjestelmän käyttäjiä haastateltiin vain sen verran että voitiin todeta ongelmien olevan järjestelmästä vastaavien henkilöiden kertoman mukaisia.

Haastattelujen tuloksien, ja niistä koottujen affiniteettidiagrammien, perusteella loin mallin käytettävyysongelmien synnystä ja siirtymisestä järjestelmän mukana tuotantoon. Malli on kuvattu luvussa ”7 Johtopäätökset”. Mallin, artefakti-analyysin sekä haastattelujen tulosten perusteella tein Liikenteen turvallisuusvirastolle joitakin suosituksia, joiden avulla tämän tyyppisiä käytettävyysongelmia voidaan vähentää tai välttää tulevissa järjestelmäkehitysprojekteissa. Suositukset löytyvät luvusta ”8

Suosituksel”. Lopuksi tein yhteenvedon tutkimuksen kulusta ja sen aikana esiin tulleista asioista. Yhteenveto löytyy luvusta ”9 Yhteenveto”.

5. Järjestelmien keskeiset käytettävyysoongelmat

Tässä luvussa kuvaan järjestelmien nykytilaa, siten kun olen kartoittanut sen järjestelmien dokumentaation sekä käyttäjien ja järjestelmästä vastaavien henkilöiden haastattelujen perusteella. Luvussa kuvaan järjestelmien keskeisiä käytettävyysoongelmia sanallisesti, sekä pyrin tiivistämään tärkeimmät ongelmat affiniteettidiagrammien perusteella piirrettyjen kuvien avulla.

5.1. VERO:n keskeiset käytettävyysoongelmat

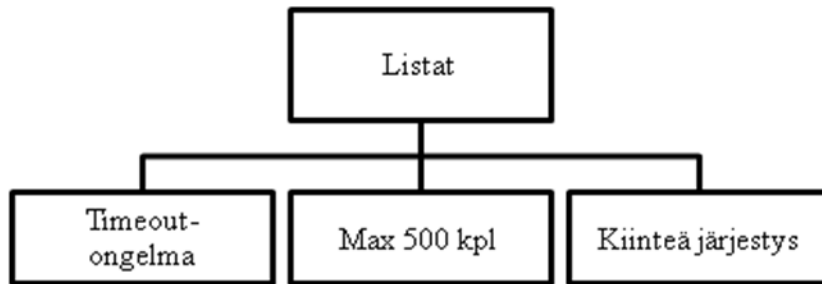
Tässä osassa esittelen haastatteluissa esiin nousseet VERO- järjestelmän käytettävyysoongelmat. VERO- järjestelmän keskeisimmiksi käytettävyysongelmiksi nousi haastatteluissa

- Pitkien listojen toiminta (ks. kohta ”5.1.1 Listat”)
- Tietojen synkronointi järjestelmien välillä (ks. kohta ”5.1.2 Tietojen synkronointi”)
- Eräajot (ks. kohta ”5.1.3 Eräajot”)
- Käyttöliittymän yleiset käytettävyysoongelmat (ks. kohta ”5.1.4 Käyttöliittymän yleiset ongelmat”)

Näiden lisäksi haastateltujen käyttäjien työhön vaikutti maksujärjestely-ominaisuuden erittäin hankala toiminta, viitenumeroiden saannin rajoittuneisuus sekä prosessiin liittyvät asiat (ks. kohta ”5.1.5 Muita VERO käyttäjien kokemia ongelmia”).

5.1.1. Listat

Listalla tarkoitan tässä yhteydessä käyttöliittymällä listamuodossa esitettävää informaatiota. VERO järjestelmässä esiintyvien erinäisten listojen toiminta nousi haastatteluissa erääksi käyttäjien pääongelmista.



Kuva 1 VERO:n ongelmat - listat

VERO järjestelmässä käsitellään todella suuria materiaaleja, jolloin tulee usein vastaan tilanteita (esim yritysten yhteislasku, alla), joissa tietojen kerääminen tietokannasta käyttöliittymän esittämälle listalle kestää pitempään kuin Ajoneuvoliikenteen tietojärjestelmän (ATJ) yleinen kahden minuutin ”*timeout*” raja. Tämä johtaa tilanteeseen jossa käyttöliittymältä ei ole mitenkään mahdollista tarkastella esimerkiksi tiettyjen suurten kumppaneiden yhteislaskuja, koska järjestelmä ei koskaan saa haluttua tietomäärää koostettua alle kahden minuutin rajan. Toisissa listoissa haku on puolestaan rajattu 500 riviin, jolloin lista saadaan tämän kokoisena näkyviin. Järjestelmään ei kuitenkaan ole toteutettu mahdollisuutta hakea mahdollisesti tätä seuraavia rivejä, jolloin osa riveistä on myös tässä tapauksessa kokonaan käyttäjien ulottumattomissa.

Tilanteet joissa tietoja ei saa näkyviin järjestelmästä käyttöliittymän avulla ovat ongelmallisia eritoten suorissa asiakaskontakteissa. Mikäli tällainen tilanne syntyy, esimerkiksi asiakkaan kanssa käytävän puhelinkeskustelun aikana, ei käyttäjälle jää mitään muuta vaihtoehtoa kuin ilmoittaa että hän ottaa yhteyttä asiakkaaseen kun tiedot on poimittu esiin muilla keinoin. Usein tämä tarkoittaa erillisellä tietokannan raportointityökalulla tehtävää, manuaalista, poimintaa.

Listat aiheuttavat myös muuta, pienempää harmia käyttäjille. Järjestelmä pakottaa tiettyjen hakujen yhteydessä käyttäjän kulkemaan hakutulostilan kautta, vaikka hakutuloksia olisi vain ja ainoastaan yksi. Tällaisissa tapauksissa olisi luonnollista olettaa että järjestelmä ei turhaan näytä hakutuloksia, vaan tuo ainoan hakutuloksen suoraan käyttäjän nähtäväksi. Toisissa tapauksissa ei muodostetulle listalle ole mahdollista palata sen jälkeen kun käyttäjä on aukaissut yhden listan olion, vaan lista on muodostettava uudelleen. Tämä hidastaa käyttäjän työtä ja hankaloittaa tietojen löytämistä, mikä vaikuttaa mahdolliseen asiakaskontaktiin.

Useassa haastattelussa nousi esiin eräs esimerkki listoista, ja niihin liittyvistä ongelmista, nimittäin *yhteislasku*. Yhteislaskulla tarkoitetaan järjestelmän toiminnallisuutta jonka avulla yrityksen omistamien autojen ajoneuvoverot yhdistetään yhdelle, kuukausittaiselle, laskulle. Suuria määriä autoja omistavien yritysten kohdalla tämä voi tuoda merkittäviä säästöjä hallinnollisissa kuluissa. Yhteislaskulle muodostuu siis lista kaikista yrityksen laskutus sopimuksen piiriin ilmoitetuista autoista. Yllä mainittujen listojen yhteisten ongelmien lisäksi, yhteislasku on ongelmallinen muodostaa. Eräajo joka muodostaa pohjan seuraavan kuukauden yhteislaskua varten, vaatii että järjestelmässä ei ole ajon aikana mitään muuta toimintaa. Lisäksi yhteislaskulla kuukausittain saadut rahat jyvitetään yksittäisille ajoneuvoille, jolloin yksittäisen ajoneuvon kohdistuva laskun osuus on usein alle laskutusrajan (10€). Mikäli ongelmia esiintyy, on erittäin hankalaa seurata maksuliikennettä yksittäisen auton osalta. Tämä tarkoittaa että ongelmatilanteissa ei yksittäisen ajoneuvon puuttuvia maksuja välttämättä huomata tai laskuteta. Koska valvonta on manuaalista, syntyy tässä mahdollisuus jopa veronkiertoon.



Kuva 2 VERO:n ongelmat - yhteislasku

5.1.2. Tietojen synkronointi

VERO- järjestelmä on suurilta osin riippuvainen muiden järjestelmien tuottamasta tiedosta ajoneuvoista ja niiden omistajista. Ajoittain tämä tieto ei kuitenkaan kulje VERO- järjestelmään ajanmukaisesti tai oikein, jolloin seurauksena on tiedon synkronointiin liittyviä ongelmia. Erityisen vakavana haastatellut kokivat tämän ongelman tilanteissa, joissa pitäisi tehdä takautuvia muutoksia. Tällaisia ovat esimerkiksi VERO- järjestelmän historiatietojen muuttaminen jälkikäteen saatujen, verovelvollisuutta tai veroluokkaa koskevien tietojen kirjaaminen järjestelmään. Mikäli tällaisissa tilanteissa havaitaan että VERO- järjestelmässä ja ajoneuvorekisteröinnin

REKI- järjestelmässä olevat tiedot eivät ole samalla tasolla, ei VERO- järjestelmä välttämättä kykene mitenkään käsittelemään tapausta.

Tällaiset tilanteet pakottavat usein tietokannan päivittämiseen ohi järjestelmän, suorilla tietokantakomennoilla. Tällainen nk. ”kantapuukutus” on aina riskialtis ja sellaisen tuloksena saattaa syntyä uusia virheitä ja siten käyttäjille ja järjestelmän ylläpitäjille enemmän töitä.

5.1.3. Eräajot

Vahvasti eräajopohjaisessa järjestelmässä on odotettavissa että osa ongelmista koskettaa nimenomaan eräajoja. Haastatteluissa ongelmaksi koettiin usein aineiston massiivinen koko ja sen jatkuva kasvaminen. Kävi ilmi että järjestelmän toiminta on hidastunut aineiston kasvun myötä. Kriittiseksi ongelman tekee eräajojen kesto ja niille varattujen aikaikkunoiden täytyminen. Esimerkiksi ”ajoneuvoverolippuajo” kestää nykyään noin 6 tuntia ja se on ajettava 6 kertaa viikossa, jolloin jo muutamankin ajon viivästyminen tai epäonnistuminen aiheuttaa pitkän ja hankalan korjaus ja ”kiinniajo” rupeaman.



Kuva 3 VERO:n ongelmat - eräajot

VERO- järjestelmällä ei siis nykyisellään ole juurikaan joustovaraa ongelmatilanteiden tai yllättävän huollon hallittuun järjestämiseen.

5.1.4. Käyttöliittymän yleiset ongelmat

Vero- järjestelmän käyttöliittymä oli haastateltujen mukaan suhteellisen toimiva. Jonkin verran käyttäjät kaipaivat vanhan, tekstipohjaisen sovelluksen toiminnallisuuksia kuten pikanäppäimiä, ”enter”- napilla tapahtuvaa tallennusta ja tiivistä tiedon esittämistapaa.

Myös järjestelmän hitautta aikaisempaan verrattuna kritisoitiin. Hiirellä toimiminen vaatii käden suurempaa liikuttelua kun aiemmin ja on siten hitaampaa kuin pikanäppäinten käyttö. Navigaatiopolku ja siirtymät koettiin myös ajoittain hankaliksi, samoin kuin huomautustietojen käyttö.

Järjestelmän yleistä hitautta siirtyessä näytöltä toiselle kritisoitiin myös. Tilanne kärjistyi sellaisissa tilanteissa joissa VERO- järjestelmä hakee tietoja näytölle useasta muusta järjestelmästä. Lisäksi järjestelmän alkuaikoina esiintyi järjestelmän epävakautta, mikä aiheutti paljon lisätyötä niin käyttäjille kuin järjestelmän ylläpitäjille.



Kuva 4 VERO:n ongelmat - käyttöliittymä

5.1.5. Muita VERO käyttäjien kokemia ongelmia

Maksujärjestely- toiminnallisuuden oli tarkoitus mahdollistaa sen että auton verot maksaa joku muu kuin ajoneuvon omistaja. Maksujärjestely- toiminnallisuus koettiin kuitenkin niin hankalaksi käyttää että sitä ei enää käytetä!

Viitenumeroita on mahdollista osoittaa ajoneuville enintään 9 kpl per päivä. Tämä muodostuu ongelmaksi takautuvien muutosten osalta, koska järjestelmä muodostaa ajoneuvon verohistorian laskuineen uudelleen muutoshetkestä eteenpäin. Mikäli muutoshetken jälkeen on paljon verotapahtumia, pakottaa tämä takautuvien tapahtumien käsittelyn jättämisen puolitiehen odottamaan seuraavaa päivää ja seuraavaa 9 viitenumeroa.

Käyttäjät kokivat prosessin mukaisen työn muuttuneen monimutkaisemmaksi samalla kuin henkilöstön määrä oli vähentynyt ja tehtäviä pyritty ulkoistamaan. Alun perin määritellyt työtehtävät eivät enää vastanneet todellisuutta, eikä niitä enää nykyisin

suunniteltaisi samankaltaisiksi. Lisäksi Liikenteen turvallisuusviraston ajoneuvoverotus yksikkö on päätetty alueellistaa Rovaniemelle. Kuulemani mukaan yksikön mukana Rovaniemelle ei lähde juuri kukaan. Nämä asiat tekevät ajoneuvovero-yksikön toiminnan varsin haasteelliseksi lähitulevaisuudessa.

VERO- järjestelmän keskeisiksi käytettävyysoongelmiksi nousi siis haastattelujen pohjalta pitkien listojen toiminta, tietojen synkronointi-ongelmat, eräajojen hitaus ja kaatuilu sekä käyttöliittymän yleiset käytettävyysongelmat. Näiden syntyhistoria selvitetään luvussa ”6.1 VEROn käytettävyysongelmien synty”.

5.2. REKIn keskeiset käytettävyysongelmat

Tässä osassa esittelen haastatteluissa esiin nousseet REKI- järjestelmän käytettävyysongelmat. REKI- järjestelmän keskeisiksi käytettävyysoongelmiksi nousi haastatteluissa

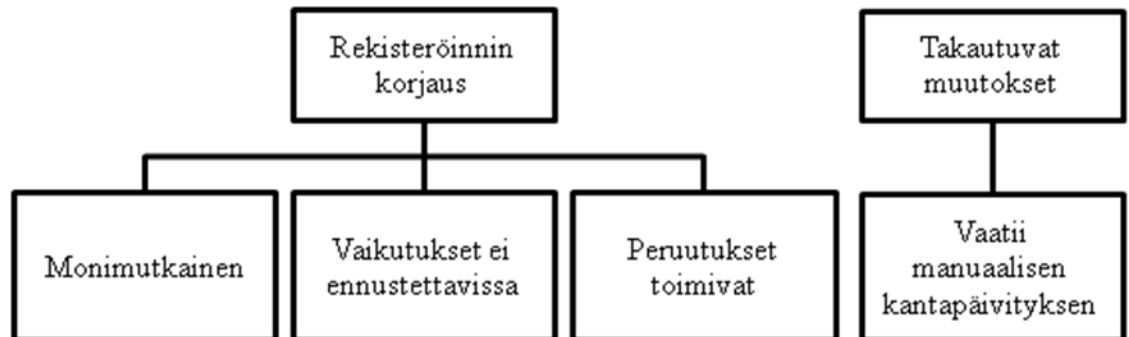
- Rekisteröintien korjaukset ja takautuvat muutokset (ks. kohta ”5.2.1 Rekisteröintien korjaukset ja takautuvat muutokset”)
- Käyttöliittymän yleiset käytettävyysongelmat (ks. kohta ”5.2.3 Käyttöliittymän yleiset käytettävyysongelmat”)
- Tulostus (ks. kohta ”5.2.3 Käyttöliittymän yleiset käytettävyysongelmat”)

Näiden lisäksi käyttäjien työhön vaikutti ATJn yleiset ongelmat ja järjestelmän tuottamiin raportteihin liittyvät ongelmat (ks. kohta ”5.2.4 Muut REKI käyttäjien kokemat ongelmat”).

5.2.1. Rekisteröintien korjaukset ja takautuvat muutokset

Haastatteluissa tuli esiin että käyttäjien mielestä vakavimmat ongelmat liittyivät rekisteröintien korjauksiin ja niistä aiheutuviin takautuviin muutoksiin. Rekisteröinnin korjaustoiminnot koettiin epävakaaiksi ja arvaamattomiksi. Korjaukset ovat erityisen hankalia ja monimutkaisia toteuttaa koska järjestelmässä olevat ajoneuvojen tiedot ovat toisistaan riippuvaisia, usein odottamattomalla ja reaaliaikaisesti kytkeytymättömällä tavalla. Pahimmassa tapauksessa yhden ajoneuvon tietoihin tehty korjaus aiheuttaa

järjestelmän toimesta hallitsemattoman muutoksen toisen ajoneuvon tietoihin. Korjaukset koettiin myös monimutkaisiksi suorittaa. Yksikertaisemmat tapahtumien poistot kuitenkin koettiin hyvin toimiviksi.

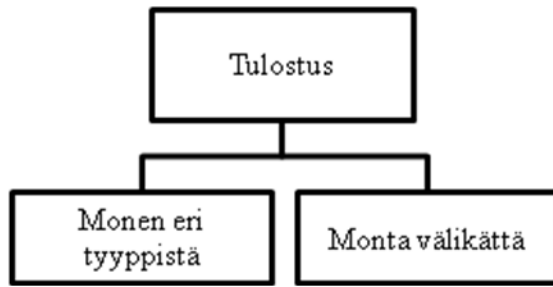


Kuva 5 REKIn ongelmat - korjaukset ja takautuvat muutokset

Korjaustoimintojen kautta tehtävät takautuvat muutokset todettiin käytännössä mahdottomiksi toteuttaa järjestelmän kautta. Useimmiten takautuvat muutokset joudutaan tekemään manuaalisesti suorina tietokantakomennoilla tehtävinä kantapäivityksinä, eli niin kutsuttuina ”kantapuukotuksina”. Tällaiset manuaaliset toimenpiteet suoraan tietokantaan ovat riskialttiita ja voivat aiheuttaa suurta sekaannusta ja lisätyötä jos niitä ei tehdä harkiten.

5.2.2. Tulostus

Tulostuksessa on REKI- järjestelmässä useita eri vaihtoehtoja. Käyttäjän käytettävissä olevat vaihtoehdot riippuvat tämän työtehtävästä sekä kumppanin sopimuksellisesta tilanteesta. Vaihtoehtojen moninaisuuden vuoksi on usein vaikea päätellä mitä tulosteelle tapahtui, mikäli se ei ilmesty halutulle tulostimelle. Pahimmassa tapauksessa mukana on monta toimijaa (REKI- järjestelmä, sanomanvälityksen tukijärjestelmä SAIMI, käyttöpalvelutoimittajan välityspalvelu sekä kumppanin oma tulostusjärjestelmä), jolloin ongelman paikallistaminen on erittäin haastavaa.



Kuva 6 REKIn ongelmat - tulostus

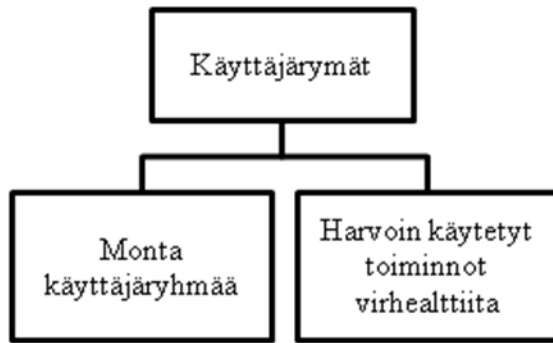
5.2.3. Käyttöliittymän yleiset käytettävyysoongelmat

Haastattelujen aikana tuli esiin että REKI- järjestelmässä on jonkin verran perinteisiä käytettävyysongelmia, jotka hidastavat ja hankaloittavat järjestelmän käyttöä. Suuri osa näistä liittyy näyttöjen määrään ja niiden käyttöön eri tilanteissa. Ajoneuvoihin liittyvän tietomäärän vuoksi, ja voimassa olleen käyttöliittymästandardin mukaisesti, on tietoja jaettu usealle eri näytölle.



Kuva 7 REKIn ongelmat – käyttöliittymän yleiset ongelmat

Tietojen jakaminen usealle näytölle aiheuttaa tarpeen siirtyä näiden välillä. Käyttäjät joutuvatkin usein työtä tehdessään navigoimaan usean näytön kautta voidakseen suorittaa haluamansa toimenpiteen. Näyttöjen välinen navigaatio ei kuitenkaan aina toimi oikein, vaan siirtymän aikana aiemmin syötettyjä tietoja saattaa kadota. Navigaatio ei myöskään ole aina täysin looginen, vaan haastateltavat totesivat että se joudutaan usein opettelemaan eri lailla eri työtehtävissä. Lisäksi mainittiin erityisen ongelmallisena järjestelmässä navigointiin tarkoitetut linkit, joita tottumaton käyttäjä joutuu etsimään sekä järjestelmän antamat ilmoitukset, jotka koettiin olevan huonosti näkyvillä.



Kuva 8 REKIn ongelmat - käyttäjäryhmiin liittyvät ongelmat

Tässä tutkimuksessa lasken käytettävyysoongelmiin myös eri käyttäjäryhmien erikoispiirteistä johtuvat ongelmat. Haastatteluissa tuli esiin koulutuksen ja jokapäiväisen käytön tärkeys REKI- järjestelmän käytön sujuvuuden parantamisessa. Sellaiset käyttäjät ja käyttäjäryhmät jotka eivät päätökseen käytä järjestelmää, tai ovat vasta aloittelemassa järjestelmän käyttöä, tekevät useammin virheitä kuin järjestelmää päätyökseen käyttävät. Lisäksi käyttäjien virhealttiutta lisäsi sellaiset työtehtävät joissa käytetään vain osaa järjestelmästä. Tottumattomat käyttäjät voivat helposti unohtaa jonkin askeleen työjärjestyksessä koska järjestelmä ei aktiivisesti ohjaa käyttäjää toimimaan oikein. Tällöin esimerkiksi ajoneuvon tietoihin tehdyn muutoksen hyväksyminen päänäytöllä tietojen tallennuksen jälkeen voi helposti unohtua, jolloin muutokset eivät astu voimaan vaan tehty työ menetetään.

5.2.4. Muut REKI käyttäjien kokemat ongelmat

Muista ongelmista mainittiin päällimmäisenä viime aikojen ATJn yleiset ongelmat, jotka peittävät alleen muut mahdolliset järjestelmäkohtaiset ongelmat ja viat. Näistä ATJ-tasoisista ongelmista tärkeimpänä mainittiin järjestelmien jatkuva hidastelu ja kaatuminen, joka on häirinnyt käyttäjien työtä erityisesti asiakaspalvelutilanteissa. Lisäksi haastatteluissa mainittiin tiettyjen raporttien selkeyden puute mahdollisena ongelmana.



