

Janika Toivonen

# Liikepaikkarekisterin kartoitus

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietoverkot

Insinööriytyö

18.8.2015

Tekijä(t) Otsikko	Janika Toivonen Liikepaikkarekisterin kartoitus
Sivumäärä Aika	47 sivua + 4 liitettä 18.8.2015
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Tietoverkot
Ohjaaja(t)	Ylläpitöpäällikkö Timo Hietaharju Tuotepäällikkö Timo Lahtela Yliopettaja Janne Salonen
<p>Insinööriyön tavoitteena oli selvittää liikepaikkarekisterinä toimivan Nasta-järjestelmän nykytila. Tarkoituksena oli tehdä esiselvitys järjestelmän uusimisprojektia varten, joka on tarkoitus aloittaa vuonna 2016. Kartoituksesta syntyi manuaali nykyjärjestelmästä sekä alustavaa vaatimusmäärittelyä ja käyttötapauskuvauksia.</p> <p>Työn teoriaosuudessa käsiteltiin ohjelmistosuunnittelun työn kulkua yleisesti, joista hieman laajemmin elinkaarimalleja, esitutkimusta, vaatimusmäärittelyä ja käyttöliittymäsuunnittelua. Esitutkimusta ja vaatimusmäärittelyä käsiteltiin enemmän siksi, koska oma työni käsittelee näitä aiheita. Erilaisia elinkaarimalleja käsiteltiin, koska mallit vaikuttavat projektien kulkuun ja projektien toteutukseen. Käyttöliittymäsuunnittelun halusin nostaa esiin teoriaosuudessa, koska monissa lähteissä ja elinkaarimalleissa käyttöliittymäsuunnittelua ei käsitellä ollenkaan. Käyttöliittymän suunnittelu tai suunnittelun puute vaikuttaa ensisijaisesti järjestelmän käytettävyyteen.</p> <p>Lopputuotoksena syntyi kokonaisuudessaan kolme dokumenttia, jotka jäivät Neste Markkinointi Oy:n käyttöön. Työn lopusta löytyy liitteinä osioita dokumenteista, mutta pääasiallisesti dokumenttien sisältö on salaista, eivätkä ne ole osana tätä insinööriyötä.</p>	
Avainsanat	Kartoitus, ohjelmistosuunnittelu, elinkaarimallit, esitutkimus, vaatimusmäärittely, käyttötapauskuvaus ja käyttöliittymäsuunnittelu.

Author(s) Title	Janika Toivonen Analysis of the Business Location Register
Number of Pages Date	47 pages + 4 appendices 18 August 2015
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Information Networks
Instructor(s)	Timo Hietaharju Maintenance Manager Timo Lahtela Product Manager Janne Salonen Principal Lecturer
<p>The purpose of this thesis was to examine the current state of a business location register information system called Nasta. The objective was to make a feasibility study for upgrading the system, which is scheduled to launch in 2016. The outcome of the survey was a manual of the current system, as well as a preliminary requirement specification and use case models.</p> <p>The theoretical part focuses on software engineering workflow in general, with a special focus on life cycle models, feasibility studies, requirements specification and user interface design. A variety of life cycle models are explored, because the models affect the progress of the projects and the implementation of projects. The user interface design is highlighted since many sources and life cycle models do not deal with interface design at all. Still, the interface design or a design flaw primarily affects the usability of the system.</p> <p>The outcome of this thesis consists of three documents that were created for Neste Marketing Ltd.'s use. Parts of these documents can be found attached to this thesis, but mainly the content of the documents is confidential and, as such, is not part of this thesis.</p>	
Keywords	Survey, Software development, Life cycle models, Feasibility study, Requirements specification, Use case model and interface design.

## Sisällys

### Lyhenteet

1	Johdanto	1
2	Neste	2
3	Ohjelmistosuunnittelu	4
3.1	Elinkaarimallit	4
3.2	Esitutkimus	13
3.3	Vaatimusmäärittely	14
3.4	Suunnittelu	19
3.5	Käyttöliittymäsuunnittelu	20
3.6	Toteutus ja integrointi	24
3.7	Testaus	24
3.8	Käyttöönotto ja ylläpito	25
3.9	Elinkaaren loppu	25
4	Työn taustat	26
4.1	Työn tarkoitus	26
4.2	Työn toteutus	27
4.2.1	Haastattelut	27
4.2.2	Tutustuminen järjestelmään	28
4.3	Työn suorittamisen haasteet	29
4.3.1	Haastattelut	29
4.3.2	Käyttöoikeudet	30
4.3.3	Ylläpito ja päivittäminen	30
5	Nasta-ohjelmisto	31
5.1	Järjestelmän tarkoitus ja liiketoimintaprosessit	31
5.2	Järjestelmän sisältämät liikapaikkatiedot	31
5.3	Järjestelmän käyttäjät	32
5.4	Ohjelmiston nykytilanne	33
6	Nasta-järjestelmän haasteiden ja ehdotuksien ylätasoinen tiivistelmä	34

6.1	Toiminnallisia vaatimuksia järjestelmälle	34
6.2	Järjestelmän datan hallinta	37
7	Käyttötapauskuvaukset	41
8	Yhteenveto	44
	Lähteet	45
	Liitteet	
	Liite 1. Nykytilanne dokumentista päävalikko esittely	
	Liite 2. Nykytilanne dokumentista vapaan asemanumeron haku	
	Liite 3. Nykytilanne dokumentista aseman perustaminen	
	Liite 4. Käyttötapaus: Aseman perustaminen PE_01	

## Lyhenteet

PEK	Polttoaineen erotuskaivo.
PEK-purku	Polttoaineen erotuskaivon purkupaikka.
Neste.fi	Nesteen asiakkaille suunnattu internetsivusto.
KauppiasExtranet	Nesteen kauppiaille tarkoitettu oman aseman tietojen hallinnointipaikka.
ISO	International Organization for Standardization eli ISO on kansainvälinen standardisoimisjärjestö.

## 1 Johdanto

Tämä insinööriyö on tehty Neste Markkinointi Oy:n tilauksesta. Työn tarkoituksena on kartoittaa perusliikepaikkarekisterinä toimivan Nasta-ohjelmiston nykytila. Systemaattisen kartoituksen välineinä käytettiin pääsääntöisesti käyttäjähaastatteluita ja ohjelmistoon tutustumista. Nykytilan kartoituksen sivutuotteena syntyivät järjestelmän haasteet ja ehdotukset -osio, joka on periaatteessa alustavaa vaatimusmäärittelyä. Lisäksi syntyi myös alustavia käyttötapauskuvauksia korvaavaan järjestelmään.