Kuva 9 REKIn ongelmat - muut

ATJn yleinen hidastelu ja kaatuilu olivat ongelmallisia myös tämän tutkimuksen kannalta, koska se peitti alleen muut ongelmat useiden haastattelujen alkupuolella. Haastatteluissa kului jonkin verran ylimääräistä aikaa siihen että haastateltavat saivat purkaa turhautumistaan tähän ongelmaan. Vasta tämän jälkeen pystyin ohjaamaan haastattelua itse REKI- järjestelmän ongelmiin.

5.3. PIIKOn keskeiset käytettävyysoingelmat

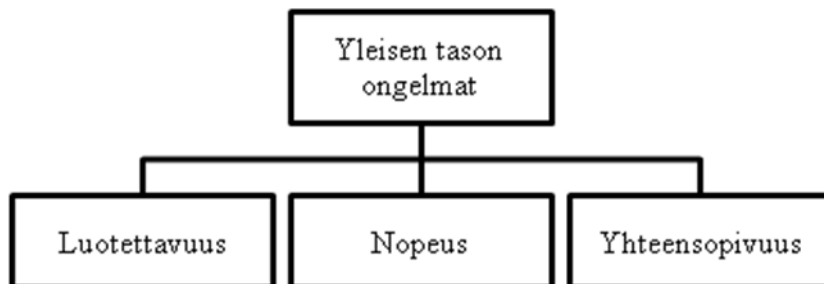
Tässä osassa esitellään haastatteluissa esiin nousseet PIIKO järjestelmän käytettävyysoingelmat. PIIKO järjestelmän keskeisimmiksi käytettävyysoingelmiksi nousi haastatteluissa

- Yleisen tason ongelmat (ks. kohta ”5.3.1 Yleisen tason ongelmat”)
- Listoilte jäävät hakemukset (ks. kohta ”5.3.2 Listoilte jäävät hakemukset”)
- Korvaavat kortit (ks. kohta ”5.3.3 Korvaavat kortit”)
- Yrityskortit (ks. kohta ”5.3.4 Yrityskortit”)
- Laskut (ks. kohta ” 5.3.5 Laskut”)
- Käyttöliittymän muut käytettävyysoingelmat (ks. kohta ”5.3.6 Muut PIIKO-järjestelmän käytettävyysoingelmat”)

Näiden lisäksi käyttäjien työhön vaikutti varsinaisen järjestelmän ulkopuoliset ongelmat kuten ohjeistus ja järjestelmätuki sekä prosessiin ja lomakkeisiin liittyvät asiat (ks. kohta ”5.3.7 Järjestelmän ulkopuoliset ongelmat”).

5.3.1. Yleisen tason ongelmat

PIIKO järjestelmä kärsii yleisellä tasolla käyttäjien kokemasta hitaudesta, järjestelmän satunnaisesta kaatuilusta sekä rajoitetun selaintuen aiheuttamista ongelmista. Luotettavuuden puute on eräs yleisimmistä käyttäjiä vaivaavista ongelmista PIIKO-järjestelmän käytössä. Ongelma on tuttu myös muissa ATJ- järjestelmissä ja PIIKO-järjestelmässä se vaikeuttaa huomattavasti käyttäjien toimintaa asiakaspalvelutilanteissa.



Kuva 10 PIIKOn yleisen tason ongelmat

Järjestelmän kaatuessa kesken asiakaspalvelutilannetta joutuu käyttäjä selittämään tilanteen asiakkaalle, vastaanottamaan asiakkaan hakemuksen ja syöttämään sen järjestelmään myöhempänä ajankohtana jolloin järjestelmä on jälleen toiminnassa. Mahdolliset käteismaksut on mahdollista suorittaa kumppanin oman kassajärjestelmän avulla ilman PIIKO- järjestelmää, mutta laskua ei ole mahdollista tulostaa ilman järjestelmän päällä oloa. Tämä johtaa siihen että käyttäjän on muistettava pyytää asiakkaalta mahdolliset toimitus- ja laskutusosoitteet sekä kirjata ne ylös niin että ne ovat käytettävissä kun hakemusta päästään syöttämään järjestelmään.

Luotettavuuden puute on myös erityisen ongelmallista syötettäessä PIIKO- järjestelmään asiakkaiden ammattipätevyyteen liittyviä koulutustietoja järjestelmän ”massasyöttö” toiminnallisuuden avulla. Tämä toiminnallisuus mahdollistaa usean asiakkaan, kuten esimerkiksi kaikki tietyn kurssin läpäisseet, tietojen syöttämisen samanaikaisesti yhdeltä näytöltä. Mikäli järjestelmä kaatuu kesken tietojen syötön, katoavat jo syötetyt tiedot käyttäjän näkyvistä, ilman että tämä pystyy helposti päättelemään tallentuiko edes osa tiedoista. Tämä on johtanut siihen että käyttäjät ovat usein syöttäneet kaikki tiedot uudelleen järjestelmään, jolloin osalle henkilöistä on kirjattu useampi koulutusmerkintä. Tämä voi pahimmillaan aiheuttaa ansaitsematonta hyötyä näille henkilöille koska heille

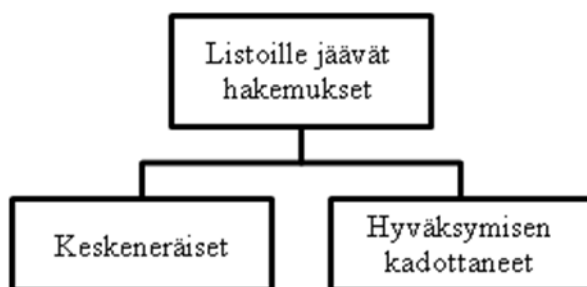
kertyy ammattipätevyuden ylläpitämiseen tarvittavia koulutuspäiviä ilman todellista koulutustapahtumaa.

Nopeus, tai lähinnä sen puute, on tutkittujen järjestelmien yleinen ongelma. Käyttäjän valitseman toimenpiteen suorittamiseen voi kulua pitkään. Pahimmillaan toimenpide kestää niin pitkään että ATJ- arkkitehtuurin määrittelemät ”*timeout*” arvot ylittyvät ja järjestelmä antaa vain virheilmoituksen. Ajoittain järjestelmä toimii niin hitaasti että käyttäjä joutuu pyytämään asiakkaalta lupaa palata asiaan myöhemmin esimerkiksi puhelimitse. Tällaiset tilanteet rasittavat niin käyttäjiä kuin asiakkaitakin. Lisäksi ne antavat huonon kuvan järjestelmästä ja viraston palvelusta vaikka itse asia tuleekin hoidettua.

ATJ- järjestelmien, ja siten myös PIIKO- järjestelmän, selaintuki on varsin rajattua (ks. kohta ”2.2 Ajoneuvoliikenteen tietojärjestelmä”). Tämä on aiheuttanut ongelmia PIIKO- järjestelmää käyttäville organisaatioille vanhojen selainten tuen loppumisen ja näiden selainten tunnetun turvattomuuden vuoksi, jolloin koneille on jouduttu asentamaan erilliset selaimet erikseen PIIKO- järjestelmän käyttöä varten.

5.3.2. Listoille jäävät hakemukset

Kirjaututtuaan PIIKO- järjestelmään käyttäjälle näytetään aloitussivu, jolla näkyy luovuttamista odottavat kortit sekä keskeneräiset korttihakemukset. Valitettavasti tällä listalla näkyy myös hakemuksia joita käyttäjä ei ole tarkoittanut jäävän sille.



Kuva 11 PIIKOn listoille jäävät hakemukset - ongelma

Näitä listoille jääviä hakemuksia syntyy kahdella tapaa. Ensinnäkin sinne ilmestyy keskeneräisiä hakemuksia, joita syntyy joka kerta kun hakemusta aletaan syöttää järjestelmään. Hakemus muodostuu ja tallentuu aina automaattisesti kun käyttäjä aloittaa

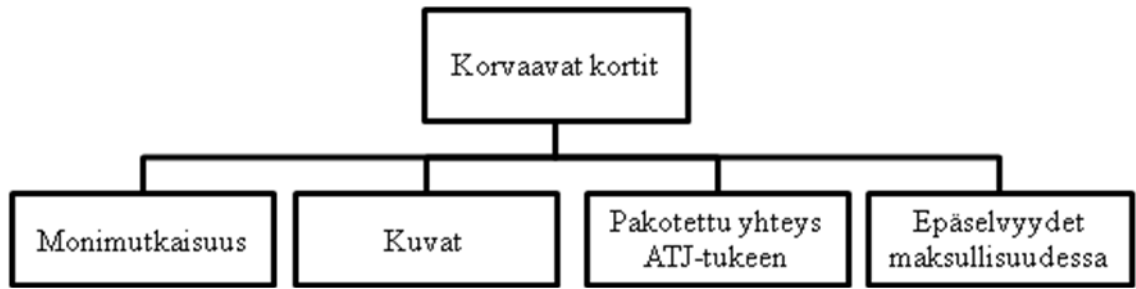
korttihakemuksen käsittelyyn. Käyttäjän ei ole mahdollista peruuttaa hakemuksen muodostumista mikäli huomataan että hakemus on turha, virheellinen, järjestelmä kaatuu, tai hakemusta ei jostain muusta syystä saateta loppuun. Tämä johtaa siihen että listoille ilmestyy turhia keskeneräisiä hakemuksia joita ei voida saattaa loppuun eikä poistaa käyttäjän toimesta. Järjestelmän pääkäyttäjä saa nämä poistettua, mikäli asiasta tehdään ilmoitus ATJ-tukeen. Hakemukset poistuvat myös itsestään ajan myötä mutta tämä vie pitkän aikaa, jona aikana hakemukset täyttävät turhaa listaa.

Toinen tapa jolla hakemukset jäävät käsittelylistalle, vaikka niiden pitäisi siirtyä prosessissa eteenpäin, on kun hakemus kadottaa hyväksymismerkintänsä. Tämä tapahtuu silloin kun käyttäjä on ensin merkinnyt hakemuksen hyväksytyksi, mutta sitten muistaakin että esimerkiksi toimitusosoite on väärin ja käy korjaamassa sen ilman että tallentaa hakemuksen ensin. Osoitteen korjaamisen yhteydessä käyttäjä siirtyy toiselle näytölle, jolloin aiemmalla näytöllä tehty hyväksyminen katoaa. Mikäli käyttäjä ei ole tästä ominaisuudesta tietoinen voi hän hyvinkin olettaa että hyväksyminen on voimassa, jolloin hän vain tallentaa hakemuksen olettaen että se siirtyy prosessissa eteenpäin. Tosiasiassa hakemus jääkin ”hyväksymätön” tilaan siihen asti kunnes käyttäjä huomaa ottaa sen käsittelyyn tai, pahimmassa tapauksessa, asiakas ottaa yhteyttä ja kyselee hakemuksensa perään.

Listoille jäävät hakemukset häiritsevät luovutettavien korttien ja muiden listoille kuuluvien hakemusten löytämistä virheellisten joukosta. Pahimmassa tapauksessa tämä voi johtaa asiakkaalle viikkojen viivästyksen esimerkiksi piirturikortin saamisessa.

5.3.3. Korvaavat kortit

Mikäli asiakas kadottaa piirturikorttinsa, tai se lakkaa toimimasta teknisen vian vuoksi, tulee hänen ilmoittaa asiasta Liikenteen turvallisuusvirastoon Ajovarma Oyn asiointiverkoston kautta mahdollisimman pian, jotta kortti voidaan kuolettaa ja uusi kortti anoa. Valitettavasti tämä prosessi on käyttäjien kannalta hankala ja suhteellisen altis virheille. Tätä samaa prosessia käytetään myös, mikäli asiakas haluaa muusta syystä vaihtaa nykyisen korttinsa, esimerkiksi nimen muuttumisen vuoksi.



Kuva 12 PIHKOn korvaavien korttien ongelmat

Korvaavan kortin käsittelyprosessi on suhteellisen monimutkainen, erityisesti verrattuna tavallisen piirturikortin hakemiseen. Tämä korostuu tilanteissa joissa käyttäjä törmää kadonneisiin kortteihin vain harvoin, mikä on tilanne suurimmalla osalla käyttäjiä. Käyttäjälle epätavallisessa tilanteessa tämä joutuu turvautumaan ohjeistukseen ja niihin muistikuviin mitä koulutuksesta sekä aiemmista tapauksista on jäänyt mieleen. Tämä tekee korvaavien korttien hakemisesta tavallista virhealttiimman toimenpiteen.

Tilannetta hankaloittaa piirturikorttiin liittyvien kuvien epäselvä käsittelyprosessi. Kuvat on periaatteessa tallennettu tietokantaan, jolloin on täysin mahdollista käyttää asiakkaan vanhoja kuvia. Järjestelmän hakemusnäytöllä kuvien liittäminen hakemukseen on kuitenkin toteutettu siten että käyttäjä helposti sekoittaa uusien ja vanhojen kuvien liittämiseen käytetyt valinnat. Mikäli hän vahingossa tekee järjestelmässä valinnan jonka mukaan asiakas on toimittanut uudet kuvat, vaikei näin olekaan, jää järjestelmä odottamaan kuvien digitoimista kyseiselle hakemukselle. Nyt kun kuvia ei ole tulossa jää koko hakemus tällöin jumiin.

Koska tilanne ei tule keskivertokäyttäjällä vastaan kovin usein, on Liikenteenturvallisuusvirasto ohjeistanut että käyttäjien on otettava yhteyttä ATJ-tukeen joka kerta kun he joutuvat kosketuksiin korttien korvaamisen kanssa. Tämä on aiheuttanut turhautumista, koska usein tukipuhelimeen vastaa henkilö jolla ei myöskään ole kokemusta tällaisesta tilanteesta. Tällöin käyttäjä ja tukihenkilö ”*tavaavat yhdessä ohjeita*”, kuten yksi haastateltu asian ilmaisi. Tilanne koetaan turhaksi ja nykyisen käytännön sijaan toivotaankin parempia ohjeita.

Joillekin loppukäyttäjille oli myös epäselvää missä tilanteissa korvaavat kortit ovat maksullisia. Myös tähän toivottiin parempaa ohjeistusta joko suoraan järjestelmään tai käyttäjien ohjeisiin.

5.3.4. Yrityskortit

PIIKO järjestelmän kautta myönnettävistä eri piirturikorttityypeistä nk. ”yrityskortit” ovat aiheuttaneet hämmennystä käyttäjien keskuudessa. Toisin kuin yleisemmissä kuljettajakorteissa ei yrityskorttihakemukselle automaattisesti liitetä varsinaista korttitilasta. Toisin sanoen, hakemuksen korttien oletusmäärä on nolla, kun se yleisempien ”kuljettajakorttien” kohdalla on aina yksi. Tämä toiminnallinen ero hämää käyttäjiä.



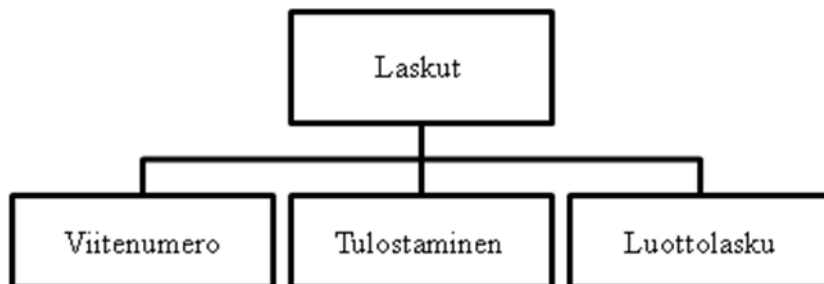
Kuva 13 PIIKOn yrityskorttien ongelmat

Kortteja on mahdollista lisätä hakemukselle selkeästi merkityllä ”Lisää” napilla, mutta kortteja ei ole mahdollista poistaa samankaltaisella toiminnallisuudella. Korttien poisto tapahtuu näytön eri osassa, hakemuksen kaikki kortit näyttävältä listalta. Listan oikeassa reunassa on ”ruksi” jota painamalla kortti poistetaan hakemukselta. Tietyissä tilanteissa tämä painike ei ole automaattisesti näkyvillä vaan käyttäjän on siirrettävä listaa oikealle vierityspalkin avulla jotta painike löytyy. Tämä hankaloittaa yrityskorttien käsittelyä verrattuna kuljettajakorttien käsittelyyn. Erityistä ongelmaa tämä aiheuttaa kun yrityskorttia hakevan yrityksen edustaja ei itse tiedä montako korttia hänen yrityksensä tarvitsee.

Koko yrityskorttihakemuksen toiminnallisuus on myös siitä erikoinen, että se ei ohjaa käyttäjää enää takaisin järjestelmän päänäytölle. Mikäli käyttäjä ei itse tiedä että hänen tulee hakemuksen tallentamisen jälkeen siirtyä navigointipolun avulla päänäytölle, ei hän pääse yrityskortti näytöltä enää eteenpäin. Tämä vaikeuttaa järjestelmän tämän toiminnallisuuden oppimista ja satunnaista käyttöä.

5.3.5. Laskut

PIIKO- järjestelmän avulla haettavat kortit ovat maksullisia, eikä kortin valmistusta aloiteta ennen kuin asiakas on suorittanut maksun. Näiden maksujen suorittaminen käteisellä tai pankkikortilla hakemuksen jättämisen yhteydessä onnistuu yleensä ongelmitta. Käyttäjillä on kuitenkin esiintynyt hankaluuksia laskujen muodostamisessa ja tulostamisessa.



Kuva 14 PIIKOn laskutuksen ongelmat

Etenkin heti järjestelmän käyttöönoton jälkeisenä aikana oli käyttäjillä usein vaikeuksia saada laskulle muodostumaan viitenumeroa. Koska viitenumero on hakemuksen ja suoritettun maksun yhdistävä tekijä, ei laskua saa tulostaa ilman sitä. Usein järjestelmä jäi kokonaan jumiin viitenumeroa muodostettaessa. Tällöin ratkaisuksi jäi käyttäjän uudelleen kirjautuminen järjestelmään ja hakemuksen ottaminen uudelleen käsittelyyn, mikä luonnollisesti vie aikaa asiakaspalvelutilanteessa. Pahimmassa tapauksessa käyttäjä on joutunut ilmoittamaan asiakkaalle että laskua ei ole juuri nyt mahdollista muodostaa, vaan se lähetetään postitse asiakkaan ilmoittamaan osoitteeseen. Tämä vie luonnollisesti aikaa, jolloin myös kortin saaminen asiakkaalle viivästyy saman verran.

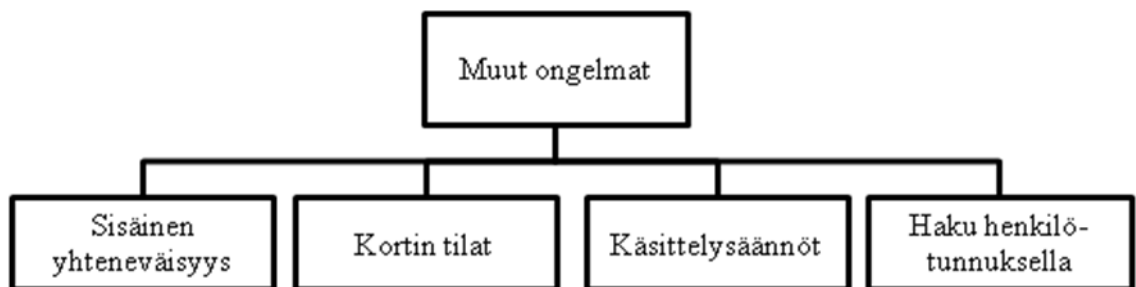
Tietyissä tilanteissa, erityisesti järjestelmän alkuaikoina, laskulle saatiin viitenumero mutta laskua ei jostain syystä ollut mahdollista tulostaa tai tulostus ”katosi” jonnekin, eikä koskaan saapunut tulostimelle. Mikäli tulostuksen uudelleen yritykset eivät onnistuneet, jouduttiin lasku lähettämään postitse asiakkaalle, kuten viitteen puuttumisen yhteydessä.

Luottolasku on erityistapaus. Viranomaisille myönnettävän ”*valvontakortin*” tuotanto voidaan aloittaa ilman että laskua on vielä maksettu. Tällöin viranomaiselle tulostetaan luottolasku. Luottolaskun kohdalla järjestelmä on kuitenkin toteutettu siten, että käyttäjä

ei mitenkään pääse näkemään varsinaista laskua. Tämä on johtanut ongelmatilanteisiin kun esimerkiksi osoitteet tai kortin haltijan tiedot ovat tulleet laskulle puutteellisesti tai väärin.

5.3.6. Muut PIIKO- järjestelmän käytettävyysongelmat

PIIKO järjestelmässä esiintyi myös muita käyttäjien toimintaan vaikuttavia käytettävyysogelmia. Yksi käyttäjien esiin nostama ongelma on järjestelmän sisäisen yhteneväisyyden puutteet. Esimerkkinä käyttäjät toivat esiin järjestelmän antamat ilmoitukset tallennuksien onnistumisesta. Toimitusosoitteen tallennuksen jälkeen järjestelmä ilmoittaa että tallennus onnistui, mutta maksutietojen tallennuksen jälkeen järjestelmä ei anna palautetta tallennuksen onnistumisesta.



Kuva 15 PIIKOn muut käytettävyysongelmat

Toinen tämän tyyppinen ongelma on eri aikaan kehitettyjen toiminnallisuuksien erot käsitellä samankaltaisia tilanteita. Syötettäessä raskaan kaluston ammattipätevyystodistuksia järjestelmään massasyöttönä, riittää kun käyttäjä syöttää oppilaiden henkilötunnukset toinen toisensa perään ja painaa rivinvaihtonäppäintä jokaisen henkilötunnuksen jälkeen. Taksien ammattipätevyystodistusten massasyötössä, eli periaatteessa identtisen toiminnon kohdalla, tulee jokaisen henkilötunnuksen syötön jälkeen ”tiedon vahvistus” ponnahdusikkuna. Kyseisen ikkunan tarkoitus on vähentää virhesyöttöjen määrää, pakottamalla käyttäjä tarkistamaan että henkilö tuli valittua oikein. Valitettavasti se toimii samalla käyttäjän totuttua toimintatapaa vastaan, hidastaen käyttäjän työtä.

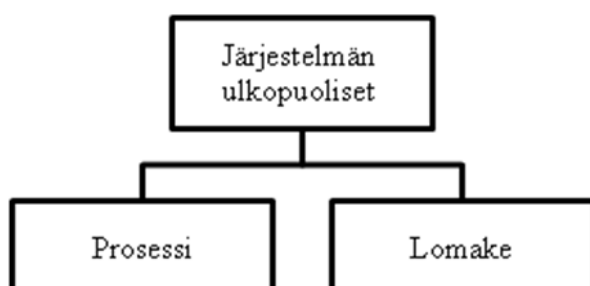
Korttien ”tiloilla” pyritään kuvaamaan käyttäjille missä vaiheessa hakuprosessia PIIKOn kautta haettavat kortit kulloinkin ovat. Tilojen nimet ovat kuitenkin monen käyttäjän mielestä epäselviä ja niihin toivottiin selkeytystä tai parempaa ohjeistusta.

Ammattipätevyyksien voimassaolon laskemisesta huolehtii järjestelmään rakennetut käsittelysäännöt. Käyttäjät ovat kuitenkin todenneet että käsittelysäännöt eivät ole luetettavia, vaan ne antavat ajoittain vääriä tuloksia. Ongelmaa pahentaa järjestelmän puutteellinen syötteiden tarkistus. On esimerkiksi mahdollista syöttää sisään koulutustietoja joissa koulutuksen hyväksymispäivämäärä on aikaisemmin kuin koulutuksen päättymispäivämäärä, minkä ei pitäisi olla loogisesti mahdollista. Tällaiset virheet syötteissä aiheuttavat lähes väistämättä myös virheellisen lopputuloksen käsittelysääntöjen toimesta.

Henkilötietojen haku henkilötunnuksella on käyttäjien mukaan ongelmallista PIIKO järjestelmässä. Haku on toteutettu siten että kun asiakkaan henkilötunnuksen tunnisteosa on kirjan, on hakukenttään kirjoitettava iso kirjain (eli A,B jne) jotta haku toimisi. Niissä tilanteissa joissa käyttäjät ovat tietoisia tästä, ongelma ei ole suuri, mutta se nostaa kuitenkin järjestelmän oppimiskynnystä.

5.3.7. Järjestelmän ulkopuoliset ongelmat

Haastatteluissa ilmeni myös että PIIKO- järjestelmän käyttöön vaikuttaa itse järjestelmän ulkopuoliset ongelmat, joita käsitellään tässä lyhyesti. Haastatteluissa kävi ilmi että käyttäjien mielestä hakemusprosessi on monimutkainen rutiineista poikkeavissa tapauksissa. Käyttäjille ei ole aina täysin selvää mitä kaikkia lomakkeita ja liitteitä tarvitaan jotta hakemus voidaan vastaanottaa. Esimerkiksi ulkomaalaisten hakijoiden kohdalla tarvittavat asiakirjat vaihtelevat sen mukaan onko hakija Suomessa asuva työluvan saanut henkilö, vai toisen EU maan henkilö joka sattuu asioimaan suomalaisessa toimipisteessä. Tällaisiin erikoistilanteisiin haastatellut pyysivät nykyistä parempaa ohjeistusta.



Kuva 16 PIIKOn järjestelmän ulkopuoliset ongelmat

Paperinen hakulomake sai myös käyttäjiltä kritiikkiä. Erityisesti hakijan allekirjoitus on haastateltujen käyttäjien mukaan toteutettu hankalasti. Lomake on allekirjoitettava nimenomaan 0.8mm mustalla tussilla lomakkeelle merkitylle viivalle, siten että nimikirjoitus ei saa ylittää viivaa ympäröivää ruutua. Asiakkaiden ongelmana on että sekä viiva että sitä ympäröivä ruutu ovat väriltään niin haalean sinisiä että niitä on vaikea erottaa. Lisäksi yllä mainittu vaatimus mustan tussin käytöstä ei ole riittävän selvästi esillä lomakkeella, vaan usein asiakkaat joutuvat täyttämään hakemuksen uudelleen vain koska allekirjoitus on kirjoitettu sinisellä kuulakärkikynällä. Tämä haittaa haastateltujen mielestä asiakaspalvelutilannetta valitettavan usein ja aiheuttaa närää asiakkaissa.

5.4. Järjestelmien yhteiset ongelmat

Kaikilla tarkastelun kohteena olevilla järjestelmillä on ongelmia järjestelmien nopeuden ja luotettavuuden kanssa. Lisäksi Liikenteen turvallisuusviraston ulkopuolisia käyttäjiä omaavilla järjestelmillä (REKI ja PIIKO) on ongelmia järjestelmien rajoitetun selaintuen kanssa.

VERO- ja REKI- järjestelmillä on ongelmia tietojen synkronoinnissa, mikä näkyy kyseisten järjestelmien virheherkkyytenä toisten järjestelmien tietoja käsiteltäessä. PIIKO- järjestelmässä ongelmaa ei juuri esiinny, koska PIIKO ei käsittele ajoneuvojen tietoja, vaan ainoastaan kyselee henkilöiden tietoja sekä koodeja tukijärjestelmistä.

Kaikissa järjestelmissä käyttäjän on tunnettava järjestelmän toiminta, jotta käyttäjän työ järjestelmän kanssa olisi mielekästä ja tehokasta. Järjestelmät eivät siis juurikaan ohjaa käyttäjiä toimimaan oikein eri tilanteissa. Navigointi järjestelmissä ei ole aina selkeää, eivätkä kaikki loogisesti samankaltaiset toiminnot toimi samalla tavoin järjestelmän sisällä, järjestelmien välisestä yhteneväisyydestä puhumattakaan.

Järjestelmiin liittyvät työprosessit ovat ajoittain monimutkaisia ja tämä vaikuttaa järjestelmiin rakennettuihin käsittelysääntöihin, jotka ovat tietyissä tapauksissa monimutkaiset ja virheelliset. Käyttäjien näkökulmasta ongelma on hankala, koska järjestelmän käsittelysäännön tekemää virhettä ei yleensä ole mahdollista muuttaa käyttäjän toimesta.

REKI- ja VERO- järjestelmien sisäiset tietomallit ovat monimutkaisia, mikä johtaa ”kaikki vaikuttaa kaikkeen” ongelmaan, eli tiettyyn tietoon tehty muutos saattaa aiheuttaa ennakoimattoman muutoksen johonkin muuhun tietoon. Tämä tekee virhetilanteiden korjaamisesta haastavaa ja avaa mahdollisuudet laajempimittaisiin ongelmiin, jos tehtävää muutosta ei pystytä testaamaan kunnolla. Käyttäjien kannalta tämä näkyy virheellisten tietojen säilymisenä pitkään järjestelmissä.

Yhteisenä ongelmana on myös järjestelmien käsittelemän tiedon määrä. REKI- ja VERO- järjestelmissä tilanne on jo akuutti eräajojen ja käyttöliittymän kautta tehtävien tapahtumien käsittelyaikojen venyessä niin pitkiksi että niille suunnitellut aikaikkunat eivät enää riitä. PIIKO- järjestelmässä on vasta viime aikoina törmätty tiedon määrästä johtuviin suorituskykyongelmiin, ja niitä on ratkottu tietokantakyselyjä parantamalla. Kaikissa järjestelmissä on kuitenkin edessä tietomäärän jatkuva kasvu, joten käyttäjillä on edessä entisestään hidastuvat järjestelmät mikäli asiaan ei keksitä ratkaisua.

Kun tietty ongelma esiintyy useammassa kuin yhdessä järjestelmässä, on syytä tutkia onko ongelman takana jokin systemaattinen ongelma. Tämä ongelma voi liittyä järjestelmien yhteiseen arkkitehtuuriin, organisaatioiden kehitysprosesseihin tai johonkin muuhun järjestelmien kehittämisessä vastaan tulevaan toiminta- tai ajatusmalliin. Näitä ongelmia pyritään selvittämään seuraavassa luvussa tutkimalla tässä luvussa käsiteltyjen käytettävyyso Ongelmien syntyhistoriaa.

6. Käytettävyysohjelmien syyt

Tässä osiossa esitellään järjestelmien tärkeimmät käytettävyysohjelmien ja näiden syntymiseen vaikuttaneet syyt. Kaikissa kolmessa tutkimuksen kohteena olevassa järjestelmissä on käyttäjän toimintaa haittaavia tai hidastavia ongelmia. Näistä käytettävyysohjelmista voidaan havaita seuraavia yhteisiä piirteitä:

- tietoja joudutaan käsittelemään usealla näytöllä
- tiettyjen tietojen muokkaaminen vaatii tietokannan käsin muokkaamista
- tiettyjen toimintojen liiallinen monimutkaisuus / vaikeus
- järjestelmien hitauden aiheuttamat katkot käyttäjien työhön

Ongelmien taustat vaikuttavat haastattelujen perusteella olevan yhtenevät tutkimuksen oletusten kanssa. Suurin osa niistä on mukana jo järjestelmän määrittelyssä.

6.1. VERO:n käytettävyysohjelmien synty

Tässä osassa esitellään VERO- järjestelmän ongelmien synty. Aloitan esittelemällä VERO- järjestelmän kehitysprosessin kulun ja ongelmien synnyn yleisellä tasolla, kuten olen sen haastattelujen ja järjestelmän kehitykseen liittyvän dokumentaation pohjalta ymmärtänyt. Tämän jälkeen tarkastelen tarkemmin yksittäisten ongelmien syntymiseen vaikuttaneita syitä.

Vero järjestelmä toteutettiin yhtenä PALKO hankkeen ensimmäisistä järjestelmistä. Järjestelmän ohjelmistokehitysprosessi oli perinteisen vesiputousmallin mukainen. Toteutunut kehitysprosessi voidaan jakaa seuraaviin vaiheisiin.

- Määrittely tilaajan toimesta (2000)
- Kilpailutus (1/2001)
- Määrittelyn tarkennus (2001)
- Suunnittelu ja toteutus (2001 - 2003)

- Uudelleen suunnittelu (2003)
- Suunnittelu, toteutus, testaus, tuotantoon otto (tuotantoon 1.1.2004)
- VERONA projekti (2004 - tuotantoon 11.12.2007)

Vero järjestelmä määriteltiin ensin tilaajan (Liikenteen turvallisuusvirasto) toimesta Tieto Oyj:n avustamana. Tämän perusteella syntyneen määrittelydokumentaation avulla projektin toteutus kilpailutettiin. Kilpailutuksen jälkeen määrittelyjä tarkennettiin yhdessä ja siirryttiin suunnitteluun. Suunnitteluvaiheessa toimittaja kuitenkin teki suunnittelua ja toteutusta pitkälti ilman tilaajalta saatua palautetta. Muun muassa tämä johti ristiriitoihin ja välien tulehtumiseen, minkä korjaaminen vei aikaa ja pakotti käymään suunnitelmat läpi uudelleen yhdessä. Tämän ”*uudelleen suunnittelun*” yhteydessä vaihdettiin myös toimittajan projektipäällikkö. Uudelleen suunnittelun jälkeen projekti eteni normaalin vesiputousmallin mukaisesti. VERONA projektissa muokattiin tuotannossa olevaa VERO- järjestelmää siten että se siirtyi käyttämään tietoja muista ATJ- järjestelmistä, jotka tulivat tuotantoon vasta tässä vaiheessa.

Suuri osa VERO- järjestelmään liittyvistä ongelmista johtaa haastattelujen perusteella juurensa järjestelmän määrittelyihin. Määrittelyt tehtiin tilaajalle uudella tavalla. Uutena asiana kummallekin osapuolelle oli määrittelyissä käytettävät UML kaaviot ja käyttötapaukset. Näitä tekniikoita opeteltiin niin tilaajan kuin toimittajankin osalta toimittajan mentoreiden kanssa. Tilaajalla ei ollut vielä olemassa valmista määrittelystandardia, eikä muutakaan arkkitehtuuria ja työtapoja ohjeistavaa dokumentaatiota, joten työn pohjana käytettiin toimittajan omaa kehitysmallia. Standardien puute vaikutti käyttötapausten tekoon, erityisesti toteutustapojen ja liittymien osalta. Käyttötapaukset kirjattiin niin hyvin kun osattiin, mutta ohjeiden puuttuessa käyttötapaukset eivät olleet kaikki riittävän tarkkoja tai kattavia. Myöskään järjestelmän kokonaisuutta ei hahmoteltu riittävästi, vaan keskityttiin nimenomaan yksittäisten käyttötapausten määrittelemiseen.

Haastattelujen perusteella tunnelma oli määrittelyvaiheen alussa toimittajan puolella hyvä ja työtä tehtiin hyvässä yhteistyössä tilaajan kanssa. Tilaajan vanhasta tottumuksesta käyttötapauksia tehtiin teknisesti tarkalla tasolla, siten että niissä oli jo määrittelyvaiheessa mukana muun muassa tietokannan kentänimiä.

Prosessinäkökulmasta käyttötapauksia ei mietitty yhtä tarkasti, koska prosessituntemus oli tilaajalla ja hankalasti kirjattavissa eksplisiittisesti. Käyttötapauksille ei tehty pöytä-eikä käytettävyydestä. Järjestelmän näytöistä ei ollut vielä näkemystä vaan käyttötapauksiin listattiin pelkästään halutut tiedot ja mitä niillä pitää voida tehdä.

Kilpailutuksen yhteydessä toimittaja ilmoitti jälkikäteen ajatellen urakalle turhan optimistisen työmäärä-arvion. Koska kilpailutuksen säännöissä projektin kustannuksilla oli suuri painoarvo, voitti Tieto kilpailutuksen. Seuraukset tästä olivat kahden tasoiset. Ensinnäkin projektiin varattiin, niin toimittajalla kun tilaajalla, selkeästi liian vähän aikaa ja resursseja. Toiseksi tilaaja ”tiesi” nyt kuinka paljon tämän kokoluokan ohjelmistokehitys maksaa ja kestää, mikä vaikutti kaikkien tätä seuraavien projektien kilpailutuksien alkulähtökohtiin.

Kilpailutuksen jälkeen toimittaja aloitti järjestelmän suunnittelun ja toteutuksen. Tässä vaiheessa realisoituivat liian alhaiseksi säädetyt arviot työmääristä ja aikatauluista. Jotta arviot saataisiin pitämään paikkansa, joutui toimittaja tekemään paljon suunnittelua ja toteutustakin ilman riittävää yhteistyötä tilaajan kanssa. Tämä johti ristiriitaihin siitä minkälaiseksi järjestelmä ja sen näytöt oli mielletty määrittelyissä. Tilaajan reagoidessa tilanteeseen hitaasti pääsivät ristiriidat kärjistymään, ja lopulta ajauduttiin tilanteeseen jossa jouduttiin palaamaan järjestelmän uudelleen määrittelyyn. Samassa yhteydessä vaihdettiin toimittajan projektipäällikkö ja lisättiin yhteistyötä suunnittelussa.

Suunnittelu ja toteutus tehtiin pitkälti määrittelyissä syntyneiden luokkamallien pohjalta. Näyttöjen sopivuutta ajoneuvoverotuksen prosesseihin ei tarkasteltu erityisesti, vaan näytöt ja järjestelmän toiminnot kehitettiin puhtaasti järjestelmäkehitysnäkökulmasta, eikä prosessikehitysnäkökulmaa juurikaan huomioitu. Näyttöjen suunnittelussa ei ollut vielä olemassa käyttöliittymästandardia, joten tietojen esitykseen ja syöttöön jouduttiin etsimään projektikohtaisia käyttöliittymäratkaisuja. Käyttötapauksista tehtiin toimittajan toimesta sisäisesti tarkempia nk. teknisiä käyttötapauksia ja käsittelysääntöjä. Nämä tekniset käyttötapaukset olivat sillä tasolla että niistä saatiin aikaiseksi varsinaista ohjelmakoodia.

Tiettyihin järjestelmän ratkaisuihin päädyttiin koska toimittajan toteuttajat sanoivat että jotkin asiat, kuten haun käynnistys enter- painikkeella, eivät olleet teknisesti mahdollisia annetun arkkitehtuurin piirissä. Myöhemmin näitä ratkaisuja on kuitenkin toteutettu

muissa saman arkkitehtuurin alaisissa projekteissa. Tekniikkaa ja sen rajoituksia opeteltiin niin tilaajalla kuin toimittajalla. Liian monella toteuttajalla ei kuitenkaan ollut riittävästi kokemusta tämän suuruusluokan projektiin. Osa toteuttajista tuli suoraan koulun penkiltä ja osa vasta opetteli käytettyä tekniikkaa. Tekniikan tuntemattomuus aiheutti epävarmuutta kaikissa osapuolissa.

Koska toteutus oli vahvasti käyttötapaus pohjaista, ei yhteisiä palveluja ja toimintoja aina huomattu vaan ne toteutettiin useaan kertaan, esim. maksuunpano toiminto koodattiin alun perin viiteen kertaan. Myös tietojen haku muodostui tehottomaksi kun käyttötapausten hakutarpeita ei käsitelty kokonaisuutena.

Tilaajan ohjeistuksen ja erillisen tietoarkkitehdin puute näkyi yhtenäisten linjauksien puuttumisena ja arkkitehtuuriin vaikuttavien päätösten hitaudessa. Esimerkiksi eräajojen toteutustapa saatiin kiinnitettyä vasta varsin myöhään projektissa. Lisäksi tilaajan tietohallinnosta painotettiin että suuria massoja ei tule, koska ajot jaksottuvat pitkin vuotta.

Aikataulun riittämättömyydestä kertoo se että testausvaiheessa ei virheiden saturaatiopistettä tunnutta saavutettavan millään, vaan joka testikierroksella löytyi aina uusia virheitä. Kiire, yhteisen ohjeiston puute ja kokemattomuus johti myös joihinkin ongelmallisiin ratkaisuihin toteutuksessa. Esimerkiksi osa asioista on kovakoodattu järjestelmään vaikka alkuperäisenä tarkoituksena oli että nämä tiedot haetaan sopivasta koodistosta. Korjauksia on hankaloittanut kiireen, työtapojen (mm. ei koodikatselmuksia) ja kokemuksen puutteen aiheuttama koodin puutteellinen kommentointi ja dokumentointi.

6.1.1. Pitkien listojen toiminta

Pitkien listojen esittämiseen liittyviin ongelmiin ei osattu määrittely- ja suunnitteluvaiheessa varautua. Aiemmassa VERO- järjestelmässä pitkät listat toimivat ”itsestään” siten, että haetaan tietty määrä osumia ruudulle, jonka jälkeen käyttäjä pystyi pyytämään seuraavan ruudullisen osumia. Nyt käyttötapausten määrittelyssä ei erikseen vaadittu tämän tyylistä pitkien listojen käsittelyä. Listojen toiminta toteutettiin ilman erillistä suunnitelmaa pitkien listojen käsittelyä varten. Ongelmiin havahduttiin vasta kun testattavaa materiaalia oli niin paljon että listoille

alkoi tulla enemmän rivejä kuin mitä järjestelmä pystyi käsittelemään. Tällöin tilaajalta pyydettiin keskitettyä ja yleistettävää arkkitehtuuriratkaisua pitkien listojen toteutukseen. Ratkaisua ei kuitenkaan saatu pitkään aikaan. Kun ratkaisu vihdoinkin saatiin, projekti oli jo niin pitkällä ja järjestelmän kiinnitetty tuotantoonsiirto oli niin lähellä, että merkittäviä muutoksia ei ollut mahdollista enää tehdä vaarantamatta ajoneuvoverolain muutokseen sidottua aikataulua.

VERO- järjestelmän toteutuksen aikana ei ollut vielä olemassa kunnan käyttöliittymästandardia, joka olisi määritellyt miten esimerkiksi listojen sivuttamista tai niiden sisällön dynaamisesta järjestämisestä olisi ollut mahdollista toteuttaa. Tämä esti siten pitkien listojen käsittelyn kokonaan niissä tapauksissa kun koko listan hakeminen kestää pitempään kuin ATJn yleisen kahden minuutin ”timeout” rajan. Lyhyemmät listat saadaan näkyviin, mutta niiden käsittely on työlästä, koska listojen sisältöä ei ole mahdollista dynaamisesti järjestää työprosessia tukevalla tavalla.

Erityisesti nämä ongelmat tulivat esiin ”Yhteislasku” toiminnallisuuden kohdalla. ”Yhteislasku” toiminnallisuuden tarkoituksena on mahdollistaa yrityksille niiden omistamien ajoneuvojen verojen maksamisen yhdellä laskulla, sen sijaan että jokaisen ajoneuvon vero maksettaisiin omalla, erillisellä laskulla. Koska suurilla yrityksellä voi olla omistuksessa tuhansia ajoneuvoja, tulisi niitä olla mahdollista käsitellä yhdessä transaktiossa ja listassa. Nyt kuitenkin listan muodostaminen ei onnistu yllä mainitun ”timeout” rajan puitteissa, eikä listojen osittaiseen muodostamiseen ole luotu mitään toiminnallisuutta. Tämä tarkoittaa että tiettyjen yritysten kohdalla laskut on muodostettava osin käsin.

6.1.2. Tietojen synkronointi-ongelmat

Tietojen synkronointi-ongelmien juuret löytyvät ATJ- järjestelmien määrittelyvaiheesta. Koska ATJ- järjestelmät luotiin mahdollisimman pitkälti erillisinä kokonaisuuksina, on niiden integraatiossa ollut haasteita. Liittymien määrittelyt on ymmärretty eri tavoin eri projektien henkilöiden toimesta ja projektien välinen kommunikaatio ei ole ollut yhtä tehokasta kuin projektien sisäinen.

Yhtenä tietojen synkronoinnin perustavana ongelmana on järjestelmien eri tavat käsitellä tietoa. VERO- järjestelmässä tietojen nykytilanne on historiatietojen summa, kun taas

REKI- järjestelmässä, jonka tiedoista VERO- järjestelmä on riippuvainen, nykytilanne on aina tallennettuna. Tämä on pyritty ratkaisemaan siten että REKI- lähettää, ja VERO- vastaanottaa, kaikki ajoneuvojen muuttuneet tiedot ja VERO- päättelee niiden perusteella mikä VEROn sisäinen muutostyyppi on kyseessä.

Tietojen synkronoinnin määrittely, suunnittelu ja toteutus, vaati yhteistyötä eri tietojärjestelmäprojektien välillä, mikä ei aina ollut kitkatonta. Tämä korostui erityisesti koska Liikenteen turvallisuusviraston organisaatio on perinteisesti toiminut siilomaisesti, mikä on havaittavissa myös tietojärjestelmäprojektien organisoinnissa. Asioiden läpikäynti yhdessä auttoi kuitenkin ymmärtämään miksi ja mitä tietoja tarvitsee välittää järjestelmien välillä. Kuitenkin ongelmana on, että eri järjestelmät eivät aina sisäistä mitä seurauksia toisille järjestelmille on ”oman” järjestelmän tietojen ja liittymien pienistäkin muutoksista. Erityisesti ongelmia aiheuttavat tapaukset joissa yhteen tietoja välittävään sanomaan on sidottu liian paljon tietoa. Ongelmaa pahentaa tietojen tarkistuksien vähäisyys, joka johtaa suurempaan määrään virheellisiä sanomia, jotka aiheuttavat ongelmia vastaanottavissa järjestelmissä.