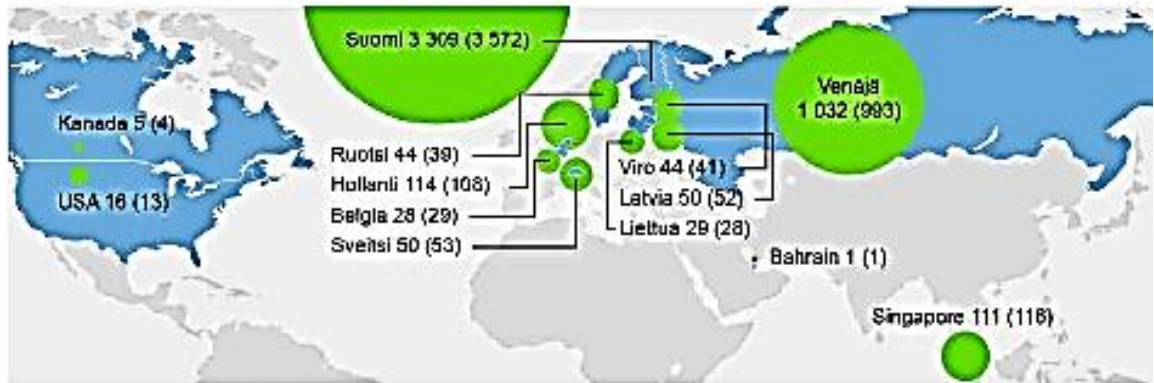
Työn toteutusperusteena on nykyohjelmiston tuleminen elinkaarensa päähän. Järjestelmä ei enää täytä nykyisiä vaatimuksia ja tarpeita. Myös järjestelmän tuki on päättynyt. Järjestelmään ei enää tehdä muutoksia, vaan sille tehdään ainoastaan pakolliset ylläpitoon liittyvät toimet.

Ohjelmisto on ollut käytössä jo 90-luvun alusta alkaen. Vaikka nykyisin järjestelmä ei enää palvele sidosryhmiään toivotulla tavalla, on pakko kuitenkin todeta, että ohjelmisto on kestänyt todella hyvin aikaa ja toiminut luotettavasti. Ohjelmistossa ei juuri koskaan ole häiriöitä ja monelle sen sidosryhmäläiselle se on ollut luotettava ja turvallinen työkalu jo pari vuosikymmentä.

Insinööriyössä käsitellään ohjelmistosuunnittelun vaiheita yleisesti sekä kerrotaan tarkemmin ja laajemmin esitutkimuksesta, elinkaarimalleista, vaatimusmäärittelystä, käyttötapauskuvauksista ja käyttöliittymäsuunnittelusta. Esitutkimusta, elinkaarimalleja, vaatimusmäärittelyä ja käyttötapauskuvauksia käsitellään työssä laajemmin, koska oma työni koskee näitä ohjelmiston kehitysvaiheita. Käyttöliittymä suunnittelun halusin nostaa esiin, koska ohjelmistojen elinkaarimalleissa ja ohjelmistosuunnitteludokumenteissa harvemmin käsitellään käyttöliittymä suunnittelua. Hyvä käyttöliittymäsuunnittelu on kuitenkin suoraan verrannollinen ohjelmiston käytettävyyteen.

## 2 Neste

Neste Oyj on suomalainen pörssiyhtiö, joka perustettiin vuonna 1948 huolehtimaan Suomen öljynhuollosta. Ensimmäinen jalostamo käynnistyi Naantalissa vuonna 1957 ja Porvoon jalostamo käynnistyi vuonna 1965. Nykyisin Nesteellä on tuotantolaitoksia myös Singaporessa ja Hollannissa sekä Bahrainissa. [1.] Yhteensä toimintaa on 14 maassa ja henkilöstöä noin 5000 (kuva 1) [2].



Kuva 1. Nesteen liiketoiminta alueet ja henkilöstömäärät [31 s.71].

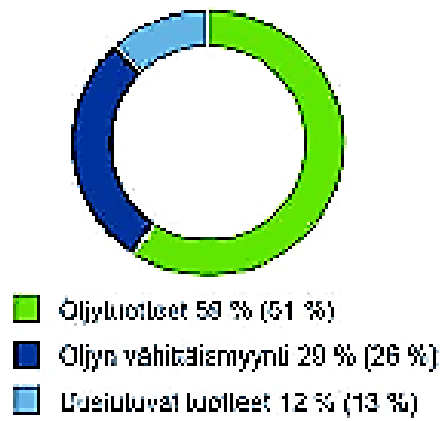
Neste myy tuotteitaan asemaverkoston kautta sekä suoratoimituksina ammattiliikenne-, teollisuus-, maatalous- ja lämmitysasiakkaille sekä jälleenmyyntiin. Nesteellä on Suomen kattavin asemaverkosto. Suomen lisäksi huoltoasema toimintaa on Virossa, Latviassa, Liettuaissa sekä Venäjällä Pietarin alueella. [2.]

Nesteen liiketoiminta-alueet ovat öljytuotteet, uusiutuvat tuotteet sekä öljyn vähittäismyynti (kuva 2). Uusiutuvien tuotteiden osuus liiketoiminnasta kasvaa vuosittain, ja Neste on maailman johtava uusiutuvan dieselin toimittaja. [1.] Nesteen jalostamia tuotteita ovat [2]

- bensiinit
- dieselpolttoaineet
- lento- ja laivaliikenteen polttoaineet
- kevyet ja raskaat polttoöljyt
- perusöljyt
- bensiinikomponentit



- erikoispolttoaineet, kuten pienmoottoribensiini
- liuottimet
- nestekaasut
- bitumit.



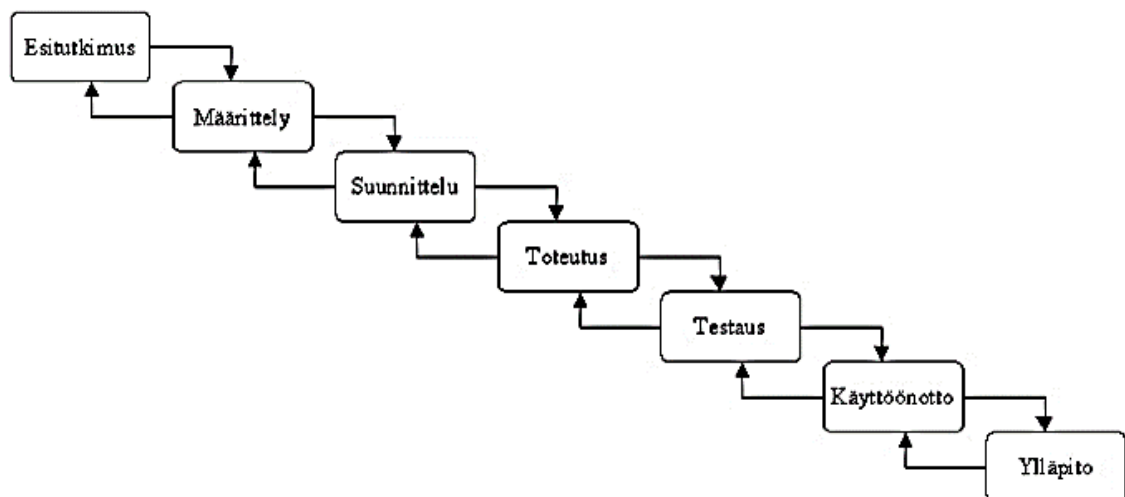
Kuva 2. Nesteen liiketoiminta-alueiden prosentuaaliset osuudet vuodelta 2013 [31 s. 17].

### 3 Ohjelmistosuunnittelu

#### 3.1 Elinkaarimallit

##### Vesiputousmalli

Vesiputousmalli (Waterfall Model) on kehitetty 1960-luvun lopulla [3]. Malli etenee lineaarisesti vaihe vaiheelta eteenpäin kuten vesiputous (kuva 3), josta se on saanut nimensä. Ideana on, että jokainen vaihe saatetaan loppuun ja hyväksytään, eikä päätettyyn vaiheeseen enää palata. Tähän ei kuitenkaan käytännössä koskaan päästä. [6.]



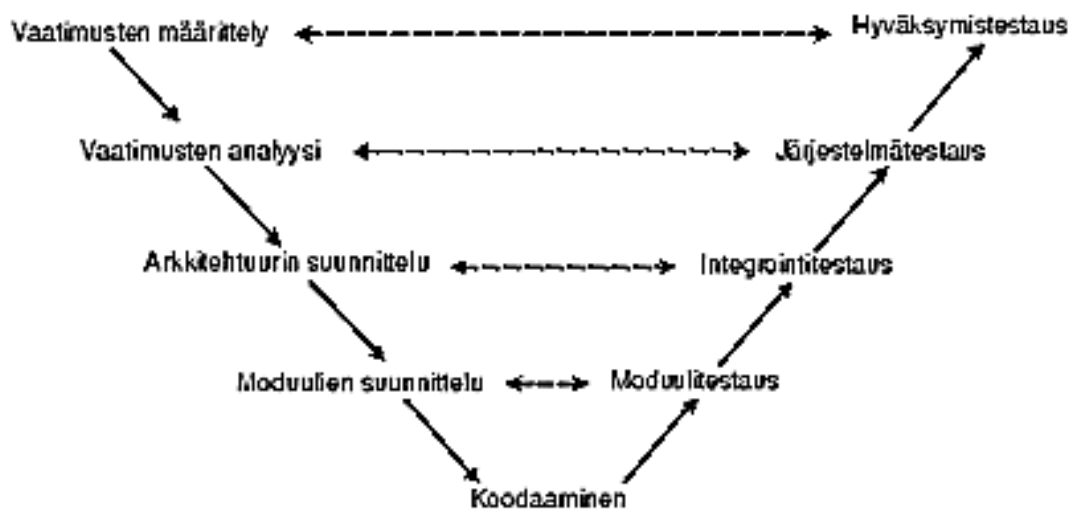
Kuva 3. Vesiputousmalli [3].

Vesiputousmallin etuja ovat, että jos suunnittelu tehdään alussa huolella, saadaan merkittäviä säästöjä projektin myöhemmissä vaiheissa. Virheet eli bugit, jotka löydetään varhaisessa vaiheessa ovat rahallisesti, työmäärällisesti ja ajallisesti halvempia korjata. Mallissa pyritään nimenomaan korostamaan suunnittelun tärkeyttä. Muita etuja vesiputousmallissa on helppo ymmärrettävyys ja hallittavuus, selkeät kokonaisuudet sekä projektin helppo seurattavuus. [17.]

Vesiputousmallin heikkouksina pidetään, että on lähes mahdotonta tehdä täydelliset suunnitelmat etukäteen, joihin ei tarvitsisi palata myöhemmin. Suunnittelijoiden on lähes mahdotonta ennakoita toteutusvaiheessa tulevia ongelmia. Käytännössä tämä tarkoittaa

taa vesiputousmallissa, että alkuperäiset suunnitelmat joudutaan hylkäämään ja suunnittelu aloittamaan alusta. Vesiputousmallissa myös aiemmissa vaiheissa tulleet virheet saattavat kertautua valtaviksi projektin edetessä. [17.]

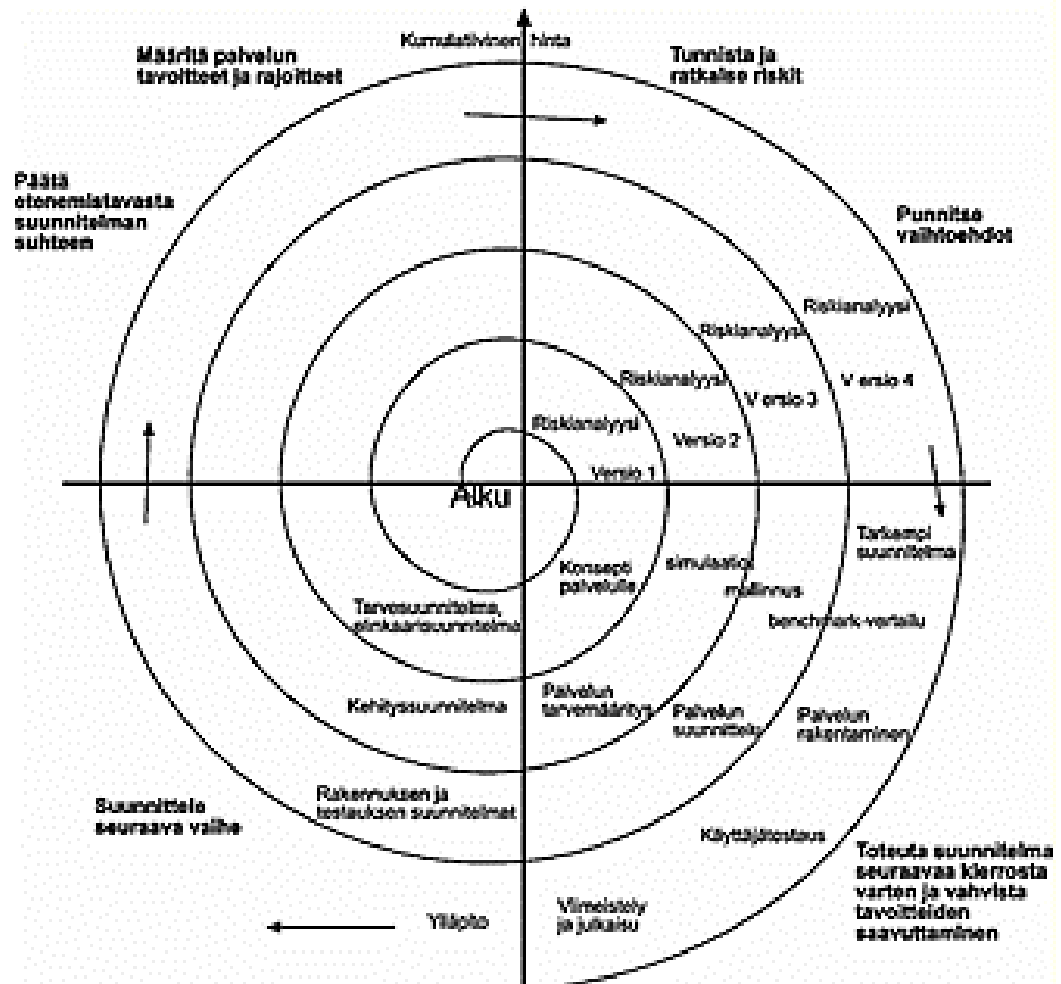
Vesiputousmallista on johdettu monia muunnoksia ja kokonaan uusia elinkaarimalleja. Yksi tunnetuimmista muunnoksista on kehittyneempi versio, jota kutsutaan V-malliksi (V-model). V-mallissa (kuva 4) korostetaan myös huolellista dokumentaatiota, mutta suurimpina eroavaisuuksina ovat ehkä tarkastukset. V-mallissa jokainen vaihe testataan ennen siirtymistä seuraavaan vaiheeseen. [32 s. 136–142.]