Tietojen käsittelyperiaatteiden aiheuttamien ongelmien lisäksi tilannetta hankaloittaa VERO- järjestelmän perustavanlaatuinen tarve pystyä käsittelemään tapahtumia niin käyttöliittymän kautta syötettynä, kuin järjestelmien välisen liittymän kautta tullessa. Tämä pakotti miettimään miten varmistetaan, ettei samaa tietoa käsitellä samanaikaisesti eri tahoilta. Ratkaisu löytyi tietokannan tietojen lukituksissa, eli siitä miten tiedon käsittely on varattu yhdelle taholle kunnes tämä vapauttaa tiedon jälleen käsiteltäväksi. Ongelmaksi jäi kuitenkin miten varmistetaan että tiedot tulevat järjestelmään oikeassa järjestyksessä. Tietojen järjestys on erityisen hankalaa varmistaa tilanteissa joissa sanomajonoihin on kertynyt paljon tietoja, ja niitä pitäisi lukea sisään usean sanomajonokuuntelijan voimin samanaikaisesti. Pahimmassa tapauksessa järjestelmän käyttäjä vielä käsittelee tilannetta käyttöliittymän kautta samaan aikaan. Koska tietojen väärä järjestys saattaa aiheuttaa ajoneuvon tilan muuttumisen väärin perustein, ja voi pahimmillaan vaatia kantapuukotuksen tilanteen korjaamiseksi, on kyseessä vakava ongelma. Osittainen ratkaisu on löytynyt siitä että ongelmatilanteiden jälkeen sanomajonot voidaan purkaa yhden kuuntelijan avulla. Silloin liittymän kautta tulleet tiedot pysyvät samassa järjestyksessä kuin ne on lähetetty. Liittymän kautta tulneiden

tietojen käsittelyn hankaluudesta huolimatta on varsinainen sanomakäsittelytekniikka kuitenkin todettu teknisesti toimivaksi.

6.1.3. Eräajojen hitaus ja kaatuilu

Eräajojen ajoaikojen arviointi ei ollut helppoa määrittelyvaiheessa, vaikka järjestelmään vaikuttavien tapahtumien määrät oli aika hyvin tiedossa. Yhtenä syynä tähän oli edeltävän järjestelmän täysin eri tekniikka, jolloin eräajot eivät ole suoraan vertailukelpoisia. Tilaajan valitsema tekniikka (J2EE, DB2 tietokannat) mahdollisti vastaavanlaiset eräajot kun mihin oli aiemmin totuttu, mutta suorituskyky ei ollut yhtä hyvä.

Koska järjestelmä toteutettiin pitkälti käyttötapaus kerrallaan, ei eräajojen toimintaa tullut tarkasteltua kokonaisvaltaisesti ennen suorituskykytestejä. Eräajojen suorituskykyä testattiin, kuten vesiputousmalliin kuluu, vasta projektin loppuvaiheilla kun riittävän suuri, vanhasta tuotantodatasta konvertoitu testiaineisto oli käytettävissä. Kun ongelmiin havahduttiin, ei enää ollut riittävästi aikaa hakea kokonaisvaltaisia ratkaisuja eräajojen suorituskykyongelmiin. Tämä ongelma esiintyy myös järjestelmän muissa tietokantaoperaatioita käyttävissä toiminnallisuuksissa. Tietokantaoperaatiot on tehty yksittäisten käyttötapausten ehdoilla, eikä niitä käyty läpi kokonaisuutena, mikä olisi saattanut johtaa parempiin ratkaisuihin niin operaatioiden kuin itse tietokannankin osalta. Tämä vaikuttaa myös itse tietokannan ominaisuuksiin, kuten taulujen indekseihin, joita jouduttiin muokkaamaan pitkin projektia. Tietokannan ominaisuuksilla on luonnollisesti kerrannaisvaikutuksia kaikkiin tietokantaoperaatioihin.

VERO- järjestelmän sisältämä tietomäärä on suuri, ja sen tiedettiin jo määrittelyvaiheessa kasvavan entisestään. Varsinaisia ratkaisuja ei aineiston myötä kasvaviin eräajojen kestoihin kuitenkaan toteutuksen aikana ollut, vaan se jätettiin myöhemmin ratkaistavaksi. Ongelmaa hankaloittaa se, että järjestelmästä ei ole käytännössä mahdollista poistaa mitään, mikä johtuu sen monimutkaisesta sisäisestä logiikasta. Tietojen elinkaariajattelua ei käytetty riittävästi projektin aikataulurajoitusten vuoksi.

Eräajojen toiminnallisuus liittyy myös järjestelmän toiminnallisuuksiin. Esimerkiksi tulosten esi- ja jälkikatselumahdollisuus vie kannasta paljon tilaa, mikä vaikuttaa

läpikäytävään massaan, hidastaen siten eräajoja. Koska eräajot käsittelevät noin 99% kaikista tapauksista, lisää niiden virheet suoraan käyttäjien toimesta käsiteltävien tapausten määrää. Tämä tekee eräajojen toiminnasta erittäin tärkeän osan järjestelmästä saatavaa käyttökokemusta.

6.1.4. VERO:n käyttöliittymän käytettävyysongelmat

Jo alun perin oli tiedossa että uusi selainpohjainen järjestelmä tulisi olemaan käyttäjien mielestä hitaampi, kuin mihin edistyneet käyttäjät olivat tottuneet. Syyksi tiedostettiin ainakin edeltävän, merkkipohjaisen, järjestelmän suuri nopeus, tiivis tiedon esitys koodimuodoissa sekä suuri määrä pikanäppäimiä. Toisaalta nämä ominaisuudet tekivät edeltävästä järjestelmästä hankalan oppia, tilanne joka haluttiin välttää uuden järjestelmän osalta. Valitettavasti selainpohjaisuus esti joidenkin vanhan järjestelmän hyvien puolten, kuten pikanäppäinten, käyttämisen uudessa järjestelmässä.

Koska järjestelmä on vahvasti eräajopohjainen, keskityttiin toteutuksessa ensin eräajojen vaatimien toiminnallisuuksien toteuttamiseen. Näytöt toteutettiin suhteellisen myöhään, jolloin näyttöjen nopeuden optimoimiseen ei ollut enää mahdollista käyttää riittävästi resursseja tai aikaa.

Käyttöliittymän toteutuksen aikana ei ollut käytettävissä valmista käyttöliittymä-standardia, vaan se kehittyi projektin aikana. Osaksi tämän, ja osaksi käyttötapauksiin painottuvan toteutustavan, vuoksi käyttöliittymäelementit toteutettiin usein käyttötapauskohtaisesti mikä ei mahdollistanut niiden toiminnan optimoimista vaan johti ajoittain toiminnallisuuksien kahdentumiseen.

Käyttöliittymän toteuttajat sanelivat millaisia käyttöliittymäelementtejä oli yleensä mahdollista toteuttaa. Myöhemmissä järjestelmissä on ollut mahdollista käyttää monimutkaisempia ratkaisuja jotka nopeuttavat käyttäjän kokemaa toimintaa. VERO-järjestelmän käyttöliittymän nopeutta on parannettu hieman ajan saatossa.

6.1.5. Maksujärjestely- toiminnallisuus

Ajoneuvoverojen käsittelyssä käsitellään maksuja rutiininomaisesti. Tämän vuoksi maksuihin liittyvien toiminnallisuuksien tulisi olla toimivia ja käytettäviä. Näin ei

kuitenkaan ole. Maksujärjestely- toiminnallisuuden tarkoituksena oli mahdollistaa ajoneuvojen verojen maksamisen järjestely eri syiden perusteella joustavasti järjestelmän toimesta. Näitä syitä on muun muassa, ajoneuvo on yrityksen omistuksessa tai veron maksajana toimii muu kuin varsinainen ajoneuvon omistaja/haltija. Nyt kyseisen toiminnallisuuden käytöstä on luovuttu koska se on todettu liian hankalaksi ja epäluotettavaksi käyttää, jolloin nämä tapaukset käsitellään nyt enimmäkseen käsin.

Maksujärjestely- toiminnallisuutta ei määritelty eikä suunniteltu riittävän tarkalla tasolla projektin alkuvaiheessa, eikä niitä suunnitelmia ehditty tarkentaa annetussa aikataulussa. Maksujärjestelyjen ollessa mukana useassa käyttötapauksessa, ei ollut selvää miten kyseinen toiminnallisuus toteutettaisiin keskitetysti. Maksujärjestely-toiminnallisuus onkin edelleen toteutettuna järjestelmässä pitkälti alkuperäisessä asussaan, ilman kunnan kokonaisvaltaista suunnittelua. Joitakin näihin liittyviä ongelmia on toki korjattu jälkikäteen, joskin dokumentaation puute ja tiettyjen toiminnallisuuksien kovakoodaus vaikeuttavat korjauksien toteuttamista.

Maksujärjestely- toiminnallisuuksiin kuuluva ”maksuunpano” toiminnallisuus on ehkä räikein esimerkki työmäärien aliarvioinnin aiheuttamasta turhasta työstä. Kun työmäärät arvioitiin alakanttiin, ei jäänyt aikaa tarkastella kokonaisuutta laajemmin vaan järjestelmän toiminnallisuudet toteutettiin käyttötapauksittain eri tekijöiden toimesta. Koska ”maksuunpano”-toiminnallisuus esiintyi useassa käyttötapauksessa, oli lopputuloksena se että toiminnallisuus koodattiin alun perin *viisi* kertaa. Ongelmaan ei havahduttu koska toimittajan toteuttajilla oli kiire eivätkä eri osa-alueiden tekijät huomanneet että heillä kaikilla oli työlistalla sama toiminnallisuus. Maksuunpanoon kohdistuneet korjaukset jouduttiinkin toteuttamaan aina jokaisen toiminnallisuuden ”kopioon” erikseen, mikä luonnollisesti on aiheuttanut paljon ylimääräistä työtä. Toiminnallisuuksia on pyritty yhdistämään yhdeksi komponentiksi ajan kanssa mutta tämä on ollut työlästä.

Toisena ”maksujärjestely” toiminnallisuuksien ongelmana voidaan mainita laskuihin liittyvä viitenumeroiden käsittely. Järjestelmässä viitenumeroita voidaan määrätä tietylle ajoneuvolle enintään 9 kpl / vrk. Tämä johtuu siitä että standardin muotoisen viitenumeron pituusrajoituksen vuoksi sille sopii muiden tietojen jälkeen enää yksi numero laskun juoksunumeroksi. Sen vuoksi takautuvia korjauksia tehtäessä, voidaan

korjata enintään yhdeksän tapahtumaa per ajoneuvo per vuorokausi, jonka jälkeen on tapaus laitettava hyllylle, ja jatkettava seuraavana päivänä. Tämä ongelma oli kuitenkin jo olemassa nykyjärjestelmän edeltäjässä, eikä siihen silloin reagoitu. Tilaaja ei myöskään ollut nostanut tätä esiin määrittely- ja suunnitteluvaiheessa, vaikka viitenumerokäsittely määriteltiin tilaajan toimesta. Ongelman syntyä on edistänyt vesiputousmallin sanelema marssijärjestys, resurssien vähyys ja prosessikehityksen pitäminen erillään järjestelmäkehityksestä. Vesiputousmallin seuraaminen tarkoittaa testauksen sijoittumista projektin loppuun, jolloin myös kunnollisen ja riittävän laajan testimateriaalin saatavuus jää projektin loppuvaiheille. Tällöin monimutkaisten ongelmien, kuten yllä mainitun viitenumerojen käsittely-ongelman, havaitsemiseen jää vain vähän aikaa testausvaiheessa, eikä kattavia korjauksia ole enää mahdollista tehdä. Projektin aiemmissa vaiheissa taas resurssipulan aiheuttama kiire esti tulevan ongelman havaitsemisen koska kokonaisuutta ei tarkasteltu riittävästi. Järjestelmä- ja prosessikehityksen pitäminen erillään puolestaan esti ongelman havaitsemisen prosessikehityksen kautta. Mikäli samaan aikaan, tai ennen järjestelmäkehityksen alkua, olisi käyty läpi silloisen järjestelmän ja prosessin ongelmia yhdessä, saatettaisiin tällaiset ongelmat havaita ajoissa.

VERO- järjestelmään ohjelmoidun verojen ja maksujen laskentaprosessin mutkikkuus aiheuttaa myös ongelmia maksujen yhteydessä. Eräs ongelmia aiheuttava ratkaisu on että jokaisen ajoneuvon/henkilön verosaldo lasketaan koko ajan uudestaan aiemmin tulleista tapahtumista. Tämä pakottaa järjestelmän käymään läpi suuren tietomäärän aina nykytilanteen määrittämiseksi, sen sijaan että nykytilanteeseen päivitetään muuttuneet tiedot, tai että poikkeustilanteissa jonkin ajoneuvon tilanne lasketaan alusta asti uudelleen.

6.1.6. Työtehtävien käyttäjäroolit

VERO- järjestelmään alun perin suunnitellut ja toteutetut käyttäjäroolit eivät enää vastaa todellisuutta ja prosessin tarpeita. Tämä on johtanut siihen että käyttäjillä joudutaan ajoittain myöntämään huomattavan laajoja oikeuksia järjestelmään, jotta he voisivat tehdä työtään tehokkaasti.

Mikäli prosessi- ja järjestelmäkehitystä olisi tehty samanaikaisesti, olisi tällaiset tilanteet ehkä pystytty välttämään. Ongelman olisi ehkä voinut kiertää osittamalla järjestelmän käyttäjäroolit eri lailla, tai muokkaamalla prosessia eri suuntaan kuin mihin nyt on päädytty.

6.1.7. Yhteenveto VERO:n ongelmien synnystä

Ylivoimaisesti suurin osa VERO- järjestelmän käytettävyysongelmista voidaan todeta saaneen alkunsa jo järjestelmän määrittelyvaiheessa. Tilaajan tekemät valinnat ovat taustalla suuressa osassa ongelmia. Valittu toteutustekniikka rajoittaa jonkin verran, joskaan ei ratkaisevissa määrin, eräajojen tehokkuutta sekä käytettävissä olevien käyttöliittymän osien määrää. Vesiputousmallin seuraaminen puolestaan aiheutti ongelmia usealla alueella, lähinnä riittävän laajan testauksen ja testiaineiston myöhäisen sijoittumisen vuoksi jolloin virheitä ei enää ehditty korjaamaan. Tämän tyyppisen, tiettyyn päivämäärään sidottuun lakimuutokseen liittyvää ja suurta sisäistä sidonnaisuutta sisältävän, järjestelmän toteuttaminen esimerkiksi iteratiivisten menetelmien avulla vaatisi kuitenkin erittäin vahvaa kyseisen menetelmän osaamista ainakin toimittajalta mutta todennäköisesti myös tilaajalta.

Tilaajalla ei ollut palveluksessaan täysipäiväistä tietoarkkitehtiä eivätkä ohjeistukset olleet täysin valmiita projektiin lähettäessä. Arkkitehtuurin ja muun ohjeistuksen epäyhtenäisyys ja niihin tulleet projektin aikaiset muutokset aiheuttivat päänvaivaa sekä myöhästelyjä kun toimittaja odotti tilaajalta arkkitehtuurilinjauksia. Täysipäiväistä arkkitehtiä olisikin tarvittu ongelmien ratkaisemiseen ja arkkitehtuuristen linjojen määrittämiseen.

Tilaajan tottumus organisaation siilomaisuuteen heijastui myös projektien väliseen kommunikaation. Koska projekteja ajateltiin pitkälti erillisinä organisaatioina, ei näiden välinen kommunikaatio ollut vaivatonta tai välitöntä. Nämä kommunikaatio-ongelmat vaikuttavat järjestelmien välisten liittymien rakentamiseen ja toimivuuteen.

Toimittajan suurin virhe lienee ollut turhan optimistinen työmäärä-arvio. Tämä johti kahdenlaiseen ongelmaan. Ensinnäkin VERO- järjestelmän toimitusprojekti vaati loppujen lopuksi toimittajan asiantuntijoiden alun perin arvioiman työmäärän tai jopa sen yli, mikä luonnollisesti aiheutti ongelmia tilaaja-toimittaja suhteissa sekä

aikatauluissa ja toteutuksen laadussa. Toiseksi tilaaja sai turhan optimistisen kuvan tietojärjestelmien kehittämisen kustannuksista koska ensimmäisen järjestelmän toimittaja oli antanut turhan optimistisen työmäärä-arvion.

Projektiin lähdettiin siis vääristä lähtökohdista resursoinnin, aikataulutuksen ja kustannusten suhteen. Yhteistyötä ei alkuvaiheessa tehty riittävästi tilaajan ja toimittajan välillä, vaan toimittaja teki paljon asioita omatoimisesti. Tilaaja puolestaan ei reagoanut epäkohtiin tarpeeksi nopeasti ja voimakkaasti. Toimittaja koulutti henkilöstöään samanaikaisesti toteutuksen kanssa, mutta ei varannut siihen tarpeeksi aikaa. Lisäksi toimittajan toteuttajat opettelivat vasta monille uutta tekniikkaa. Toimittajan käyttämät kuvausmenetelmien opettajat vaihtuivat, jolloin myös tietyt ohjeistukset vaihtuivat. Toteuttajat koottiin projektiin toimittajan eri toimipisteistä, jolloin heillä oli aluksi omat työskentelytapansa. Myöhemmin toimittaja sai mukaan teknisen arkkitehdin koordinoimaan työtä, mutta tämä tapahtui turhan myöhään.

Projektin edistyessä toimittaja paransi resursointia ja tilaaja ohjeistuksiaan. Alun puutteelliset ohjeet ja resursoinnit, ajoittain heikot tilaaja - toimittaja suhteet sekä yleinen kiire tuntuivat kuitenkin läpi koko projektin ja projektissa mukana olleet kokivat projektin erittäin raskaaksi.

6.2. REKIn käytettävyyso Ongelmien synty

Tässä osassa esitellään REKI- järjestelmän käytettävyyso Ongelmien synty. Ensin esitellään kehityso Ongelmien synty yleisellä tasolla, minkä jälkeen tarkastellaan tarkemmin yksittäisten ongelmien syntyso Ongelmien synty.

REKI- järjestelmä toteutettiin yhtenä PALKO hankkeen pääasiallisista järjestelmistä. Järjestelmän ohjelmistokehityso Ongelmien synty oli perinteisen vesiputousmallin mukainen. Toteutunut kehityso Ongelmien synty voidaan jakaa seuraaviin vaiheisiin.

- Määrittely tilaajan toimesta (-2002)
- Kilpailutus (2002)
- Määrittelyn tarkennus (2002)

- Suunnittelu ja toteutus (2002-2006)
- Testaus ja tuotantoon otto (2006-2007)

REKI- järjestelmän toimittajana oli Logica Suomi Oy.

6.2.1. Rekisteröintien korjaukset ja takautuvat muutokset

REKI- järjestelmään tallennettujen ajoneuvojen tietojen korjaukset ovat ongelmallisia. Tietyissä tapauksissa tietojen muokkaaminen on aiheuttanut arvaamattomia vaikutuksia ajoneuvon nykyisiin tai aiempiin tietoihin, ja joskus muutoksia on tapahtunut jopa täysin tapahtumaan liittymättömiin ajoneuvoihin. Erityisen hankalia ovat olleet tapaukset joissa muutoksia on tehtävä takautuvasti, eli niillä on vaikutuksia myös varsinaista muutettua tapahtumaa seuranneisiin tapahtumiin.

Ongelmien syitä tutkittaessa kävi ilmi että nämä kaksi ongelmaa ovat kytköksissä toisiinsa, joten käsittelen niiden syntymekanismeja yhdessä. Kummankin ongelman suoranainen syy on REKI- järjestelmän monimutkaiset ja pitkät päättelyketjut ja käsittelysäännöt. Tämä monimutkaisuus puolestaan johtuu useammasta tekijästä.

Eräs perussyistä tuntuu tässäkin järjestelmässä, olevan tilaajan teknisen arkkitehdin puute projektin alkuvaiheessa. Kenelläkään ei ollut riittävän hyvää teknistä yleiskuvaa järjestelmästä, jotta esiin tuleviin ongelmiin saataisiin ratkaisuja nopeassa aikataulussa ja ennen kaikkea keskitetysti. Olemassa ollut ohjeisto ei ollut riittävän laajaa ja tarkkaa näin laajojen projektien läpiviemiseen ilman projektin aikaista tarkennusta. Nyt esiin tuleviin ongelmiin jouduttiin hakemaan ratkaisuja monen eri toimijan kautta, mikä vei kalenteriaikaa ja siten pienensi mahdollisuuksia toteuttaa tarvittavia muutoksia.

Tilaaajaorganisaation siilorakenne heijastui tietojärjestelmäkehityksessä tietojärjestelmäprojektien varsin itsenäisessä toiminnassa. Tässä tilanteessa arkkitehdin puuttuminen mahdollisti määrittelymateriaalin projektikohtaisen tulkitsemisen, mikä johti samojen tietojen erilaiseen esittämistapaan eri järjestelmien tietokannoissa, tietokentissä sekä näytöillä. Kun näitä tietoja on tarve käyttää ristiin järjestelmien välillä, ajaututaan helposti ongelmallisiin tilanteisiin, kun haluttu tieto ei olekaan oletetussa muodossa.

ATJ- järjestelmien monimutkainen tietomalli on myös osasyynä rekisteröintien korjausten ja takautuvien muutosten ongelmiin. Tietomalli sisältää usean järjestelmän käyttämiä yhteisiä tietoja sekä pelkästään jonkin yksittäisen järjestelmän käyttämiä tietoja. Kun REKI- järjestelmän käyttämien tietojen väliset yhteydet ovat monimutkaiset, päädytään tilanteeseen jossa yhden tiedon muuttamisen seuraamuksia ei pystytä ennakoimaan tai edes tutkimaan pelkästään yhtä järjestelmää tarkastelemalla. Tällöin tietojärjestelmän käsittelysääntöjen ohjelmoiminen toimiviksi ja kaikki tilanteet kattaviksi muuttuu erittäin hankalaksi.

Käsittelysääntöjen monimutkaisuutta lisää historialliset seikat. REKI- järjestelmän tiedot konvertoitiin vanhasta Liikennetietojärjestelmästä (LTJ). LTJ ei ole tietosisällöltään ehjä järjestelmä, jollaiseksi kuitenkin REKI haluttiin. Tämän vuoksi konversiossa luovuttiin osasta historiatiedoista jotta saataisiin aikaiseksi tietosisällöltään riittävän ehjä alkutilanne REKIn toimintaa varten. Absoluuttinen tietojen alkupiste tarkoittaa yhtä ylimääräistä erikoistilannetta josta käsittelysääntöjen on selvittävä.

Tarjouskilpailun materiaalissa oli mukana vaatimus korjauksista ja muutoksista. Valitettavasti vaatimus oli tasoa ”*Kaikkea on voitava korjata*”, mikä on niin laaja että sen toteuttaminen vaatisi huomattavan määrän tarkempaa määrittelyä jotta toteutus ei olisi ylivoimaista. Korjausvaatimuksen purkaminen auki paremmin käyttötapauksille olisi saattanut auttaa ongelmien huomaamisessa. Nyt ongelmat tulivat havainnollisesti esiin vasta integraatiotestausvaiheessa, jolloin päästiin käyttämään järjestelmiä ensi kerran kunnolla ristiin. Vesiputousmallin mukaan toimiessa, tämä vaihe on kuitenkin niin myöhään prosessissa, että laajoja muutoksia ei ole enää mahdollista tehdä.

6.2.2. Näyttöjen välinen navigointi

REKI järjestelmän näyttökuvaukset olivat mukana jo tarjouskilpailumateriaalissa. Määrittelyn tarkennusvaiheessa toimittaja kävi läpi käyttötapaukset ja niihin liittyvät näytöt. Näistä tehtiin prototyyppi, jota esiteltiin projektiryhmälle, joka hyväksyi sen ilman käyttöliittymään merkittävästi vaikuttavia muutoksia. Tämä tarkoittaa että koko käyttöliittymä ja siihen kuuluva näyttöjen välinen navigointi oli pitkälti kiinnitetty jo projektin alussa. Lisäksi käyttötapauksiin painottuvan järjestelmäkehityksen johdosta

voidaan todeta että järjestelmän käyttöliittymä on syntynyt osittain käyttötapausten sivutuotteena.

Tarjouskilpailumateriaalissa oli mukana myös tilaajan käyttöliittymästandardi, joka oli kiinnitetty. Standardi määritteli käytettävät resoluutiot, fontit ja värit sekä erinäisten käyttöliittymäelementtien paikat. Lisäksi standardi määritteli että tietojen on mahdollista yhdelle näytölle ilman ruudun vierittämistä. Nämä asiat yhdessä käyttötapausten asettamien ehtojen kanssa johtivat tulokseen jossa useat tiedot on jouduttu jakamaan useammalle näytölle, vaikka ne ovat loogisesti kytköksissä toisiinsa, koska ruudun vieritys tai käyttöliittymäelementtien uudelleen sijoittelu on kielletty käyttöliittymästandardissa. Suurin ongelma tässä jäykkyydessä oli ehkä se että toimittajan käytettävyyssiantuntijan kommentteja ei otettu edes käsittelyyn, vaan todettiin että standardia seurataan.

REKIn järjestelmäkehityksestä vastaavassa projektiryhmässä ei ollut mukana tilaajan sidosryhmiä, jotka kuitenkin muodostavat suurimman osan järjestelmän käyttäjistä. Tilaajalla oli erilliset yhteistyöryhmät sidosryhmien kanssa mutta suoraa kontaktia käyttäjien ja toteuttajien välillä ei juuri koskaan ollut. Tämä on osaltaan saattanut vaikuttaa navigaation toteutustapaan.

Näyttöjen välisen navigoinnin ongelmat ovat siis perua tarjousmateriaalissa määritellyistä käyttötapauksista, joita ei ole merkittävästi muutettu käytettävyyssiantuntijan tai käyttäjiltä saadun palautteen perusteella. Järjestelmälle ei myöskään ole tehty merkittävässä määrin käytettävyystestausta, joka olisi nostanut esiin käytettävyysongelmat ennen tuotantoon siirtoa.

6.2.3. Tulostuksen ja raporttien ongelmien synty

REKI- järjestelmästä on mahdollista tulostaa tietoja useaa eri kautta; Oheistulostuksena suoraan käyttäjän omalle tulostimelle, keskitetysti Itellan palvelun avulla sekä käyttäjäorganisaation oman tulostuspalvelun kautta.

Tulostukseen liittyviä kysymyksiä ratkaisemaan muodostettiin tilaajan, toimittajan sekä tilaajan sopimuskumppaneiden yhteinen työryhmä. Työryhmässä saatiinkin suurin osa avoimista kysymyksistä ratkaistua niin itse tulostukseen liittyvän tekniikan kuin

tulostusta käyttävien käyttötapausten ja sitä kautta tulostukseen oikeuttavien käyttäjäroolien osalta.

Ongelmat erilaisten tulosteiden tilannekohtaisen valinnaisuuden ja pakollisuuden kanssa tulivat esiin testausvaiheessa kun tulostusta päästiin testaamaan käytännössä. Joihinkin ongelmiin saatiin korjaukset ennen tuotantoon siirtoa, jotkin ongelmat puolestaan havaittiin tai todettiin vakaviksi vasta tuotantoon siirron jälkeen.

Tulostukseen liittyvät ongelmat olisi ehkä voinut havaita aiemmin, mikäli prosessikehitystä olisi tehty samanaikaisesti järjestelmäkehityksen kanssa. Ongelmallista tässä tapauksessa olisi kuitenkin ollut se, että kyseessä olisi ollut niin tilaajan kuin tämän sidosryhmien prosesseista.

Raporttien ongelmat liittyvät pitkälti kehitysprojektin hallintoon. Raportteihin liittyviä päätöksiä saatiin aikaiseksi vasta projektin loppuvaiheessa, jolloin tietokannat ja niiden kentät oli jo määritelty ja toteutettu. Koska näihin ei enää ollut mahdollista saada aikaan muutoksia, eivät raporttien muodostavat tietokantakyselyt ole optimoituja, eikä kaikkia tarvittavia tietoja välttämättä saada yhdisteltyä järkevällä tavalla eri kannoista.

Raporttien toteutus projektin loppupäässä ei myöskään antanut mahdollisuutta koekäyttää ja viimeistellä raportteja riittävässä määrin ennen tuotantoon siirtoa.

6.2.4. Yhteenveto REKIn ongelmien synnystä

Myös REKI- järjestelmään liittyvät käytettävyysongelmat ovat suurelta osin syntyneet määrittelyvaiheessa. Toimittajalle luovutetussa kilpailutusmateriaalissa on ollut puutteita. Tilaajan organisaatiokulttuuri, teknisen arkkitehdin puute sekä ohjeistuksen riittämättömyys ovat mahdollistaneet projektikohtaisia tulkintoja, jotka ovat vaikeuttaneet ja monimutkaistaneet tietojen käsittelyä REKI- järjestelmässä. Tilaajan projektien hallintotapa on ollut ajoittain raskas ja päätöksiä on jouduttu odottamaan, tietyissä tapauksissa projektin loppumetreille asti, jolloin muutosten toteuttaminen on ollut vaikeaa tai mahdotonta. Tilaajan joustamattomuus palautteen ottamisessa ja kiinnitetyissä standardeissa pysymisessä johti näyttöjen säilymiseen lähes tarjousmateriaalissa määritellyissä muodoissa myös tuotantoon otetussa versiossa.

Käytönhallintajärjestelmien, sanomavälitysjärjestelmä SAIMIn sekä testiympäristöjen puutteellinen valmiusaste yllätti toimittajan ja pakotti näiden toimintojen kehittämiseen varsinaisten järjestelmien kehittämisen rinnalla, toisin kuin mihin toimittaja oli tarjousmateriaalin perusteella varautunut. Tämä hidasti töiden etenemistä ja vaikeutti erityisesti järjestelmien integraatiotestausta ja sitä kautta ongelmien havaitsemista ajoissa.

ATJn monimutkainen tietomalli, edellä mainitut projektikohtaiset tulkinnat sekä edeltävän järjestelmän vaikutus on hankaloittanut päivitysketjujen ja käsittelysääntöjen aukotonta toteuttamista. Ongelmien syntymistä on edistänyt käyttötapausten ja vaatimusten riittämätön tarkkuus. Lopputuloksena korjaukset ja erityisesti takautuvat muutokset ovat käyttäjille vaikeita.

Vesiputousmallin mukainen tulosten testaus vasta projektin loppuvaiheilla johti muutaman ongelman liian myöhäiseen havaitsemiseen jotta niitä olisi enää ehditty korjata. Raporttien toteutuksessa tarvittavien päätösten venyminen aina projektin viime metreille johti raporttien epäoptimaaliseen tietokantojen käyttöön. Lisäksi tietokantojen luonnissa ei tällöin ollut mahdollista huomioida raporttien tarpeita, jolloin tietyt raporttien tarvitsemat toiminnallisuudet ovat nyt turhan hitaita tai vaikeita, mikä heijastuu osaltaan myös raporttien sisältöihin.

6.3. PIIKOn käytettävyysongelmien synty

Tässä osassa esittelen PIIKO järjestelmän käytettävyysongelmien syntyä. Ensin esittelen järjestelmän kehitysprosessin kulun ja ongelmien synnyn yleisellä tasolla, minkä jälkeen tarkastelen tarkemmin yksittäisten ongelmien syntymiseen vaikuttaneita syitä.

PIIKO- järjestelmän perusversio toteutettiin yhtenä PALKO hankkeen järjestelmistä. Järjestelmän ohjelmistokehitysprosessi oli perinteisen vesiputousmallin mukainen. Käyttöönoton jälkeen PIIKO- järjestelmää on laajennettu kahteen otteeseen. Järjestelmään lisättiin raskaan kaluston ammattipätevyden tutkintojen ja ammattipätevyyskorttien hallinnointi vuonna 2008 PRODRI projektissa. Taksien ammattipätevyden tutkintojen ja ammattipätevyyskorttien hallinnointi lisättiin järjestelmään 2010. Toteutunut kehitysprosessi voidaan siis jakaa seuraaviin vaiheisiin.

- PIIKOn perusversio (2005)
- PRODRI ammattipätevyys projekti (2008)
- Taksikuljettajien ammattipätevyys toiminnallisuus (2010)

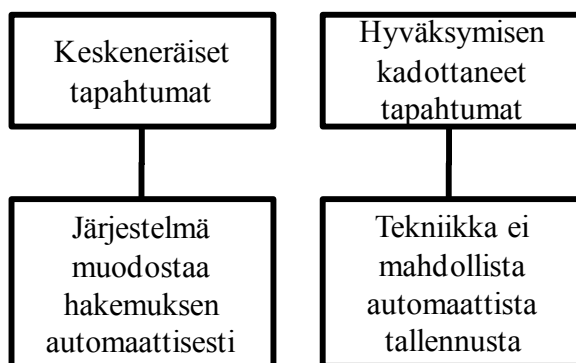
PIIKO järjestelmän keskeisimmiksi käytettävyysoongelmiksi nousi käyttäjien haastatteluissa yleisen tason ongelmat, listoille jäävät hakemukset, korvaavat kortit, yrityskortit, laskut sekä sekalaiset käytettävyysongelmat. Näiden lisäksi joitakin itse järjestelmän ulkopuolisia ongelmia, kuten ohjeistus ja tuki sekä prosessiin ja lomakkeisiin liittyvät asiat (ks luku ”5.3 PIIKOn keskeiset käytettävyysongelmat”).

6.3.1. Yleisen tason ongelmat

PIIKO järjestelmän luotettavuus, hitaus ja ongelmat yhteensopivuudessa, ovat suurilta osin koko ATJ- järjestelmän taustalla olevien teknisten ratkaisujen syytä. ATJn tekninen määrittely, joka määrää että järjestelmät tukevat vain määriteltyjä selaimia heijastuu kaikkiin osajärjestelmiin. Näitä käsitellään tarkemmin alla, kohdassa ”6.4 Yhteenveto järjestelmien ongelmien syistä”

6.3.2. Listoille jäävät hakemukset

PIIKO- järjestelmän listoille jää sellaisia hakemuksia, joita käyttäjät eivät toivo niillä olevan. Näihin listoille jäävien hakemusten ongelmaan havahduttiin vasta tuotannossa. Haastateltujen käyttäjien mielestä ne sotkevat työprosessia ja hankaloittavat siten asiakaspalvelutilannetta.



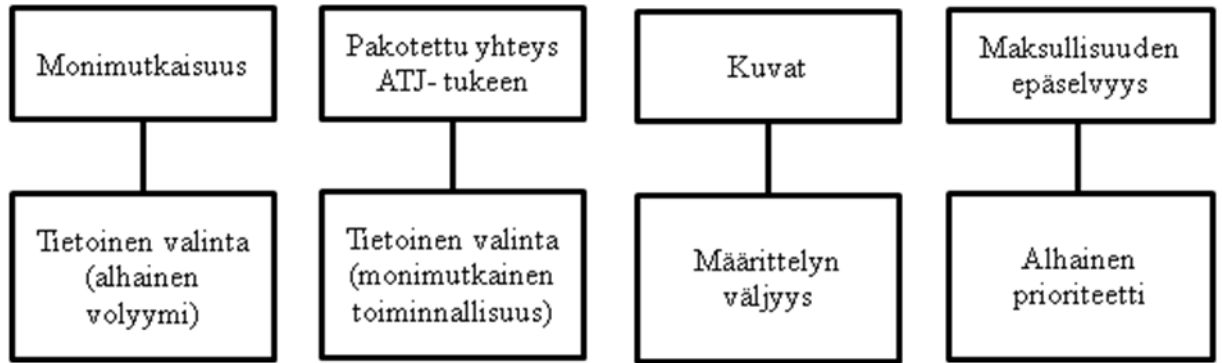
Kuva 17 PIIKOn listoille jäävien syyt

Määriteltäessä PIIKO- järjestelmän kautta hallittavan ”kuljettajakortin” toimintaa haluttiin että kortti luodaan samalla kun itse hakemuskin. Tämän huomattiin tuotannossa johtavan siihen, että kun käyttäjä jostain syystä ei vie tapausta loppuun, on hakemus ja siihen liittyvä kuljettajakortti jo luotu ja tallennettu kertaalleen järjestelmän toimesta. Koska kyseessä on silloin tilanne, jossa hakemusta ja korttia ei ole mahdollista erottaa tahallaan kesken jätetystä tapauksesta, jää tämä tahaton hakemus käsiteltävien listalle. Ongelman havaitsemisen jälkeen tutkittiin mitä asialle voisi tehdä, mutta hakemuksen nykymuotoinen käsittely on kuljettajakortin osalta niin oleellinen osa PIIKO- järjestelmää, että sen muuttaminen vaatisi kohtuuttoman paljon työtä nähtyyn hyötyyn verrattuna.

Kaikkia PIIKOn kautta hallittavia korttityyppejä koskettava ongelma, jossa hakemus kadottaa hyväksymismerkintänsä, johtaa juurensa valittuun ATJ- tasoiseen arkkitehtuuriratkaisuun. Koska PIIKO, muiden ATJ- järjestelmien tavoin, on verkkosovellus jota käytetään yhteensopivan selaimen avulla, rajoittaa sen toimintaa sovelluksessa käytävissä oleva tekniikka. Tässä tapauksessa PIIKOn näytöt ovat periaatteessa html- ja javascript- pohjaisia sivuja, jolloin yhdellä sivulla tehdyt muutokset eivät automaattisesti siirry toiselle sivulle ilman tietojen eksplisiittistä tallentamista käyttäjän toimesta. Tämä tarkoittaa että, kun käyttäjä on merkinnyt hakemuksen hyväksytyksi, ei tietoa ole varsinaisesti vielä tallennettu järjestelmään. Tallennus tapahtuu vasta kun käyttäjä painaa ”tallenna” painiketta. Mikäli käyttäjä siirtyy toiselle sivulle, esimerkiksi korjaamaan osoitetietoja, katoaa hyväksymismerkintä yhdessä muiden edellisen tallennuksen jälkeisten tietojen kanssa. Tämä on valitun toteutustekniikan rajoitus, jonka kiertäminen ei ole mahdollista kyseisen tekniikan avulla.

6.3.3. Korvaavat kortit

Piirturikorttien korvaamisen ongelmien taustalla on useampi syy. Päällimmäisenä se, että kortin korvaaminen on prosessimielessä selkeästi monimutkaisempi kuin kortin hakeminen. Kadonnut tai vioittunut kortti on ensin kuolettettava ja siitä on tehtävä katoamisilmoitus. Vasta tämän jälkeen voidaan uutta korttia hakea.



Kuva 18 PIIKOn korvaavien korttien syyt

Kortin korvaaminen on jo suunnittelussa tiedostettu normaalitapausta monimutkaisemmaksi. Tuotannossa ongelman on kuitenkin todettu esiintyvän suhteellisen alhaisella volyymillä, joten asiaa ei ole päätetty korjata.

Järjestelmän ohjeistuksessa yllä mainittu monimutkaisuus on pyritty kompensoimaan siten, että käyttäjiä kehoitetaan soittamaan joka kerta ATJn puhelintukeen kun tällainen tapaus tulee vastaan. Tämä sysää ongelman tukipalvelulle, jonka työntekijät eivät välttämättä ole käyttäjää paremmin selvillä korttien korvaamisesta.

Korvaavaa korttia haettaessa ei asiakkaan ole välttämätöntä toimittaa uusia valokuvia ja nimikirjoitusnäytettä digitoitavaksi, vaan kortille on mahdollista hakea jo tietokannassa oleva kuva ja nimikirjoitusnäyte. Näiden kahden mahdollisuuden välillä tehtävä valinta on, järjestelmän määrittelyjen epäselvyyden vuoksi, merkitty epäselvästi käyttöliittymällä. Korjausta on mietitty, mutta jätetty toistaiseksi tekemättä.

Käyttäjien esittämä huoli korttien maksullisuuden epäselvyydestä on ilmaantunut vasta viime aikoina, eikä tätä ole priorisoitu kovin korkealle korjaustoimenpiteistä päätettäessä.

6.3.4. Yrityskortit

Yrityskorttien ongelmina oli niiden kuljettajakorteista poikkeava toiminta niin korttien oletusmäärän, määrän hallinnan sekä yrityskortin hakemusnäytön navigoinnin osalta.



Kuva 19 PIIKOn yrityskortti ongelmien syyt

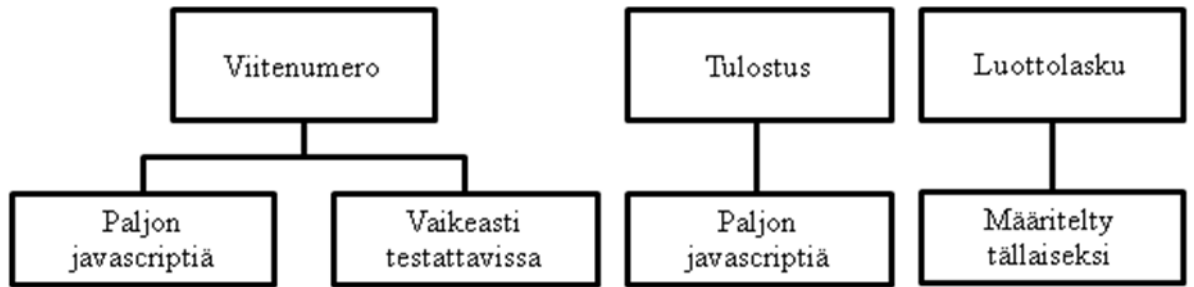
Yrityskorttien oletusmäärä poikkeaa kuljettajakortista koska määrittelyissä on todettu että yritykselle voidaan hakea sekä yritys-, että korjaamokortteja. Tämän vuoksi järjestelmän ei ole mahdollista automaattisesti päätellä millainen kortti hakemukselle tulisi lisätä hakemusta luotaessa. Mahdolliset korjaukset eivät ole mahdollisia ilman hakemusprosessin muuttamista nykyisestä.

Korttien määrän hallinnassa ongelmaa aiheuttaa ”lisää” ja ”poista” toiminnallisuuden sijoittaminen eri kohtiin näyttöä. Korttien lisäys onnistuu selkeästi merkitystä ”Lisää” painikkeesta. Korttien poistaminen puolestaan on tehtävä hakemukseen liittyvien korttien listauksesta, jossa haluttu kortti on poistettava listan oikeassa reunassa olevalla ”ruksilla”. Tämä kaksijakoisuus johtuu käytettävästä listatoiminnallisuudesta, joka on toteutettu käyttämällä uudelleen muualla järjestelmässä olevaa listatoiminnallisuutta. Lisäksi yrityksen kortit voivat olla jo määritelty tietyille henkilölle, joten käyttäjän on tällöin joka tapauksessa valittava mikä kortti poistetaan.

PIIKO- järjestelmä ei ohjaa käyttäjiä pois yrityskortin hakemusnäytöltä, vaan käyttäjien on itse tiedettävä miten, ja missä vaiheessa, näytöltä tulee poistua. Tämä ongelma todettiin jo testausvaiheessa mutta kovan kiireen, resurssipulan ja varsinaisen yrityskortin hakutoiminnallisuuden ollessa kunnossa, ei korjaustoimenpiteisiin ryhdytty.

6.3.5. Laskut

Laskut ovat olleet ongelmallisia PIIKO- järjestelmän alkuajoista asti. Ongelmiin on tullut korjauksia ajan myötä, mutta laskunäyttö on edelleen virheherkkä. Ongelmia on ollut erityisesti laskulle tulevan pankkiviitteen muodostamisessa, laskun tulostamisessa sekä koko luottolasku-toiminnallisuuden kanssa.



Kuva 20 PIIKOn laskujen ongelmien syyt

Järjestelmään toteutettu viitenumeron generointi sisältää paljon javascriptillä toteutettua toiminnallisuutta, mikä lisää virhemahdollisuuksia. Tämä tiedettiin ongelmaksi jo testauksen aikana. Toiminnallinen testaus aina pankeista järjestelmään asti oli hankala toteuttaa kokonaisuutena, eikä aika ja resurssit riittäneet täydellisen ratkaisun etsimiseen. Viitteen muodostukseen on tehty korjauksia ajan myötä, mutta näyttö on edelleen virhealtis ja järjestelmä kaatuukin ajoittain käyttäjien toimiessa tällä näytöllä.

Laskun tulostus on mahdollista vasta kaikkien tietojen jälkeen, jonka jälkeen lasku muodostetaan javascriptin avulla Adoben PDF-muotoon. Tämä on monimutkainen toiminnallisuus, mikä tekee siitä virhealttiin. Ongelmaa pahentaa ATJn tukemien selainten hidas ja vanhentunut javascript tuki verrattuna uusimpiin selaimiin.