Kuva 4. V-malli [33].

## Spiraalimalli

Spiraalimallissa (Spiral Model) (kuva 5) keskeisinä ajatuksina ovat iteratiivisuus sekä riskien jatkuva analysointi ja prosessin uudelleen ohjaaminen analysointien tulosten mukaan. Malli on jaettu vaiheisiin, joita toistetaan siihen asti, että järjestelmä on valmis. [3.]



Kuva 5. Spiraalimalli [8].

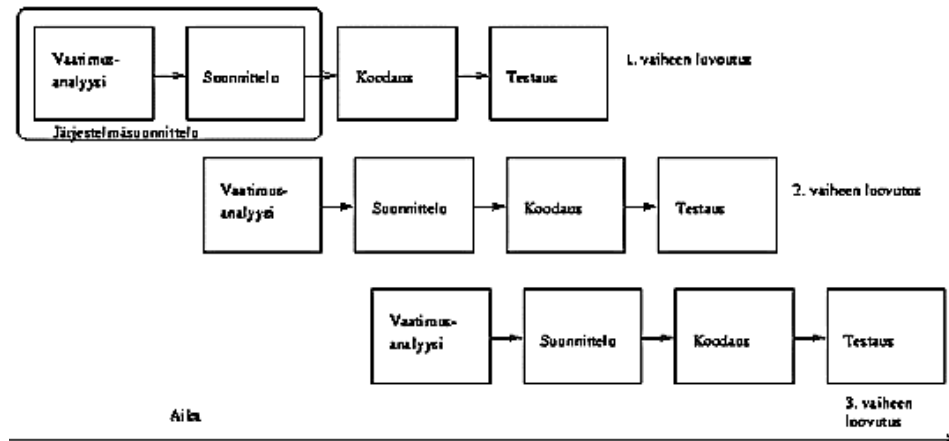
Yleensä mallin vaiheet on jaettu neljään osaan suunnitteluun (tavoite ja aikataulu sekä vaihtoehdot ja rajoitukset), riskianalyysiin (määritellään tärkeimmät riskit), tuotantoon (version valmistuminen) ja asiakkaan arviointiin (katselmointi) [3].

Spiraalimallin heikkouksina pidetään asiakkaan toistuvaa aktivointia, joka rasittaa kehitystyötä ja aiheuttaa ongelmia. Lisäksi mallin soveltaminen on hidasta. Yleensä spiraalimallissa riskit pienenevät jokaisella kierroksella, mutta on kuitenkin mahdollista, että hoitamattomat riskit kertaantuvat kierros kierrokselta moninkertaisiksi. [27 s. 33–35.]

Spiraalimallista on vain vähän käytännön kokemusta, eikä siitä ole vakiintuneita käytäntöjä. Spiraalimallia käytetään myös tuotteen ylläpitoon, kun asiakkaalle on luovutettu valmis tuote. Tällöin spiraaliin vaiheet liittyvät tuotteen ylläpitoon. [7.]

## Evoluutiomalli

Evoluutiomallissa (Evolution Model) ohjelmaa kehitetään väliversioiden avulla. Väliversiot voivat olla joko tuotantokäytössä olevia ohjelmia tai prototyyppejä. Evoluutiomallissa eri vaiheet tapahtuvat samanaikaisesti tai ainakin osittain rinnakkain (kuva 6). [25 s. 3-4.]



Kuva 6. Evoluutiomalli [7].

Mallissa lainataan lineaarisen, prototyyppisen ja iteroivan mallin parhaita puolia. Evoluutiomallissa käyttäjä antaa palautetta väliversiosta, jonka perusteella jatketaan kehitystä. [25 s. 3-4.]

Evoluutiomallin etuja ovat nopea reagointi tarpeisiin (virheet korjataan nopeasti, joka säästää rahaa), malli on helppo omaksua, kustannusarvio ja aikataulu on helppo tehdä sekä arkkitehtuuri pysyy yleensä hyvin hallinnassa [25 s. 3-4; 6 s. 7].

Mallin ongelmia ovat [25 s. 3-4]:

- Etenemisen seuranta. On vaikea tietää milloin ohjelma on valmis.
- Rakenteelliset ongelmat. Muutokset voivat rikkoa arkkitehtuurin.
- Sovelluskehittimen tarve.
- Asiakkaan on usein vaikea erottaa prototyyppiä ja lopputuotetta.

## Komponenttimalli

Komponenttimalli (Component Assembly Model) on hyvin samantapainen kuin spiraalimalli, vaikkakin sen perusajatus on erilainen. Komponenttimallissa käytetään iteraatiokierroksia kuten spiraalimallissa, mutta sen lisäksi tarkoituksena on käyttää mahdollisimman paljon valmiita komponentteja eli valmiita osia. [25 s. 4.]