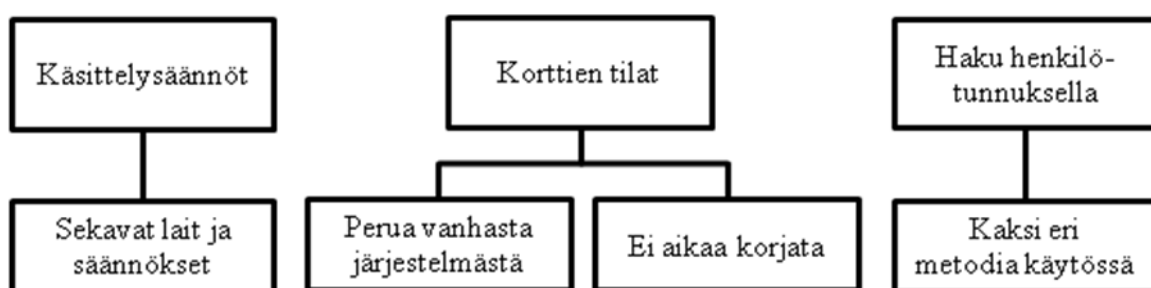
6.3.6. PIIKOn muut käytettävyysongelmat

Käytettävyysongelmiksi luokiteltiin myös järjestelmän sisäisen yhteneväisyyden puutteet, korttien tilojen epäselvä nimeäminen, ammattipätevyyksien voimassaolojen virheelliset käsittelysäännöt sekä henkilötietojen haku henkilötunnuksella.



Kuva 21 PIIKOn muiden ongelmien syyt – osa 1

Järjestelmän sisäisen yhteneväisyyden puute nousi usein esiin haastatteluissa. Tämän taustalla on muutama suuri syy. Ensinnäkin järjestelmälle ei ole tehty käyttäjätestausta missään vaiheessa, vaan testaus on pääasiassa tehty toteuttajien tai järjestelmästä vastaavien toimesta. Lisäksi PIIKO- järjestelmä nykymuodossaan on kolmen eri projektin tulos. Tämä tarkoittaa että järjestelmää on kehitetty kolmeen eri otteeseen aina hieman eri määrittelyillä, jolloin eri toiminnallisuuden määrittelyt ovat poikenneet toisistaan. Lisäksi määrittelyt ovat olleet epäselviä, jolloin niitä on voitu ymmärtää useilla eri tavoin. Projektin tiukka aikataulu ja resurssien puute karsivat pois kaikki paitsi järjestelmän perustoiminnallisuuden kannalta tärkeät korjaukset.



Kuva 22 PIIKOn muiden ongelmien syyt - osa 2

Piirturikorttien tilat on pitkälti kopioitu vanhasta ajokorttijärjestelmästä. Tilojen nimet eivät oikein vastaa järjestelmän tarpeita eivätkä kuvaa tarkasti korttien todellista tilaa, mutta koska ne ovat oleellinen osa järjestelmän toimintaa, on näiden korjaamisessa oltu varovaisia. Korjauksia on tehty lähinnä muiden projektien yhteydessä, jolloin on ollut mahdollista tehdä suurempia muutoksia. Muuten aikataulu ja rahoitus ei ole antanut myöten tilanteen korjaamiseksi.

Suurempi ongelma on ammattipätevyyskorttien voimassaoloa laskevien käsittelysääntöjen virhealttius. Kyseessä ei ole oikeastaan järjestelmätason ongelma, vaan enemmän lain tulkintaongelma. Voimassaoloa koskevat lait ja säännökset ovat erittäin monimutkaisia ja järjestelmää määriteltäessä saatiin eri tahoilta kysyttäessä eri tulkintoja. Näiden muuntaminen järjestelmässä toteutettaviksi käsittelysääntöiksi johti erittäin monimutkasiin sääntöihin, joiden tiedettiin jo toteutusvaiheessa toimivan virheellisesti tietyissä tilanteissa. Ongelmaan ei kuitenkaan ollut olemassa järkevästi toteutettavaa ratkaisua, ja tilaajaorganisaation näkemys oli että järjestelmän on tehtävä mahdollisimman suuri osa päätöksistä automaattisesti. Viime aikoina on pyritty

suuntaamaan järjestelmän kehitystä enemmän käyttäjän tekemien tulkintojen suuntaan, niin että järjestelmä vain tuo esiin päätöksessä tarvittavan tiedon, eikä yritä automaattisesti tehdä tulkintaa.

Pienempi ongelma on asiakkaan henkilötietojen haku henkilötunnuksen perusteella. Siinä käyttäjän on muistettava käyttää henkilötunnuksen loppuosassa suurta kirjainta. Tämä johtuu siitä että henkilötietojen hakuun käytetään kahta eri metodia, joista toinen on aakkoskoosta riippuva (eng. case-sensitive). Ongelmaa ei havaittu ennen tuotantoa osittain yllä mainitun käyttäjätestauksen puuttumisen vuoksi. Koska ongelma on helposti kierrettävissä, ei sen korjaamiseen ole panostettu.

6.3.7. Yhteenveto PIIKOn käytettävyysohjelmien syistä

PIIKO- järjestelmän käytettävyysohjelmien taustalta löytyy varsin samankaltaisia ongelmia kuin VERO- ja REKI- järjestelmien osalta. Lisäksi PIIKolla on muutama järjestelmälle ominainen ongelmalähde.

Määrittelyt esiintyvät käytettävyysohjelmien syynä kaikissa yllä mainituissa kategorioissa. Tämä ei ole yllättävää kun huomioidaan että määrittelyt oli luotava ennen kuin taustalla oleva lainsäädäntö ja työprosessit oli kunnolla kiinnitetty. Erityisen hankala tilanne on ammattipätevyyden voimassaolon käsittelysääntöjen osalta, koska niiden kohdalla lopullinenkin lainsäädäntö on epäselvä ja avoin tulkinnoille. Muissa tapauksissa määrittelyt on jouduttu kirjoittamaan varsin avoimiksi, jolloin niiden perusteella tehdyt käyttötapaukset ovat olleet avoimina toteuttajien tulkinnalle. Liian avoimet käyttötapaukset ovat myös johtaneet puutteisiin järjestelmän sisäisessä yhteneväisyydessä kun toteuttajat ovat tulkinneet niitä vaihtelevasti. Määrittelyiden tiettyjen osien lainaaminen vanhoilta, eri prosessia toteuttavilta järjestelmiltä on johtanut kömpelöihin nimeämiskäytäntöihin.

PIIKO- järjestelmän toiminnan laajentaminen on myös omalta osaltaan syynä yllä mainittuihin ongelmiin sisäisessä yhteneväisyydessä. Perus PIIKOn jälkeen tulleiden projektien määrittelyt on tehty toiminnallisuuden ehdoilla, eikä projekteilla ole ollut aikaa eikä resursseja synkronoida tehtyjä ratkaisuja koko järjestelmään. Tämän vuoksi muuten samankaltaiset toiminnot näkyvät käyttäjälle hieman eri lailla toimivina.

Ajan ja resurssien puutteen vuoksi järjestelmän kehityksessä on keskitytty lähinnä toiminnallisuuden toteuttamiseen, eikä käyttäjätuesta ole tehty lainkaan. Pelkästään toiminnallisuuksiin keskittyminen on johtanut tilanteeseen jossa järjestelmän tehokas käyttö vaatii järjestelmän ominaisuuksien tuntemista siten, että käyttäjä tuntee ongelmakohdat ja miten ne kierretään. Uusille käyttäjille järjestelmä ei esiinny selkeästi toimivana ja helposti käytettävänä kokonaisuutena. Lisäksi joidenkin ongelmien kohdalla on päätetty että korjausta ei tehdä, vaan ohjeistetaan käyttäjät ottamaan yhteyttä puhelintukeen ongelman hallittua kiertämistä varten.

Järjestelmän käyttämä tekniikka asettaa joitakin rajoitteita sen toiminnalle. Verkkosovelluksena toimiminen rajoittaa syötekenttien käsittelyä ja vaatii tarkempaa käyttöliittymäsunnittelua kuin tavallinen sovellus, jotta käyttökokemus olisi sujuvaa. Nyt kenttien sisältö voi kadota, mikäli siirrytään toiselle sivulle ennen tietojen eksplisiittistä tallennusta. Myös tiettyjen näyttöjen sisältämä suuri määrä javascriptiä on raskasta tietyille selaimille.

Rajoitettu selainten tuki aiheuttaa myös erilaisia ongelmia järjestelmän käytössä. Alkuaikoina tuettu selain oli niin uusi että käyttäjäorganisaatioissa oltiin vasta siirtymässä siihen. Tällä hetkellä tuetut selaimet ovat niin vanhoja että käyttäjäorganisaatio ei mielellään niitä enää käytä ja niiden virallinen tuki on jo loppunut. Lisäksi niiden tietoturvasuus ja javascriptin käsittelyn tehokkuus on varsin kaukana uudemmissa selaimista.

6.4. Yhteenveto järjestelmien ongelmien syistä

VERO-, REKI- ja PIIKO- järjestelmien yhteisiksi käytettävyysongelmiksi voidaan laskea käyttöliittymän ongelmat sekä tietoihin liittyvät ongelmat. Käyttöliittymiin liittyvät ongelmat voidaan jäljittää tilaajan käyttöliittymästandardin asettamiin rajoituksiin sekä sen riittämättömyyteen näin laajassa projektissa. Käyttöliittymien ongelmien muodostumista edisti myös tilaajan teknisen asiantuntemuksen puute, eritoten PALKO hankkeen alkuvaiheessa. Tekninen arkkitehti, jolla olisi ollut laajempi ja kokonaisvaltaisempi kuva projekteista ja niissä käytettävistä ratkaisuista, olisi saattanut auttaa ongelmien havaitsemisessa ja korjaamisessa. Lisäksi määrittelyt ovat olleet ajoittain liian väljät ja auki toteuttajan tulkinnalle. Myös ajan ja resurssien puute

on pakottanut hyväksymään järjestelmien tuotantoon siirron heti kun halutut toiminnallisuudet toimivat siedettävästi.

Käyttöliittymän ongelmiin luen tässä tutkimuksessa myös ongelmat järjestelmien nopeudessa, luotettavuudessa sekä yhteensopivuudessa. Järjestelmät ovat käyttäjien keskuudessa tunnettuja siitä, että ne toimivat ajoittain hyvin hitaasti ja myös kaatuvat joskus varoittamatta. Lisäksi järjestelmien käyttöä rajoittaa yhteensopivien selainten rajoitettu määrä.

Tietoihin liittyvät ongelmat johtavat VEROn ja REKIn tapauksessa juurensa erittäin monimutkaiseen tietomalliin, tietojen kytköksiin järjestelmien välillä, ohjeiden projektikohtaisiin tulkintoihin sekä kokonaistilanteen hahmottamisen puutteisiin. VERO-, REKI- ja PIIKO- järjestelmien tietomallit on muodostettu tilaajan prosessien pohjalta, mutta niitä olisi selkeästi kannattanut käydä läpi tarkemmin kokonaisuutena, huomioiden erityisesti eri järjestelmien väliset kytkökset sekä järjestelmän käyttäjän näkökulma. Tilaajan organisaatiokulttuurista johtuen järjestelmäkehitys jakaantui useaksi projektiksi, joiden välinen kommunikaatio ei ollut paras mahdollinen. Myös tässä kokonaiskuvan puute ja erillisen teknisen arkkitehdin puute näkyy selkeästi. Projektien aikana eri järjestelmien kehityksissä tehtiinkin projektikohtaisia tulkintoja eri tietojen kentänimistä ja –pituuksista, mikä hankaloittaa tietojen yhteiskäyttöä ja altistaa järjestelmät herkemmin erinäisiin tietojen synkronointiin liittyviin ongelmiin.

Vesiputousmallin seuraaminen on yksi osasy siihen miksi käyttöliittymiin ja tapahtumiin liittyvät ongelmat pääsivät tuotantoon asti. Vesiputousmallissa testaus tehdään projektin loppuvaiheilla, jolloin vasta tähän vaiheeseen muodostetaan riittävän laaja ja laadukas testausaineisto jonka avulla voidaan luotettavasti simuloida tuotannon kaltaisia tilanteita. Tällöin epätavallisemmat ongelmat jäävät kokonaan huomaamatta aikaisemmissa vaiheissa ja tavallisempien ongelmien vakavuus jää myös usein huomaamatta ennen tätä. Koska projektien aikataulut on yleensä kiinnitetty, ei testausvaiheessa havaittuihin laajamittaisiin ongelmiin enää ehditä kehittää korjauksia, jolloin ne pääsevät siirtymään tuotantoon.

Järjestelmäkehityksen toteuttaminen toimintopainotteisesti on perinteinen tietojärjestelmäkehityksen toimintamalli, mutta siinä sivuutetaan pitkälti käyttäjien tarpeet. Missään vaiheessa tutkittujen järjestelmien kehitystä ei tehty käyttäjätestausta tai

muuta käytettävyyteen kohdistuvaa kehitystyötä. Myös järjestelmien jatkekehityksessä ja ylläpidossa on keskitytty pelkästään toiminnallisten ongelmien ratkaisuun tai uusien toiminnallisuuden lisäämiseen. Prosessi- ja käytettävyyden näkökulma on siten jäänyt lähes kokonaan pois kehityksestä.

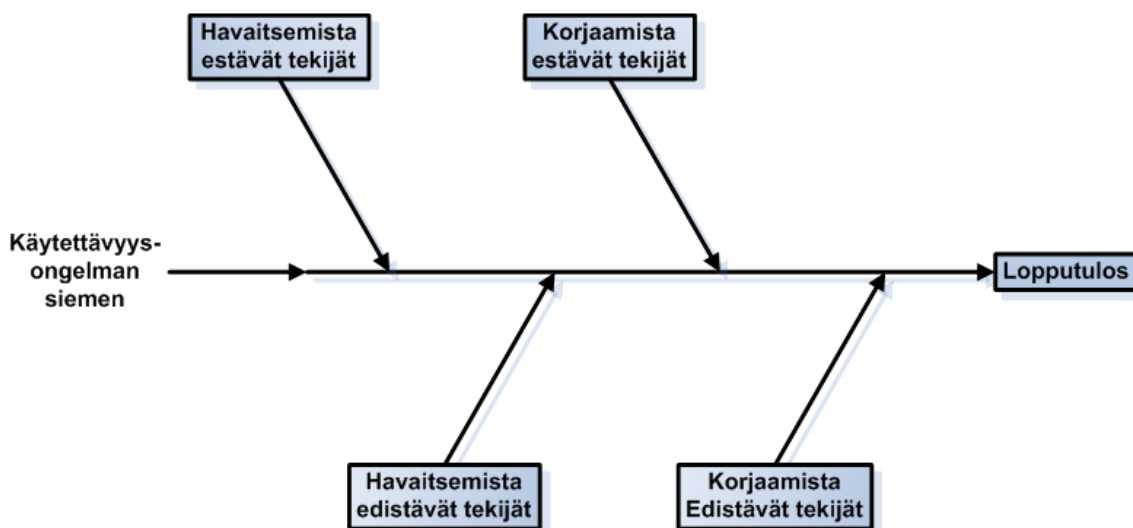
7. Johtopäätökset

Aiemmissa luvuissa esittelin löydetyt ongelmat (ks. ”5 Järjestelmien keskeiset käytettävyysongelmat”) ja niiden syntyyn johtaneet syyt (ks. ”6 Käytettävyysongelmien syyt”). Tässä luvussa esittelen kehittämäni mallin ongelmien syntymisen syistä.

7.1. Ongelmien syntymekanismi

Suurella osalla havaituista käytettävyysongelmista oli juurensa järjestelmän määrittelyissä. Tämä näkökulma on yhtenevä Jani Hanhisalon näkemykseen että ”suuret käytettävyysongelmat ... ovat ongelmia, joiden syyt ovat vaatimusmäärittelyssä” (Hanhisalo 2008). Jälkikäteen tarkasteltuna järjestelmien ongelmat ovat selkeästi havaittavissa, joten järjestelmäkehityksen aikana jotkin tekijät ovat ilmeisesti estäneet ongelmien havaitsemista tai korjaamista ennen tuotantoon siirtoa.

Tällöin karkea malli ongelmien synnystä, ja niiden siirtymisestä tuotantoon voisi näyttää ”Kuva 23” mukaiselta. Tässä mallissa huomioidaan järjestelmän kehityksessä tapahtuva eri prosessien ja toimijoiden välillä tapahtuva vuorovaikutus, joka määrää miten hyvin kehityksessä ongelmat havaitaan ja kyetään korjaamaan havaitut ongelmat. Pura auki mallin osia seuraavissa osissa.



Kuva 23 Malli käytettävyysongelmien syntyprosessista

Tämän tutkimuksen perusteella käytettävyysongelmien havaitsemiseen vaikuttavat asiat voidaan jakaa eri osa-alueisiin. Karkealla tasolla voidaan asiat jakaa ongelmien havaitsemista ja korjaamista estäviin tai edistäviin syihin. Tässä osiossa kuvaan tutkittaviin järjestelmiin vaikuttaneita, käytettävyysongelmien havaitsemiseen vaikuttavia tekijöitä.

7.2. Ongelmien havaitsemista ja korjaamista estävät tekijät

Ongelmien havaitsemista ja korjaamista estää tai hankaloittaa monta erillistä tekijää. Olen tässä työssä ryhmitellyt ne löyhästi yleisiin tekijöihin, ympäristöstä johtuviin, organisaatio- ja työkuulttuurista johtuviin, arkkitehtuurista, testauksesta tai ohjeistuksesta johtuviin tekijöihin.

Yleiset tekijät

Yleisellä tasolla tilaajaorganisaation *epäselvä tahtotila* haittaa ongelmien havaitsemista, koska projektiin osallistuville ei ole selvää mihin tilaaja tarkalleen ottaen pyrkii. Hyvänä esimerkkinä tästä toimii PIIKO- järjestelmä, jonka kehityksen alkuvaiheissa ei oikein tiedetty mitä kaikkea järjestelmän tulisi tehdä. Kehitysprojektin aikainen *kiire* sekä *resurssien puute* aiheuttavat toteuttajien keskittymistä vain omaan tehtäväänsä, sekä näiden tehtävien mahdollisimman nopeaan valmistumiseen. Tällöin mahdolliset ongelmat saattavat jäädä huomiotta tai niiden korjaaminen ohitetaan muiden kiireiden vuoksi. *Yleiskuvan puuttuminen* hankaloittaa järjestelmien osien väliseen toimintaan liittyvien ongelmien havaitsemista. Lisäksi yleiskuvan puuttuminen voi johtaa saman toiminnallisuuden toteuttamiseen useaan kertaan. Kaikissa tutkituissa järjestelmissä kiire ja resurssien puute aiheuttivat ongelmien siirtymisen tuotantoon, koska niiden korjaamiseen ei ollut aikaa eikä resursseja. Yleiskuvan puutteen vaikutus on parhaiten nähtävissä VERO- järjestelmän maksuunpano- toiminnallisuuden toteuttamisessa usean eri toteuttajan toimesta.

Ympäristö

Tilaajaorganisaation toimintaympäristö vaikuttaa myös järjestelmäkehitykseen. PIIKO- järjestelmä on hyvä esimerkki siitä miten *monimutkainen lainsäädäntö* heijastuu järjestelmään hankalina ja virheherkkinä käsittelysääntöinä. Koska

järjestelmäkehityksen pohjana toimiva lainsäädäntö on monimutkainen, ei sen perusteella toteutettavasta järjestelmästä tule helposti selkeää, jolloin mahdolliset ongelmat jäävät usein löytämättä. Myös organisaation *monimutkaiset liiketoimintaprosessit* ovat omiaan aiheuttamaan ongelmia järjestelmässä. Koska toteuttajat eivät ymmärrä järjestelmän taustalla olevaa liiketoimintaprosessia, eivät he aina saa käyttöönsä käyttötapauksiin liittyvää hiljaista tietoa. Tämä näkyy hyvin esimerkiksi VERO- järjestelmän toiminnassa, jossa käsittelylogiikka on monimutkaista eikä kaikkia tapauksia ole mahdollista käsitellä automaattisesti. Lisäksi järjestelmäkehitykseen vaikuttaa siitä erillään olevat *muutokset lainsäädäntöön tai liiketoimintaprosessiin*. PIIKO- järjestelmän kohdalla tällaiset muutokset lainsäädäntöön projektin aikana, aiheuttivat muutoksia jo valmiiksi monimutkaisiin käsittelysääntöihin, jolloin toteuttajilla ei enää ollut yhtä selkeää kuvaa käsittelysääntöjen toiminnasta. *Toimintaympäristöstä tulevat pakolliset muutostarpeet* aiheuttavat muutoshallintaa järjestelmiin, jolloin pakolliset muutokset priorisoidaan muiden korjausten edelle, mikä johtaa käytettävyysongelmien päätymiseen tuotantoon.

Organisaatio- ja työkuultuuri

Kaikkien järjestelmäkehitykseen osallistuvien organisaatioiden organisaatio- ja työkuultuuri vaikuttaa kehitykseen osallistuvien henkilöiden toimintaan. Nyt tutkittujen järjestelmien osalta on havaittavissa että kehitykseen osallistuneiden kolmen organisaation *keskittyminen toiminnallisuuksiin* on johtanut selkeiden käytettävyysongelmien päästämiseen tuotantoon. Joissakin tapauksessa käytettävyyso ongelmia ei tiedosteta tai niitä ei pidetä riittävän tärkeinä, vaan todetaan että järjestelmän toteuttama toiminnallisuus on toivotun mukainen. Hyvänä esimerkkinä tästä on PIIKO- järjestelmän loogisesti samankaltaisten toiminnallisuuksien eroavat käytötavat. Organisaatioiden ja näiden osien välinen *puutteellinen kommunikaatio* haittaa yhteisten rajapintojen toimivaa toteuttamista. REKI- ja VERO- järjestelmien välinen tietojen synkronointi-ongelma toimii varoittavana esimerkkinä siitä miten huono kommunikaatio estää ongelman havaitsemisen ennen kuin järjestelmien kehitys on jo pitkällä. Heikko kommunikaatio myös heikentää sellaista projektien välistä tiedon vaihtoa, jonka perusteella voitaisiin havaita yhteisiä ongelmia joihin kannatta kehittää keskitetty ratkaisu.

Kehitysprojektien *raskas hallinnointi* hankaloittaa tarvittavien linjauksien valitsemista ja toteuttamista. Vaikka kyse ei olisi ketterästä ohjelmistoprojektista, tulisi projektiin uhrattava hallinnollinen työmäärä pyrkiä pitämään projektin laajuuteen sopivana. Liikenteen turvallisuusvirastolla, julkishallinnon organisaationa, on pitkä historia byrokraattisesta hallinnoinnista. Tämä on tullut osaksi viraston organisaatio- ja työkuultuuria, mikä näkyy projektista päättävien erilaisten työ- ja johtoryhmien suurena määränä, jolloin etsityn linjauksen merkitys voi hukkaa byrokraatiaan. Organisaatio- ja työkuulttuurista riippuu myös osittain käytetty dokumentointimenetelmä sekä käyttötapausten kirjaamismenetelmät. Mikäli *käyttötapaukset ovat riittämättömiä tai monitulkintaisia*, johtaa se usein tilanteeseen jossa ongelmia ei välttämättä ole mahdollista havaita ennen järjestelmien laajamittaista testausta.

Organisaatiokulttuurilla on todennäköisesti ollut osansa myös *vesiputousmallin* valinnassa järjestelmäkehityksen prosessimalliksi. Vesiputousmallin yhtenä heikkoutena on testauksen sijoittuminen lähelle käyttöönottoa. Tämä tarkoittaa että testauksessa havaittuihin ongelmiin on yleensä erittäin vähän aikaa etsiä ratkaisuja. Yhdistettynä hitaaseen ja jäykkään hallinnointiin tämä lähes takaa että vain pienet tai absoluuttisen kriittiset ongelmat korjataan ennen järjestelmän tuotantoon siirtoa. Myös tapa *priorisoida korjauksia* vaikuttaa tuotantoon pääsevien ongelmien tyyppiin. Mikäli keskitytään toiminnallisten ongelmien priorisointiin, eivät käytettävyyteen liittyvät ongelmat pääse korjauslistalle.

Arkkitehtuuri

Järjestelmälle valittu arkkitehtuuri vaikuttaa myös kehitettävän järjestelmän käytettävyyteen. VERO-, REKI- ja PIIKO- järjestelmien käytettävyyteen vaikuttaa arkkitehtuurin *monimutkainen tieto- ja toimintamalli*. Monimutkaisuus aiheuttaa myös sen, että havaittuihin ongelmiin ei uskalleta puuttua koska ei olla varmoja miten korjaukset vaikuttavat kokonaisuuden toimintaan. VERO- ja REKI- järjestelmien osalta ongelma konkretisoituu tietojen hankalana synkronointina ja tietojen ajoittain riskialttiina ylläpitona. PIIKO- järjestelmässä ongelmana on virheellisiä tuloksia tuottava käsittelysäännöt. Tietomallin monimutkaisuus, yhdessä huonon kommunikaation kanssa, viivytti näissä järjestelmissä ongelman havaitsemista.

Järjestelmien välisten *liittymien suuri määrä ja monimutkaisuus* nostaa käytettävyyso Ongelmien riskiä. VERO- järjestelmän monimutkainen liittymä REKI- järjestelmään onkin eräs syy käyttäjien havaitsemiin, korjausta vaativiin, virheisiin ajoneuvojen verotiedoissa. Arkkitehtuurista johtuviin tekijöihin luen myös valitun *tekniikan rajoitukset*. Näihin kuuluu esimerkiksi verkkosovelluksien ongelma kenttien sisällön käsittelyssä, kuten PIIKO- järjestelmän tallennusten yhteydessä. Tekniikan asettamat rajoitukset ovat ongelmallisia koska niitä on yleensä lähes mahdotonta kiertää ilman valitun tekniikan vaihtamista, mihin ei käytännössä juurikaan ryhdytä.

Testaus

Järjestelmien testaus on vesiputousmallissa lähes viimeinen tilaisuus löytää ja korjata ongelmia, ennen niiden siirtymistä järjestelmän mukana tuotantoon. Tässä onkin eräs testauksen ongelmista. *Liian myöhäinen* testaus, kuten esimerkiksi vesiputousmallin mukaan toimittaessa, ei enää mahdollista suurempien ongelmien korjaamista. Mikäli käytetään *liian suppeita testitapauksia* tai *liian suppeaa testimateriaalia* on vaarana että epätavallisempia ongelmia ei löydetä testausjakson aikana. Myös *kiirehditty testaus* aiheuttaa samankaltaisen ongelman. Esimerkiksi PIIKO- järjestelmässä esiintyvä hakemusten jääminen listoille olisi voitu havaita riittävän kattavalla testitapauksilla ja testimateriaalilla.

Mikäli jonkin *toiminnallisuuden testaaminen on hankalaa* tai vaatii erikoisia järjestelyjä, saatetaan sen testausta välttää. Tällöin ongelmia ei välttämättä havaita ennen kuin tuotannossa. PIIKO- järjestelmän rahaliikenteen testauksen kanssa oli tämän tyyllisiä ongelmia, minkä vuoksi tuotantoon pääsi joitakin laskutukseen liittyviä ongelmia. Myös *tarvittavan testimateriaalin puuttuminen* tekee testaamisesta mahdotonta, jolloin korjausten varmistaminen on hankalaa.

Ohjeistus

Ohjeistus on osa jokaista kehitysprojektia. *Epäselvä, riittämätön tai ristiriitainen ohjeistus*, aiheuttaa epävarmuutta ja heikentää siten ongelmien havaitsemista. Erityisen selkeää tämä oli VERO- ja REKI- järjestelmien välisissä liittymissä, jotka on kehitetty samojen ohjeistuksien avulla, mutta joiden yhteistoiminta on ollut ajoittain haasteellista. *Ohjeistuksen joustamattomuus* rajoittaa mahdollisten ratkaisujen käyttöä. Mikäli jonkin

ongelman ratkaisu on annetun ohjeistuksen vastainen, tulisi mahdollista ratkaisua ainakin harkita, eikä tyrmätä sitä suoralta kädeltä pelkästään ohjeistukseen viittaamalla.

7.3. Ongelmien havaitsemista ja korjaamista edistävät tekijät

Järjestelmien käytettävyysongelmien havaitsemista ja korjaamista edistää tai helpottaa myös monta erillistä tekijää. Olen tässä työssä ryhmitellyt ne vastaaviin luokkiin yllä kuvailtujen ongelmien havaitsemista ja korjaamista estävien tekijöiden kanssa. Luokat ovat siis yleiset tekijät, ympäristöstä johtuvat, organisaatio- ja työkuulttuurista johtuvat, arkkitehtuurista, testauksesta tai ohjeistuksesta johtuvat tekijät.

Yleiset tekijät

Mikäli järjestelmiä kehittäville organisaatioilla on *riittävän hyvä yleiskuva* järjestelmien kehitystilasta, käytetystä arkkitehtuurista sekä järjestelmien tahtotilasta, auttaa se havaitsemaan käytettävyyso- ja muita ongelmia kehityksen aikana. Erityisesti mahdolliset päällekkäisyydet ja jaetut ongelmat nousevat helpommin esiin, ja niihin on mahdollista kehittää toimivia ratkaisuja. VERO-, REKI- ja PIIKO- järjestelmiä kehitettäessä kenelläkään ei ollut hyvää yleiskuvaa, jolloin moni ongelma jäi havaitsematta ajoissa. Järjestelmiä kehitettäessä tulisi kehittämiseen varata *riittävästi aikaa*. Kuten yllä huomattiin, hankaloittaa kiire niin ongelmien havaitsemista, kuin niiden korjaamistakin.

Ympäristö

Järjestelmiä kehittävien organisaatioiden toimintaympäristön tulisi olla mahdollisimman *stabiili*, jotta ympäristöstä tulevat muutospainet eivät vaikeuttaisi järjestelmien kehitystä edellisessä osassa kuvatulla tavalla.

Organisaatio- ja työkuulttuuri

Organisaatio- ja työkuulttuurilla voi olla merkittävä rooli käytettävyyso- ja muiden ongelmien havaitsemisessa ja korjaamisessa järjestelmien kehityksen aikana. Mikäli kuulttuuri on *avoin*, on kehityksessä mukana olevien henkilöiden helpompi nostaa esiin havaitsemiaan ongelmia ilman pelkoa että heidät leimattaisiin ”hankaliksi” tai ”pahanilmanlinnuiksi”. Tämän toteutuminen on helpompaa, mikäli organisaatiot ovat alun perin ottaneet käyttöön jonkin *iteratiivisen kehitysmallin* järjestelmien kehityksen

pohjaksi. Iteratiivisissa kehitysmalleissa kantavana ajatuksena on ongelmien asteittainen löytäminen ja korjaaminen. Tällöin ongelman esiin tuominen on varsin luonnollista kaikille kehitykseen osallistuville. Ongelmien avointa käsittelyä tukee myös *joustava hallinnointi*, joka mahdollistaa resurssien uudelleen ohjaamisen ongelmien selvittelyyn nopeasti ja tehokkaasti aina tilanteen mukaan. Tällaisessa hallintomallissa keskitytään enemmän päätösten aikaansaamaan lopputulokseen kuin itse päätösprosessiin, jolloin prosessi voidaan suunnitella mahdollisimman kevyeksi.

Ehkä tärkein organisaatio- ja työkuultuuriin liittyvä asia joka vaikuttaa käytettävyysongelmien havaitsemiseen ja korjaamiseen on organisaatioiden kyky huomioida järjestelmien kanssa kosketuksiin tulevat *työprosessit, käyttäjät* ja ennen kaikkea *asiakkaat*. Asiakas- ja käyttäjälähtöinen ajattelu asettaa järjestelmien käyttäjät ja organisaation asiakkaat suunnittelun lähtökohdaksi, yleisemmän toiminnallisuuskeskeisen ajattelun sijaan. VERO-, REKI-, ja PIIKO- järjestelmät on kehitetty pitkälti toiminnallisuuksien ehdoilla, jolloin käyttäjään ja asiakkaaseen vaikuttavat asiat ovat jääneet alemmalle prioriteetille. Tämä taas on johtanut käytettävyysongelmiin kaikkien kolmen järjestelmän tuotantokäytössä.

Yllättävää onkin että samoihin tai samankaltaisiin, ongelmiin on sorruttu kolmen erillisen järjestelmän kehittämisessä. Samojen ongelmien toistamista tulisi välttää kehittämällä organisaatiosta *oppiva organisaatio*, joka kykenee muuttumaan ympäristön mukana ja oppimaan tekemistään virheistä.

Arkkitehtuuri

Järjestelmien käytettävyysongelmien löytämistä edistää *selkeä ja stabiili tietojärjestelmäarkkitehtuuri*. Kun arkkitehtuuri on selkeä, toisin kuin ATJn tietomallin ja tietojen välityksen kohdalla oli, on kehitykseen osallistuvilla todennäköisesti paremmat mahdollisuudet havaita ongelmia. Arkkitehtuurin on kuitenkin oltava niin *joustava* että siitä voidaan poiketa tarpeen niin vaatiessa. Tämä mahdollistaa sellaiset ongelmien ratkaisut jotka eivät ole mahdollisia alkuperäisessä arkkitehtuurissa.

Testaus

Tietojärjestelmien testauksessa havaitaan suuri osa käytettävyysongelmista. Testaus on kuitenkin tehtävä *riittävän ajoissa*, jotta havaittuihin ongelmiin voidaan reagoida.

Lisäksi testauksessa tulisi olla käytössä *kattavat testitapaukset*, jotta mahdollisimman suuri osa järjestelmän käyttötilanteista on yleensä mahdollista testata. Mikäli testit ovat vielä *automatisoitavissa*, nopeuttaa se testikierrosten tekemistä ja siten tehtyjen korjausten testaamista. Lisäksi käytettävissä on oltava *riittävän hyvä testimateriaali*, jotta olisi mahdollista jäljitellä tuotannossa vastaantulevia tapauksia.

Käytettävyysohjelmien havaitsemisessa *käyttäjättestaus*, mieluiten järjestelmän tulevilla käyttäjillä, antaa erittäin arvokasta tietoa järjestelmän toiminnasta käyttäjien silmin. Mikäli käyttäjättestaus suoritetaan riittävän aikaisessa vaiheessa, on mahdollista hyödyntää käyttäjiltä saatavia parannusehdotuksia myös itse työprosessin kehittämiseen.

Ohjeistus

Järjestelmäkehitykseen liittyvät ohjeistuksen tarpeet ovat hieman ristiriitaiset. Ohjeistuksen tulisi olla *riittävä* jotta sen avulla selvittää hieman epätavallisistakin tilanteista kehityksen aikana. Riittävän laaja ohjeistus mahdollistaa ongelmien paikantamisen ja valmiiden ratkaisujen soveltamisen uudelleen esiin tuleviin ongelmiin. Ohjeistuksen tulisi myös *olla sisäistettävissä*, niin että kaikki osapuolet ymmärtävät ainakin oman vastualueensa, ja siihen suoraan liittyvien vastualueiden ohjeistuksen. Mikäli toteuttajat eivät kykene sisäistämään ohjeistusta, ei siitä ole heille mitään apua. Tämä tarkoittaa että ohjeistuksen on oltava niin tiivis että se on mahdollista lukea muiden töiden ohella. Lisäksi sen on oltava niin helppolukuinen että siihen voi palata aina epäselvän paikan tullen.

Ohjeistuksen tulisi myös *kehittyä*. Hyvässä ohjeistuksessa on mukana aiemmin vastaan tulleita esimerkkejä käytännön ongelmista ja niiden ratkaisuista. Myös muualla kehitetyt ”hyvät toimintatavat” tulisi huomioida ohjeistusta laadittaessa. Kaikkea ei kannata kehittää itse.

7.4. Ongelmien lähtökohdat

Perusongelmaan, eli käytettävyysohjelmien ”siementen” syntymiseen vaikuttaa tämän tutkimuksen perusteella pari ydintahoa. Organisaation lähtökohdat ovat yksi voimakas ongelmien lähde. Mikäli organisaation prosesseissa ei huomioida käyttäjiä tai asiakkaita, ei näin tule olemaan järjestelmässäkään. Myös organisaation prosessien ollessa epäselvät

tai kehityksen alaiset, ei järjestelmäkehityksen tuloksena todennäköisesti ole käyttäjäystävällistä tai muutenkaan selkeää järjestelmää.

Suoraan järjestelmäkehityksen kannalta ongelmallisimpana on kuitenkin organisaation omien odotusten hallinnointi. Mikäli organisaatiolla ei ole aiempaa kokemusta järjestelmäkehityksestä, voi organisaation odotukset järjestelmäkehityksen monimutkaisuudesta, kestosta ja kustannuksesta olla varsin epärealistiset. Tämä näkyy erityisesti järjestelmäkehityksen kilpailuttamistilanteissa, jolloin tilaajaorganisaatio helposti painottaa liikaa projektin kustannuksia. Tällöin pienemmälle huomiolle jää järjestelmän koko elinkaaren kustannukset, puhumattakaan järjestelmän immateriaali-arvoista kuten käytettävyydestä, laajennettavuudesta ja ylläpidettävyydestä. Käytettäessä pelkkiä projektin kustannuksia kilpailutekijänä, on toimittajalla suuri houkutus aliarvioida työn määrä kilpailun voittamiseksi. Tämä johtaa lähes väistämättä yllä mainittuihin ongelmien löytymistä ja korjaamista estäviin tekijöihin, kuten kiireeseen, resurssipuutteeseen ja yleiskuvan puuttumiseen.

8. Suositukset

Tässä luvussa kuvaan miten Liikenteen turvallisuusvirasto voi vähentää tai välttää käytettävyysoongelmia tulevaisuudessa. Tarkastelen myös miten prosesseja voidaan parantaa jotta tähän päästäisiin, missä kehitysprosessin vaiheissa tulee olla erityisen varovainen ja mitä asioita Liikenteen turvallisuusviraston tulisi muuttaa toiminnassaan tulevia järjestelmäkehitysprojekteja silmällä pitäen.

8.1. Lähtökohtien kehittäminen

Järjestelmän tilaajaorganisaatiolla on oltava organisaation omien lähtökohtien hallinta riittävän hyvällä tasolla. Tämä tarkoittaa oman nykytilan ja tahtotilan hyvää tuntemusta. Ei ole realistista odottaa tietojärjestelmäprojektin tuottavan virheettömän ja käytettävän lopputuloksen mikäli järjestelmän taustalla olevat lait, säädökset ja liiketoimintaprosessit ovat epäselviä tai jatkuvan muutoksen alla. Lähtökohtien hallintaan kuuluu myös organisaation omien prioriteettien tarkastelu ennen kehitysprojektiin ryhtymistä. Organisaation tulisi tarkastella mitä asioita kehitysprojektissa tulee priorisoida ja miten kehitysprojektin tuloksena oleva järjestelmä nivoutuu olemassa oleviin prosesseihin, käyttäjiin ja asiakkaisiin. Myös projektin jälkeisestä ajasta tulisi olla toimiva kuva. Onko järjestelmän toiminnallisuutta tarkoitus laajentaa tulevaisuudessa? Entä tulevatko järjestelmän käyttäjryhmät tai käyttäjäorganisaatiot muuttumaan, entä organisaation loppuasiakaskunta? Nämä asiat tulisi lisäksi priorisoida karkealla tasolla, jotta tiedetään mihin asioihin organisaatio aikoo panostaa ja missä järjestyksessä.

Liikenteen turvallisuusviraston kannalta tämä tarkoittaa että viraston tulisi tarkastella suhtautumistaan järjestelmiensä käyttäjiin sekä asiakkaisiinsa kokonaisvaltaisesti. Nykyisen ajattelumallin kääntäminen sellaiseksi että tarkastellaan miltä viraston kanssa asiointi eri tilanteissa näyttää loppuasiakkaan silmin, voisi tuoda mielenkiintoisia näkemyksiä järjestelmäkehitykseen. Erityisesti kansalaisten ulottuvissa olevien sähköisten palvelujen kanssa tätä tapaa tulisi suosia jo ”JHS 129 Julkishallinnon verkkopalvelun suunnittelun ja toteuttamisen periaatteet” (JHS suositukset 2006) vuoksi.

Kustannusten arviointi nykyistä laajemmalta pohjalta saattaisi nostaa esiin virastolle uusia tapoja toimia kustannustehokkaasti ja valtion tuottavuusohjelman (Valtiovarainministeriö 2010b) mukaisesti. Mikäli järjestelmän kehittämiskustannuksien lisäksi huomioidaan esimerkiksi järjestelmän elinkaaren ylläpitokustannukset, sekä järjestelmän käyttöön liittyvien ulkoistettujen toimintojen kustannukset kuten maksetut palvelukorvaukset, voi järjestelmän kokonaiskustannusrakenne poiketa selvästi nykyisen mallin mukaisesta. Tarkastelussa tulisi myös pyrkiä tunnistamaan tilanteita jotka ovat ongelmallisia esimerkiksi kilpailutuslainsäädännön osalta, kuten tilanteet joissa järjestelmää käyttäneellä taholla on selkeää kilpailuetua verrattuna järjestelmää käyttämättömään tahoon verrattuna. Tällaiset ongelmat voivat ilmetä esimerkiksi käytännön tarpeena osata järjestelmän dokumentoimattomia oikopolkuja ja kiertoteitä, jotta järjestelmän kanssa työskentely olisi tehokasta. Ominaisuuksilla jotka vaativat järjestelmän aiempaa tuntemusta ennen kuin työ on tehokasta, on taipumus korottaa palvelumaksuja ja rajoittaa kilpailua.

Ennen kehitysprojektin kilpailutusta tulisi tilaajaorganisaation myös luoda ensimmäiset määrittelyt järjestelmälle, joiden pohjalta kilpailutus tehdään. Näiden luomisessa olisi hyödyllistä käyttää monen eri sidosryhmän voimavaroja. Erityisesti järjestelmän tulevilta käyttäjiltä ja asiakkailta saadaan todennäköisesti jo tässä vaiheessa hyödyllisiä kehitysideoita itse järjestelmän toimintaan kuin siihen kuuluvaan prosessiin liittyen. Samassa yhteydessä kannattaakin tarkastella järjestelmän toimintaan liittyvien prosessien tilaa. Määrittelyjä koottaessa kilpailutusta varten kannattaa lisäksi tarkastuttaa määrittelyt jollain ulkopuolisella taholla, jotta mahdolliset ongelmakohtat havaittaisiin jo tässä vaiheessa. Ulkopuolinen tarkastelu helpottaa myös kilpailudokumentaation kirjoittamista järjestelmätoimittajien ymmärtämässä muodossa. Tämä pienentää tilaaja- ja toimittajaorganisaatioiden erilaisista toimintaympäristöistä johtuvien väärinymmärryksien vaaraa. Kilpailutuksessa käytettäviä määrittelyjä luotaessa tulisi myös muistaa että tässä vaiheessa määritellään tahtotilaa, ei teknisiä ratkaisuja jolla siihen päästään.

Tilaajaorganisaation asettamia vaatimuksia käytettävän teknologian suhteen tulisi myös varoa. Koska tilaajaorganisaatiolla ei ole toimittajien veroista kokemusta järjestelmäkehityksestä, ei sillä myöskään ole toimittajien veroista tietoa eri teknologioiden toimivuudesta tahtotilan mukaisessa ympäristössä. Mikäli ulkopuoliset

tahot, kuten lainsäädäntö tai viranomaisyhteistyö, vaatii tiettyjen teknologioiden käyttämisen tietyissä tilanteissa, tulee nämä tilanteet ja teknologiat kartoittaa, ja liittää mukaan kilpailutuksessa käytettävään materiaaliin.

Hyvien määrittelyjen tekeminen on vaikeaa, mutta panostamalla tähän on mahdollista välttyä suuremmilta ja kalliimmilta ongelmilta myöhemmin järjestelmäkehityksessä. Parhaimmillaan määrittelyjen pohjalta voidaan jo aivan järjestelmäkehityksen alussa tehdä käyttöliittymäprototyyppejä sekä testitapauksia vahvistamaan tehtyjä valintoja.

8.2. Kilpailutuksen kehittäminen

Toimittajia valitessa tulisi nykyistä kilpailutustapaa kehittää. Yllä mainittuja järjestelmän elinkaarikustannuksiin vaikuttavia asioita tulisi painottaa enemmän kilpailutustilanteessa kuin pelkkiä kehitysprojektin kustannuksia. Lisäksi toimittajia tulisi kannustaa realistisiin työmääräarvioihin esimerkiksi sopivalla tavoitehintasopimuksella. Tällöin toimittajien marginaali putoaisi sopivalla progressiolla sovitun työmäärän jälkeen, jolloin toimittajan intressissä on mahdollisimman tarkan työmäärä-arvion esittäminen tarjouksessaan.

Kilpailutuksessa tulisi myös huomioida toimittajien käyttämien toteuttajien sisäänajo haluttuun tekniikkaan tai ohjeistukseen, esimerkiksi erillisellä sisäänajotaksalla. Tämä mahdollistaisi toimittajille nykyistä enemmän joustoa resurssivajetilanteissa ja tilaajalle tämä takaa projektin tasaisemman etenemisen. Kummatkin osapuolet hyötyisivät todennäköisesti tähän liittyvästä toteuttajien osaamisen seurannasta. Järjestelmätoimittajan kilpailutuksessa tulisi myös huomioida käyttöpalvelut, erityisesti testiympäristöjen ja tietoliikenneyhteyksien osalta. Näiden joustava toiminta on tärkeää koko kehitysprojektin edistymiselle. Kilpailutusmateriaalista tulisikin selkeästi käydä ilmi mitkä testiympäristöt ovat järjestelmätoimittajan ja mitkä käyttöpalvelutoimittajan vastuulla.