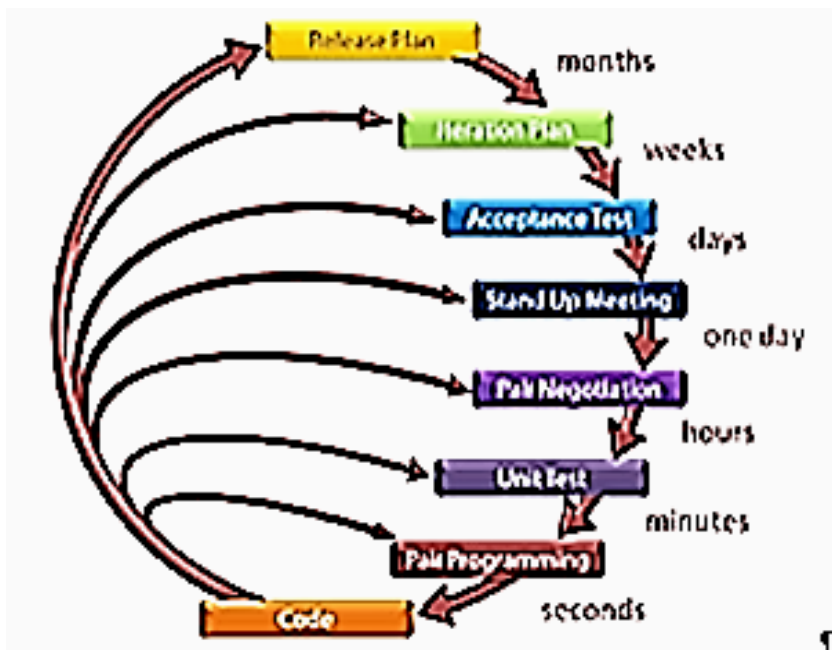
Valmiita komponentteja saadaan yleensä omista aiemmista projekteista, ostamalla sekä ilmaisista Free Software–projekteista [25 s. 4]. Jokainen spiraalin kierros aloitetaan etsimällä ohjelmakomponentteja (yleensä komponenttikirjastosta). Jos valmis komponentti löytyy, käytetään sitä, mutta jos ei, niin tehdään uusi komponentti, joka tallennetaan komponenttikirjastoon. [7.]

## Ketterä ohjelmistokehitys

Ketterä ohjelmistokehitys (Agile software development) on joukko käytettäviä menetelmistöjä. Kaikille ketterille ohjelmistokehitys menetelmille yhteisiä piirteitä on saada toimiva ohjelmisto, nopea muutoksiin reagointi, ohjelman kehittäminen lyhyin iteroinnein sekä suora viestintä. Ketterissä ohjelmistokehityksissä ei panosteta dokumentointiin ja usein niissä dokumentointi tehdäänkin vasta jälkeinpäin jos ollenkaan. [24.]

Ehkä tunnetuimmat ketterän ohjelmistokehityksen mallit ovat Extreme Programming (XP) (kuva 7) ja Scrum (kuva 8).

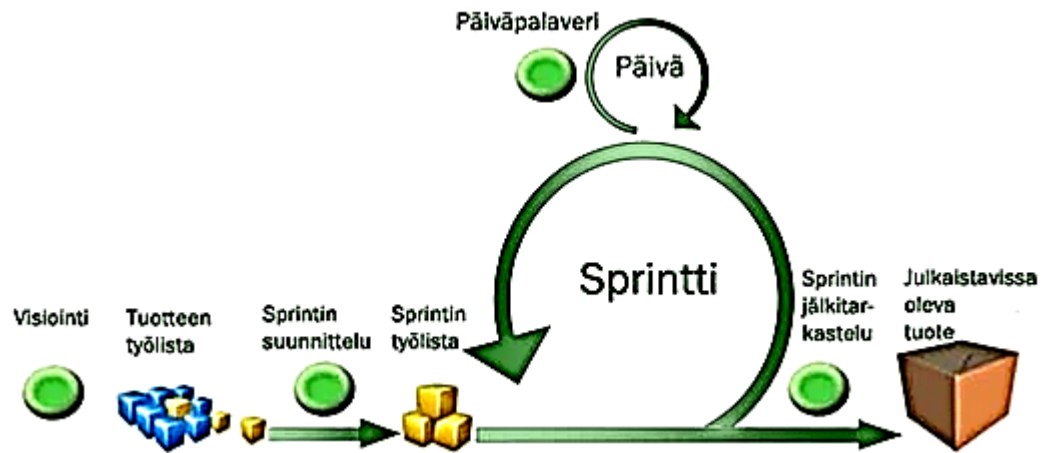
Extreme Programming hyödyntää pariohjelmointia, jossa yhdistetään molempien kokemukset ja taito. Yleensä tällä päästään parempaan tuottavuuteen ja laatuun [23].



Kuva 7. Extreme Programming-kaavakuva [14].



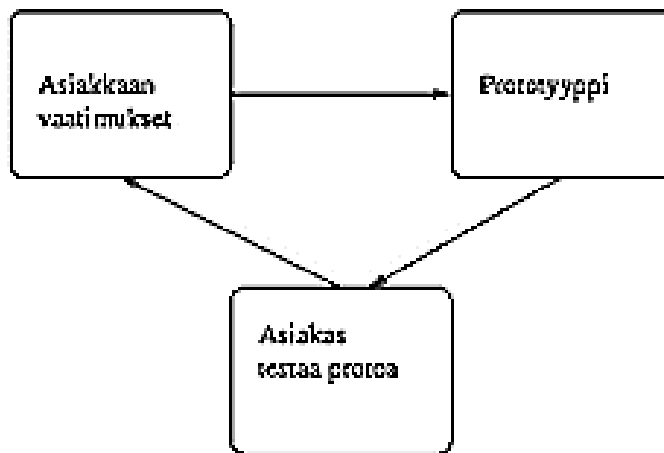
Scrum-mallissa kehitys tapahtuu toistuvien syklien mukaan, joista tärkeimmät syklit ovat sprintti ja päivä. Kehitysjakso kestää yleensä 2-4 viikkoa [10].



Kuva 8. Scrum-kaavakuva [10].

## Prototyypimalli

Prototyypimallissa (Prototype Model) tarkoituksena on tehdä yleisellä tasolla nopeasti vaatimusanalyysi ja sen perusteella prototyyppi. Tämä ensimmäinen prototyyppi pitää sisällään vain yleisen toiminnallisuuden. Prototyyppejä tulee projektin aikana kahta erilaista tyyppiä, hylättäviä ja kehitettäviä. Prototyyppiä käytetään avuksi määriteltäessä ohjelmistoon lisävaatimuksia ja prototyyppejä tehdään niin kauan, kunnes asiakas on siihen tyytyväinen (kuva 9). [7.]