8.3. Järjestelmäkehitysprosessin kehittäminen

Varsinaisessa järjestelmäkehityksessä tulisi harkita onko perinteinen vesiputousmalli paras vaihtoehto järjestelmäkehitysmalliksi, vai voisiko sen korvata tai yhdistää muihin

järjestelmäkehitysmalleihin. Erityisesti testaus riittävän kattavalla materiaalilla ja tuotannonomaisella kuormalla tulisi ajoittaa siten että testeissä havaittuihin toiminnallisuus- ja käytettävyysoongelmiin on mahdollista kehittää laajojakin ratkaisuja. Tärkeää on myös että tilaaja- ja toimittajaorganisaatioissa on olemassa henkilöt joilla on selkeä kokonaiskuva kokonaisarkkitehtuurista, järjestelmän tahtotilasta ja kehityksen nykytilasta. Tällä tavoin voidaan tunnistaa paremmin tilanteet, joissa voidaan yhdistää metodeja ja uudelleen käyttää koodia niin yhden järjestelmän sisällä, kuin useiden järjestelmien välillä. Lisäksi hyvä yleiskuva mahdollistaa vertailun muiden järjestelmien kanssa, jolloin järjestelmien yhteisiin ongelmiin voidaan hakea ratkaisuja keskitetysti. Hyvä kokonaiskuva auttaa myös tunnistamaan valittujen arkkitehtuuriratkaisujen heikkoudet, sekä löytämään mahdollisia kiertoteitä arkkitehtuuriratkaisujen tai valitun teknologian asettamiin rajoituksiin.

Järjestelmäkehityksen organisoinnin on oltava riittävän tarkasti järjestettyä jotta vastuut ja osaamisalueet ovat kaikille selvät, mutta kuitenkin niin joustavaa ja kevyttä että eteen tuleviin ongelmiin ja vähäpätöisiltäkin vaikuttaviin kysymyksiin saadaan kehitettyä ratkaisut joustavasti ja nopeasti. Tämä tarkoittaa että yleisen ohjeistuksen on oltava riittävä ja selkeä, mutta kuitenkin niin suppea että kaikki osapuolet voivat sisäistää sen. Tarkempia ohjeistuksia kannattaa toki käyttää tärkeimmillä osa-alueilla, kuten tietomallit, järjestelmien väliset liittymät sekä käyttöliittymä, mutta ohjeistuksen perusteiden tulisi olla kaikkien käytettävissä ja ymmärrettävissä. Kommunikaatio toimittajan ja tilaajan henkilöstön välillä tulisi olla avointa ja tiivistä, jolloin eteen tuleviin ongelmiin saadaan haettua ratkaisuehdotukset tai valmiit ratkaisut joustavasti ja nopeasti. Ohjeistuksen ja päätösprosessien jalkauttamiseksi voi olla hyödyllistä järjestää tasaisin väliajoin lyhyitä koulutuksia joissa käydään läpi asioiden vastuhenkilöt sekä eri asianhaaroissa tarvittavia päätösprosessit. Samalla toteuttajat tapaavat kasvokkain, mikä yleensä parantaa henkilöiden välistä kommunikaatiota.

Liikenteen turvallisuusviraston osalta tämä tarkoittaa hallinnon selkeyttämistä ja päätösprosessien nopeuttamista. Ei ole hyväksyttävää että jonkin havaitun ongelman ratkaisuun tarvittavaa päätöstä joudutaan odottamaan pahimmillaan kuukausia. Tämän tilanteen saavuttamiseksi tulisi erilaisten työryhmien tarpeellisuutta tarkastella uudestaan ja harkita vastuun jalkauttamista kustakin asiasta parhaiten tietävälle henkilölle mahdollisuuksien mukaan. Samalla on kasvavia henkilöriskejä kompensoitava

varmistamalla että henkilöillä on riittävästi työaikaa oman osaamisen ylläpitoon juoksevien asioiden käsittelyn lisäksi. Henkilöriskejä tulee myös hallita sopivin varahenkilöjärjestelyin, niin ettei vastaan tule tilannetta jossa yhden henkilön sairastuminen tai siirtyminen muualle pysäyttää päätöksenteon.

Käytettävyyssuunnittelun, ja siihen kuuluvan käyttäjätestauksen, tärkeyttä täytyy korostaa järjestelmien käytettävyysohjelmien löytämisessä ja ratkaisuvaihtoehtojen arvioinnissa. Huomioimalla järjestelmien loppukäyttäjät jo suunnittelu- ja toteutusvaiheessa, voidaan monet käytettävyysohjelmat välttää kokonaan tai korjata ennen tuotantoon siirtoa. Tämä johtaa helpommin käytettäviin järjestelmiin, joiden käyttö on helppo oppia ja joita on tehokasta käyttää. Näillä argumenteilla voidaan hillitä sopimuskumppaneille ulkoistettujen toimintojen kustannusten nousua. Hyvä käytettävyys estää myös järjestelmien käytön rajoittumisen vain pitkän kokemuksen omaavien henkilöiden piiriin, helpottaen näin tilaajaorganisaation ja sopimuskumppaneiden henkilöstöhallintaa. Käytettävyyden huomioimisessa järjestelmäkehityksessä kannattaa ottaa oppia osassa ”3.2.4 Käytettävyyden sisällyttäminen nykyiseen järjestelmäkehitykseen” mainituista lähteistä.

8.4. Elinkaariajattelun kehittäminen

Järjestelmän siirtyminen tuotantoon ei ole sen kehityksen päätepiste, vaan enemmänkin sen alku. Tuotantoon siirtymisen jälkeen löydetään yleensä vielä parhaimmistakin järjestelmistä toiminnallisia ja käytettävyysohjelmia, joita ei kehityksen aikana löydetty. Näiden korjaaminen tulee mahdollistua, sillä ongelmien poistuminen parantaa käyttäjien työtehoa ja vähentää työssä syntyviä virhetilanteita, joiden korjaaminen on usein hankalaa ja kallista. Järjestelmää tulee myös ylläpitää siten että se vastaa muuttuviin tietoturva- ja teknisiin tarpeisiin, kuten esimerkiksi alustan tai käyttöjärjestelmän vaihtoon.

Järjestelmää tulee myös kehittää prosessimielessä. Organisaatioiden toimintaympäristön muuttuessa ei ole järkevää antaa järjestelmään sijoitetun pääoman vanhentua. Järjestelmiä tulee kehittää siten että hyödynnetään uuden tekniikan ja uusien prosessien suomia mahdollisuuksia järjestelmän tehokkaampaan käyttöön toimintojen automatisoinnin, paremman käyttöliittymän, tai esimerkiksi kansalaisten sähköisen

asioimisen avulla. Tämä kehitys tulee myös huomioida prosessien osalta. Tasaisin väliajoin olisi suotavaa tarkastella liiketoimintaprosesseja ja niihin liittyviä järjestelmiä kokonaisuutena ja miettiä miten niitä voitaisiin parantaa.

8.5. Organisaation oppimisen kehittäminen

Yhden tutkimuksen piirissä ei ole mahdollista tehdä suosituksia organisaation järjestelmäkehityksen parantamisessa kaikissa mahdollisissa tilanteissa. Yleisesti voidaan kuitenkin todeta että seuraamalla yllä mainittuja ehdotuksia ei todennäköisesti ainakaan toisteta jo tehtyjä virheitä. Tämä onkin ehkä organisaation kannalta se tärkein oppitunti.

Organisaation tulisi oppia tekemästään ja olla valmis kokeilemaan jotain uutta mikäli vanhaan ei olla tyytyväisiä. Tämä koskee käytettyjä tekniikoita, hallintotapoja, prosesseja ja kaikkea muuta organisaation toimintaan liittyvää. Oppiva organisaatio kestää parhaiten sisäiset ja ympäristön tuomat haasteet, koska se ei tee samoja virheitä uudestaan. Organisaatioiden oppimisesta on olemassa paljon kirjallisuutta, josta esimerkiksi Leenamajja Otalan ”Oppimisen etu – kilpailukykyä muutoksessa” (Otala 1996.) on hyvä teos aiheeseen tutustuttaessa.

Suhteellisen nopeita tuloksia voi olla mahdollista saada aikaan esimerkiksi vapauttamalla ohjeistuksen päivitys nykyistä joustavammaksi. Ajan tasalla oleva ohjeistus tukee paremmin järjestelmäkehitystä, koska sieltä on löydettävissä eri tahojen ratkaisemia ongelmia, jolloin kaikkien ei enää tarvitse ”keksiä pyörää uudestaan”.

9. Yhteenveto ja pohdintaa

Tässä luvussa kertaan lyhyesti tutkimuksen kulun ja esittelen lyhyesti yhteenvedon tutkimuksen tuloksista. Lopuksi tarkastelen lyhyesti tutkimuksen laajennettavuutta sekä sen aikana opittuja asioita.

Tutkimuksessa selvitin mitä keskeisiä käytettävyysoongelmia käyttäjien mielestä esiintyy Liikenteen turvallisuusviraston VERO-, REKI- ja PIIKO- järjestelmässä. Selvitin myös mitkä ovat näiden ongelmien taustalla olevat syyt. Selvittelyssä käytin apunani käyttäjien ja järjestelmien kehityksessä mukana olleiden ihmisten haastatteluja sekä järjestelmien ja niihin liittyvien kehitysprojektien dokumentaatiota. Hahmotin ongelmien taustalla olevia syitä piirtämällä haastatteluiden ja dokumenttien pohjalta affiniteettidiagrammit havaituista ongelmista ja niiden syiden välisestä verkostosta. Kootun aineiston avulla kehitin karkean mallin käytettävyysongelmien syntymekanismin kuvaamiseen. Lopuksi tein Liikenteen turvallisuusvirastolle tutkimuksen tulosten perusteella suosituksia, joiden avulla on mahdollista vähentää käytettävyysongelmien esiintymistä tulevilla järjestelmäkehityksissä.

9.1. Yhteenveto tutkimuksen tuloksista

Tässä osassa esittelen lyhyen yhteenvedon tutkimuksen tuloksista. Ensimmäiseen tutkimuskysymykseen ”*Mitkä ovat tutkittavien järjestelmien keskeiset käytettävyysongelmat*” sain tyydyttävät vastaukset. Järjestelmien keskeiset käytettävyysongelmat löytyivät ja ne on kuvattu yksityiskohtaisesti luvussa ”5 Järjestelmien keskeiset käytettävyysongelmat”. Kaikille järjestelmille yhteisiä ongelmia oli ongelmat nopeuden ja luotettavuuden kanssa. Myös järjestelmien rajoittunut selaintuki aiheutti ongelmia Liikenteen turvallisuusviraston ulkopuolella. Järjestelmät eivät myöskään ohjaa käyttäjän toimintaa riittävän hyvin ja navigointi on usein opittava jokaiseen tehtävään erikseen. Lisäksi järjestelmiin liittyvät työprosessit ovat ajoittain monimutkaisia, mikä vaikeuttaa myös itse järjestelmän käyttöä. Arkkitehtuuritasolla järjestelmien tietomallit ovat monimutkaisia, jolloin tietojen korjauksia ei aina uskalleta tehdä kun ei ole varmaa mitkä korjauksen vaikutukset olisivat. Järjestelmien sisältämän

suuri tiedon määrä on myös ongelmallista järjestelmien nopeuden sekä tiedon hallinnan kannalta.

Toiseen tutkimuskysymykseen ”*Miten nämä keskeiset käytettävyysongelmat ovat syntyneet järjestelmäkehityksen aikana*” sain myös vastauksia tämän tutkimuksen aikana. Ongelmien syyt on kuvattu yksityiskohtaisesti luvussa ”6 Käytettävyysongelmien syyt”. Kaikissa tutkituissa järjestelmissä osa ongelmista johtui Liikenteen turvallisuusviraston käyttämästä vesiputousmallin mukaisesta järjestelmäkehitysmallista ja käyttöliittymästandardista, sekä liian tulkinnanvaraisista tai toteutuksen liian tarkasti kiinnittävistä määrittelyistä. Liikenteen turvallisuusvirasto kehitti lisäksi järjestelmiä toimintopainotteisesti, eikä asiakas tai käyttäjänäkökulmaa juurikaan huomioitu. Ongelmallista oli myös teknisen arkkitehdin puute järjestelmäkehityksen alkuvaiheissa. Tämä johti puutteisiin yleiskuvan hahmottamisessa, ja sitä kautta ongelmiin työn koordinoimisessa ja yhteisten ratkaisujen edistämisessä. Myös tietyt valittuun tietojärjestelmäarkkitehtuuriin kuuluvat, kaikille järjestelmille yhteiset, ominaisuudet olivat monen käytettävyysongelman taustalla. Esimerkkinä tällaisesta ominaisuudesta on monimutkainen tietomalli. Ongelmien syitä löytyi myös Liikenteen turvallisuusviraston organisaatiokulttuurista, joka ei edistänyt projektien välistä vapaata ja avointa kommunikaatiota vaan painotti dokumentaatiota ja projektien raskasta hallintomallia.

Viimeiseen tutkimuskysymykseen ”*Voidaanko ongelmien synnylle kehittää sopivaa mallia*” sain myös vastauksen. Ongelmien syntymekanismille kehittämäni malli on kuvailtu yksityiskohtaisesti luvussa ”7 Johtopäätökset”. Malli perustuu ajatukselle että käytettävyysongelmien perussyöt löytyvät yleensä järjestelmäkehityksen alkupäästä. Kehitysprosessin aikana tietyt tekijät estävät ongelmien havaitsemista ja korjaamista ennen niiden pääsyä tuotantoon. Näitä syitä tasapainottavat toiset tekijät, jotka puolestaan edistävät ongelmien havaitsemista ja korjaamista ennen järjestelmän siirtymistä tuotantoon.

Tutkimuksen aikana tekemäni havaintojen ja tulkintojen pohjalta annoin Liikenteen turvallisuusvirastolle suosituksia toimintansa parantamiseen, jotta käytettävyysongelmia voitaisiin vähentää tulevilla järjestelmäkehitysprosesseissa. Suositukset kokonaisuudessaan löytyvät luvusta ”8 Suositukset”. Ne liittyvät organisaation omien lähtökohtien ja kilpailutuksen parantamiseen sekä käytetyn järjestelmäkehitysprosessin

kehittämiseen. Annoin myös suosituksia organisaation elinkaariajattelun ja organisaation oppimisen kehittämiseen.

9.2. Pohdintaa

Koska tutkimus käsittelee yleisiä ongelmia tietojärjestelmäkehityksessä, olisi hyvä jos tutkimuksen tuloksia voitaisiin käyttää myös laajemmin kuin pelkästään Liikenteen turvallisuusviraston VERO-, REKI- ja PIIKO- järjestelmien järjestelmäkehityksen osalta. Uskon että kehittämäni malli on täysin käyttökelpoinen myös muissakin järjestelmäkehitystilanteissa. Malli ei tee oletuksia järjestelmää kehittävistä organisaatioista, vaan toimii enemmänkin tarkistuslistana jonka avulla organisaatiot voivat tarkistaa järjestelmäkehitysprosessinsa toimivuutta. Mallissani mainitut käytettävyysongelmien havaitsemista ja korjaamista estävät tai edistävät tekijät tulevat todennäköisesti vastaan useimpien organisaatioiden järjestelmäkehityksessä. Tämän vuoksi uskon että mallini on hyödyllinen useimmille järjestelmiä teettäville ja tekevillä organisaatioille.

Törmätessäni kevään 2009 opinnoissani John Landryn artikkeliin ”*Can computing professionals be the unintentional architects of evil information systems?*” (Landry 2008), en aavistanut minkälainen matka minulla on edessä. Artikkelin herätti minut miettimään Liikenteen turvallisuusviraston järjestelmien käytettävyyttä, ja sitä mistä käytettävyysongelmat oikein syntyvät. Tämän tutkimuksen aikana tulin tekemisiin monen näitä järjestelmiä työssään käyttävän henkilön kanssa. Voin vain, Eva Olssonin tavoin (Olsson 2004), nöyränä todeta että työhönsä motivoituneet käyttäjät tyytyvät hankaliin työoloihin ja kehittävät itse tapoja kiertää ongelmat työvälineissään. Tämän tulisikin toimia muistutuksena kaikille, erityisesti meille jotka olemme mukana kehittämässä toisten ihmisten työvälineitä. Jos teemme työmme huolella ja huomioimme käyttäjät sekä heidän kanssaan asioivat asiakkaat, voimme helpottaa monen ihmisen työtä ja asiointia. Päättyessäni tätä työtä mieleeni nousi Arthur R. Millerin sanat: ”*Man must shape his tools lest they shape him*”. Toivottavasti tämän työn lukijat ottavat tästä vaarin ja muotoilevat luomansa työkalut niitä käyttäville ihmisille, ettei heidän tarvitse mukautua työkaluihin.

Lähteet

- Anandhan, A., Dhandapani, S., Reza, H. & Namasivayam, K. 2006, "Web Usability Testing — CARE Methodology", *Information Technology: New Generations, 2006.ITNG 2006.Third International Conference on*, , pp. 495-500.
- Benyon, D., Turner, P. & Turner, S. 2005., *Designing interactive systems : people, activities, contexts, technologies*, Addison-Wesley, Harlow.
- Bosch, J. 2003, "Designing software architectures for usability", *Proceedings - International Conference on Software Engineering*, , pp. 757-758.
- Donahue, G.M. 2001, "Usability and the bottom line", *Software, IEEE*, vol. 18, no. 1, pp. 31-37.
- Ferre, X., Jurista, N., Windl, H. & Constantine, L. 2001, "Usability basics for software developers ", *IEEE Software*, vol. 18, no. 1, pp. 22 <last_page> 29.
- Hanhisalo, J. 2008, *Määrittelyvaiheen vaatimusten vaikutukset käyttöliittymään*, University of Helsinki, Faculty of Science, Department of Computer Science.
- Heiskari, J., Kauppinen, M., Runonen, M. & Mannisto, T. 2009, "Bridging the Gap Between Usability and Requirements Engineering", *Requirements Engineering Conference, 2009.RE '09.17th IEEE International*, , pp. 303-308.
- Hyysalo, S. 2006, *Käyttäjätieto ja käyttäjätutkimuksen menetelmät*, Edita, Helsinki.
- IEEE standards 1998, *IEEE Standard for a Software Quality Metrics Methodology*.
- ISO9241-11, S.S. 2000, "SFS-EN ISO 9241-11 Näyttöpäätteillä tehtävän toimistotyön ergonomiset vaatimukset. Osa 11: Käytettävyyden määrittely ja arviointi", *SFS-EN ISO 9241-11*, .
- JHS suositukset 2006, 2006-05-24-last update, *JHS-suositukset - JHS 129*. Available: <http://www.jhs-suositukset.fi/web/guest/jhs/recommendations/129> [2010, 4/13/2010] .
- Juristo, N. 2007, "Guidelines for eliciting usability functionalities", *IEEE Transactions on Software Engineering*, vol. 33, no. 11, pp. 744-758.
- Kwang, B.L. & Grice, R.A. 2004, "Developing a new usability testing method for mobile devices", *Professional Communication Conference, 2004.IPCC 2004.Proceedings.International*, , pp. 115-127.
- Landry, J.R. 2008, "Can computing professionals be the unintentional architects of evil information systems?", *SIGMIS CPR '08: Proceedings of the 2008 ACM SIGMIS CPR conference on Computer personnel doctoral consortium and research*, , pp. 76-78.

- Leventhal, L.M. 2007, *Usability engineering process, products, and examples*, Pearson/Prentice Hall, Upper Saddle River, N.J.
- Mäkinen, V. 2005, *Ohjelmiston käyttöliittymään kohdistuvat riskit määrittelyvaiheen käyttöliittymäsuunnittelun jälkeen*, University of Helsinki, Faculty of Science, Department of Computer Science.
- Nielsen, J. 1993., *Usability engineering*, Academic Press, Boston.
- Olsson, E. 2004, *Designing Work Support Systems – For and With Skilled Users*, Uppsala University, Department of Information Technology; Acta Universitatis Upsaliensis; Uppsala University, Department of Information Technology.
- Otala, L. 1996., *Oppimisen etu : kilpailukykyä muutoksessa*, WSOY, Porvoo.
- Pietiläinen, P. 2001, *A process model for designing usable Web applications*, Helsinki University of Technology, Department of Computer Science and Engineering.
- Robertson, T. 2006, "Ethical issues in interaction design ", *Ethics and Information Technology*, vol. 8, no. 2, pp. 49 <last_page> 59.
- Troutt, M.D. 2007, "Some Usability and Ethical Issues for Captive End-User Systems", *Information Technology Newsletter. Vol.18*, vol. 18, no. 2, pp. 6-23.
- Valtiovarainministeriö 2010a, *Valtiovarainministeriö, tiedote 11/2010*, Valtiovarainministeriö, Helsinki.
- Valtiovarainministeriö 2010b, , *Valtiovarainministeriö: Tuottavuusohjelma*. Available: http://www.vm.fi/vm/fi/05_hankkeet/02_tuottavuusohjelma/index.jsp [2010, 4/15/2010] .

LIITE 1 - haastattelurunko

Haastateltava: PIIKO järjestelmän käyttäjä

Mikä on työtehtäväsi

Miten käytät piikoa työssä

Kuvaile tyypillinen tapaus kun käytät piikoa, eli miten käsittelet hakemuksia

Onko tullut vastaan seuraavia tapauksia:

- Piirturikortit
- kuljettaja
- yritys
- korjaamo
- valvonta
- ammattipätevyys (kuorma-auto / linja-auto)
- ammattipätevyys (taksi)

Toimiiko loogisesti omaan työhön nähden

- oma työjärjestys
- lomake

Mitkä asiat ärsyttävät eniten

Missä tulee helpoiten ongelmia / virheitä

Mitkä muut asiat tuntuvat hankalilta

Miten niiden pitäisi toimia

Mitä pitäisi muuttaa "jollain tapaa"

Minkälaista ohjeistusta / koulutusta on annettu?

Olitko mukana ihan alkuvaiheessa?

Mitä silloin tapahtui?

Pystyikö kehitykseen vaikuttamaan?

Miten palaute AKELLE/TraFille?

Kysy näistä jos ei tullut jo aiemmin esiin

- Kortin tilat
- Kuvat
- Kortti ei tullut
- Korvaava kortti
- maksutiedot ongelmia
- hitaus
- hakemusnäytöt / sekavuus
- lomake vs näytöt --> loogisuus
- tuleeko uusintahakemuksia koska tiedot väärin hyväksytyssä hakemuksessa