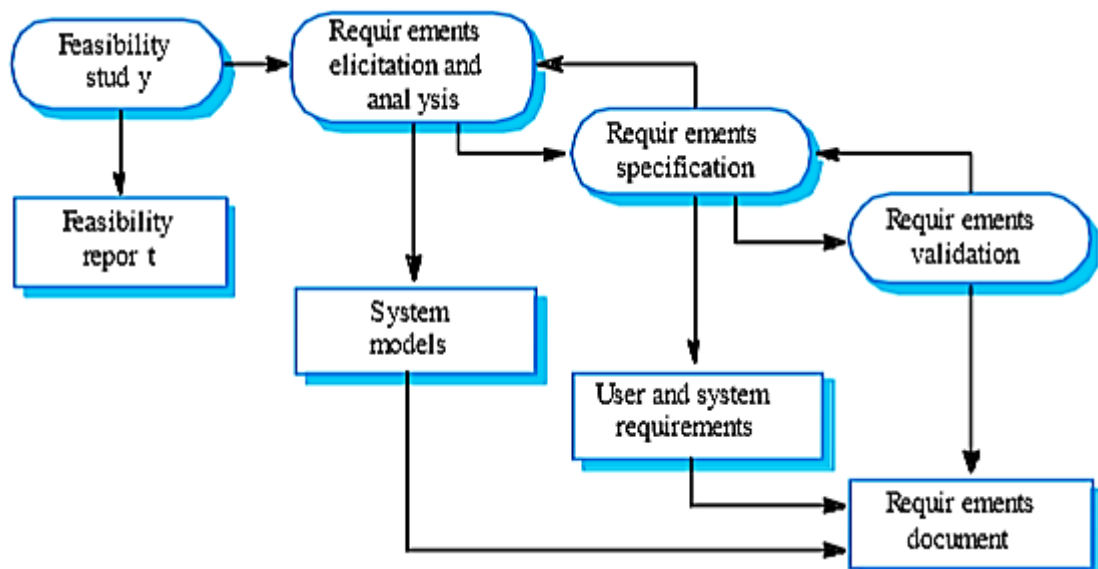


Kuva 9. Prototyypimalli [7].

Prototyypin ongelmia ovat, että järjestelmä vaatii paljon resursseja, koska järjestelmä pitää rakentaa moneen kertaan tuplatoteutuksella. Lisäksi on vaikea hahmottaa, milloin prototyyppi on valmis ohjelma. Lopputulos on usein kasattu silpuista ja huonoista ratkaisuista. [7.]

### 3.2 Esitutkimus

Esitutkimusta (feasibility study) pidetään usein osana vaatimusmäärittelyä (kuva 10), eikä omana osionaan ohjelmistokehitystä. Tämä johtuu siitä, että asiakasvaatimuksia selvitetään ja analysoidaan koko määrittelyvaiheen ajan, joten raja on häilyvä. [13.]



Kuva 10. Kuvassa esitutkimus ja vaatimusmäärittely on esitetty yhdessä [6 s. 27].

Esitutkimuksessa tuotetaan tietoa järjestelmän kehittämistä varten. Esitutkimuksen on tarkoitus vastata kysymykseen, miksi järjestelmä tulisi toteuttaa. Tutkimuksen perusteella tehdään päätös, tehdäänkö järjestelmä vai ei. Tarkoituksena on ymmärtää ongelmakenttä ja asettaa asiakasvaatimukset, mutta ei ottaa kantaa siihen, millainen järjestelmä täyttää vaatimukset. [3.]

Esitutkimus on yksi tärkeimmistä vaiheista ohjelmistokehityksessä, koska vääristä asiakasvaatimuksista ei voida päätyä hyvään järjestelmään. Asiakasvaatimusten selvittämiseen käytetään yleisimmin haastatteluita, aivoriihiä ja ideointipalavereja, markkinatutkimuksia, kyselylomakkeita sekä sidosryhmän jäsenten työn tarkkailemista. [18 s. 25.]

Esitutkimuksesta on tarkoitus saada nykyjärjestelmän kuvaus eli saada vastaukset kysymyksiin kuka ja mitä sekä missä ja milloin. Lisäksi esitutkimuksen tuotoksina on tarkoitus saada ongelmaluettelo sekä yleisellä tasolla vaatimukset tulevalle järjestelmälle. [11 s. 7.]

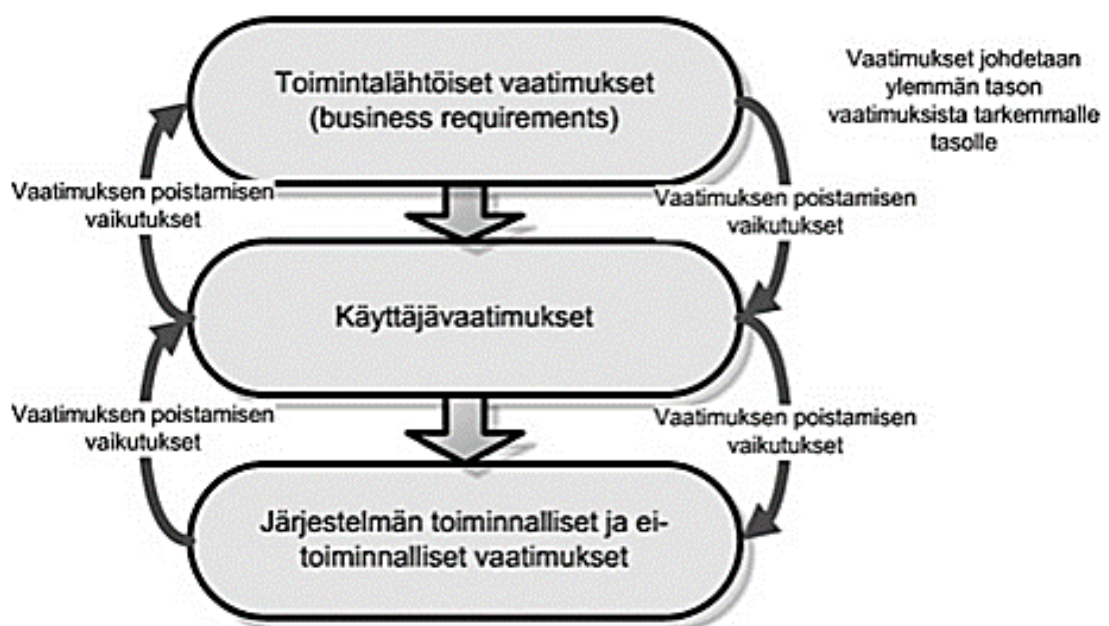
### 3.3 Vaatimusmäärittely

Vaatimusmäärittelyvaiheessa (requirements specification) kuvataan tarkasti asiakasvaatimukset sekä teknologian tuomat rajoitukset ja mahdollisuudet. Vaatimusmäärittelystä syntyy vaatimusdokumentti. Määrittelyn tulisi olla niin tarkka, että sen perusteella voitaisiin tehdä ohjelmiston kehitystyö. [28.]

Määrittelyvaiheessa asiakasvaatimukset muutetaan ohjelmistovaatimuksiksi ja tehdään alustava tekninen määrittely. Vaatimusmäärittelyssä tehdyt vaatimukset ovat yleensä epätäydellisiä, koska alussa ei osata huomioida kaikkea tarvittavaa. Lisäksi vaatimuksia väärin ymmärretään ja ne muuttuvat projektin edetessä. [15.]

Hyväksi vaatimusmäärittelyksi mielletään yksiselitteinen, täydellinen, ristiriidaton sekä todennettavissa, muutettavissa ja jäljitettävissä oleva vaatimusdokumentti. Lisäksi hyvä vaatimusmäärittely on käyttökelpoinen ylläpidon kannalta. Vaatimusmäärittelyn virheellisyys ja puutteellisuus on yksisuurimmista syistä ohjelmistoprojektin epäonnistumiseen. [13.]

Vaatimukset on tyypillisesti jaettu kolmeen ryhmään: toimintälähtöisiin vaatimuksiin, käyttäjävaatimuksiin sekä toiminnallisiin ja ei-toiminnallisiin vaatimuksiin (kuva 11).

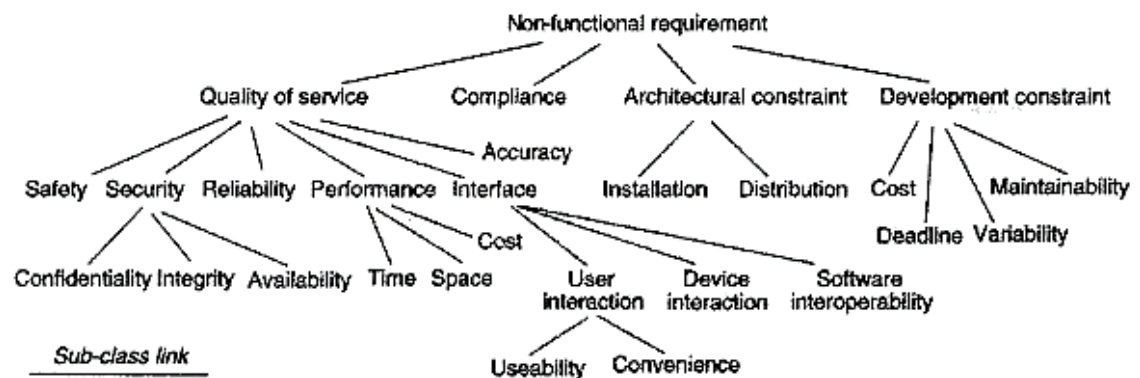


Kuva 11. Vaatimusryhmät [9].

**Toimintälähtöisillä vaatimuksilla** pyritään saavuttamaan korkean tason tavoitteita järjestelmällä. Vaatimukset perustuvat usein toimintaprosesseihin [9].

**Käyttjävaatimuksilla** kuvataan käyttäjän tekemiä toteutuksia ja/tai hyväksikäyttöjä järjestelmän avulla. Tarkoituksena on tunnistaa ja analysoida tarpeet. Käyttjävaatimukset kuvataan usein käyttötapauksina. [9.]

**Toiminnallisten ja ei-toiminnallisten** vaatimuksien määrittelyssä toiminnalliset vaatimukset vastaavat kysymykseen mitä ja ei-toiminnallisissa (kuva 12) määrittelyssä vaatimukset vastaavat kysymykseen miten [16 s. 26–27].



Kuva 12. Ei-toiminnallisten vaatimuksien luokittelu [26 s. 29].

**Ei-toiminnallisissa vaatimuksissa** on neljä pääluokkaa [26 s. 30]:

- Laatuvaatimukset (quality of service), jotka vastaavat kysymyksiin miten hyvin.
- Mukautuvuus vaatimukset (compliance), jotka kuvaavat ohjelmiston suhdetta standardeihin ja sääntöihin.
- Arkkitehtuurivaatimukset (architectural constraint), jotka kuvaavat ohjelmiston liittymistä toimintaympäristöön.
- Kehitystyön vaatimukset (development constraint), jotka kuvaavat prosesseja ja menetelmiä.

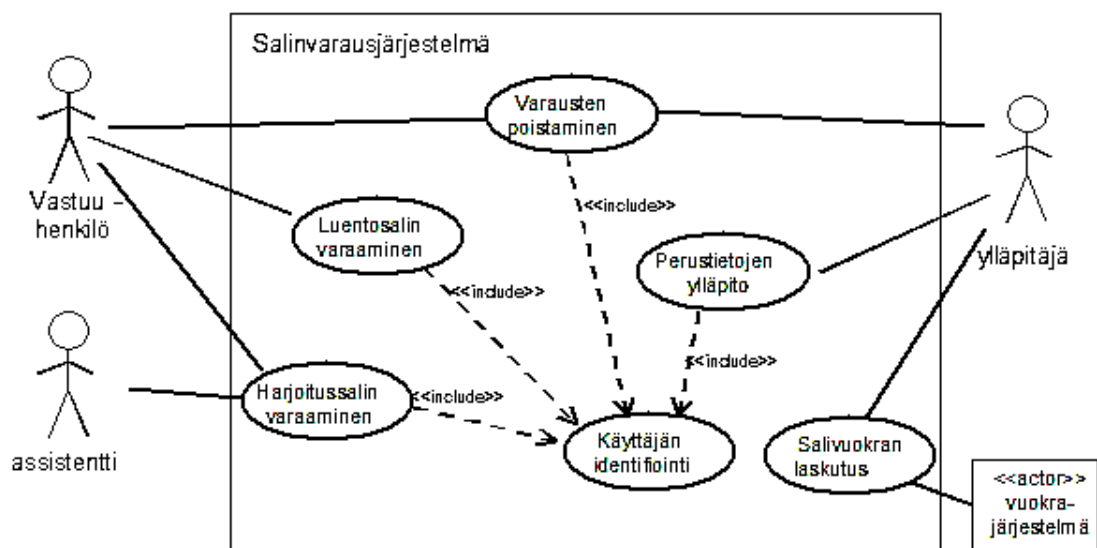
**Toiminnallisessa** määrittelyssä (functional specification) kuvataan muun muassa järjestelmän toiminnot, käyttöliittymä ja kommunikointi muiden järjestelmien kanssa. Näiden lisäksi toiminnalliseen määrittelyyn kuuluu ainakin järjestelmän yleiset tavoitteet, raportit ja järjestelmän toiminnot. Toiminnallisessa määrittelyssä ei ole tarkoitus ottaa kantaa teknisiin ratkaisuihin. [13.]

## Käyttötapauskuvaus

Käyttötapauskuvauksilla (use case model) kuvataan käyttäjän ja järjestelmän välistä vuorovaikutusta eli käyttäjän suorittamaa tehtävää. Kuvauksien esitystapaan ei ole olemassa vakiintunutta käytäntöä, mutta tavallisesti käyttötapaukset ovat tekstimuodossa (kuva 13) ja/tai käyttötapauskaavioina (kuva 14) ilmaistu. [9.]

<b>Nimi:</b>	Luentosalin varaaminen, versio 1.0 / ijh
<b>Suorittajat:</b>	Kurssin vastuhenkilö
<b>Esiehdot:</b>	Vastuhenkilö ja kurssi on syötetty järjestelmään (KT henkilötietojen ylläpito)
<b>Kuvaus:</b>	Vastuhenkilö seuraa WWW-linkkiä, joka johtaa järjestelmän pääsivulle. Hän syöttää järjestelmään käyttäjätunnuksensa ja salasanasensa (include: KT käyttäjän identifiointi). Käyttäjä pyytää järjestelmää näyttämään salin varaustilanteen haluamaltaan aikaväliltä. Hän saa eteensä salin lukujärjestysnäytön (ks. liite). Käyttäjä näkee näytöstä vapaat ajat sekä myös, mille kurseille sali on milloinkin varattu ja kuinka monelle viikolle. Käyttäjä tekee varauksen joltain vapaaksi havaitsemaltaan ajankohdalta. [Poikkeus: varaus ei onnistu].
<b>Poikkeukset:</b>	Varaus ei onnistu: Varaustilanne on voinut muuttua sillä aikaa kun varaaja tekee varausta. Järjestelmä ilmoittaa tilanteesta käyttäjälle ja käyttäjä yrittää uudelleen.
<b>Lopputulos:</b>	Varaukset kurssin luentoajoiksi on tehty.
<b>Muut vaatim.:</b>	Päivittäin käsitellään kiireisimpänäkin aikana enintään n. 100 varausta. Vastausajan on oltava alle 1 sekuntia, lukujärjestysnäytön päivitys saa kestää 5 sekuntia.

Kuva 13. Esimerkki tekstimuotoisesta käyttötapauskuvauksesta luentosalinvarausjärjestelmästä [18 s. 136].



Kuva 14. Esimerkki luentosalinvarausjärjestelmän käyttötapauskaaviosta, jossa on kolme käyttäjäroolia. Ylläpitäjä ylläpitää järjestelmän tietoja, vastuuhenkilö vastaa kurssin salivarauksista ja assistentti vastaa salivarauksista omille ryhmilleen. [18 s. 134]

Käyttötapauskuvaus alkaa aina käyttäjän aloitteesta ja loppuu virhetilanteeseen tai toivottuun lopputulokseen. Suorituspolkuja kutsutaan skenaarioiksi. Käyttötapauksen tulee olla ymmärrettävä sekä asiakkaan että suunnittelijan kannalta. [18 s. 134–136.]

Käyttötapauksella on tarkoitus liittää asiakasvaatimukset järjestelmän toimintaan ja edesauttaa vaatimusten täsmentämistä. Kuvauksilla pystytään rajaamaan järjestelmä ympäristöstä sekä määrittämään järjestelmän korkean tason toimivuus. Lisäksi käyttötapauskuvauksia voidaan käyttää iteraatioiden suunnitteluun. [5.]



### 3.4 Suunnittelu

Suunnitteluvaiheessa (design) tarkennetaan määrittelyvaiheen tuloksia ja ratkaistaan ohjelmiston toteutus sekä ohjelmiston suorittamien tehtävien suoritustapa. Tähän kuuluu ohjelmiston käyttöliittymän, tietokannan, testauksen ja käyttöönoton suunnittelu. [13.]

Järjestelmä jaetaan itsenäisiin osiin ja moduuleihin eli arkkitehtisuunnitteluksi ja tekniseksi määrittelydokumentiksi. Arkkitehtisuunnittelusta seuraa moduulisuunnittelu- vaihe, jossa suunnitellaan moduulin sisäinen rakenne. Moduuli on ohjelmasta erotettavissa oleva looginen kokonaisuus. [12.]

Ideaalitilanteessa määrittelydokumentti on niin kattava, että teknisessä suunnitteluvaiheessa tai sitä seuraavissa vaiheissa ei ole missään epäselvää siitä, miten ohjelman tulee toimia. Käytännössä tällaiseen tilanteeseen on kuitenkin mahdotonta päästä. Toisaalta määrittelydokumentti ei saa olla liian pitkä ja vaihtoehtoja poissulkeva. Suurin ongelma kuitenkin määrittelydokumenttien suhteen on, ettei niitä yleensä jakseta kirjoittaa. [19.]

Arkkitehtuurivaiheessa ei enää tehdä päätöksiä tai valintoja, millaisia ominaisuuksia ohjelmassa on. Arkkitehtuurivaiheen määrittelyssä valitaan esimerkiksi käytettävät ohjelmointikielet, komponentit kuten kirjasto, komponenttien rakenne ja hierarkia, käytettävät tietorakenteet ja sovellusrajapinnat. [19.]

### 3.5 Käyttöliittymäsuunnittelu

Käyttöliittymän pystyy periaatteessa suunnittelemana kuka tahansa. Hyvän käyttöliittymän hyvän käytettävyyden suunnittelu vaatii kuitenkin systemaattisen lähestymistavan. Suunnitteluratkaisujen tuottaminen perustuu vaatimusmäärittelyyn, yleisiin käyttöliittymän suunnitteluohjeisiin ja standardeihin, teknologian mahdollisuuksiin ja rajoituksiin sekä suunnittelijan osaamiseen, näkemyksiin ja luovuuteen. [29.]

Hyvän käytettävyyden aikaansaaminen ei synny käytettävyyssaktiviteettien suorittamisella tai sillä, että käyttäjät ovat mukana prosessissa. Sovelluksen käytettävyyden ratkaisee aina viime kädessä suunnittelijat: missä määrin käytettävyyssnäkökohdat huomioidaan suunnitteluratkaisussa. [29 s. 4.]

Monissa käytössä olevissa prosessimalleissa ohjelmiston käyttöliittymä syntyy muiden aktiviteettien sivutuotteena. Käyttöliittymä konkretisoituu vasta projektin lopussa, jolloin sitä ei voida varmistaa ennen kuin ohjelmisto on toimiva tai lähes toimiva. Vielä suurempana ongelmana on, että usein käyttöliittymä on kehittynyt toteutusnäkökohtien päälle. [30 s. 2.]

Valmiissa järjestelmässä ilmenneiden käyttöliittymän puutteiden ja ongelmien korjaaminen on kallista ja vie paljon aikaa. Erään arvion mukaan vesiputousmallissa muutostannukset kasvavat eksponentiaalisesti kymmenen potensseissa vaiheesta seuraavaan siirryttäessä. [30 s. 2-3.]

Inkrementaalisissa ohjelmistokehityksissä prosessimallit pyrkivät ratkaisemaan käyttöliittymäongelmaa liian myöhään. Jossain mielessä inkrementaalisissa ohjelmistokehitysmalleissa on jopa enemmän käyttöliittymäongelmia kuin vesiputousmallissa.

Inkrementaalisissa malleissa käyttöliittymä syntyy usein ilman kokonaiskäyttöliittymäsuunnittelua ja käyttöliittymää suunnitellaan aina jokaisen inkrementin vaiheessa. Tällä tavalla ei synny hyvää käyttöliittymää, koska samalla käyttöliittymällä on lopulta pystyttävä suorittamaan kaikki työtävät alusta loppuun. Toinen ongelma inkrementaalisesti syntyvässä käyttöliittymässä on, että lisääessä järjestelmään uusia osia, myös käyttöliittymään on lisättävä vastaavasti uusi pala. Uuden palan lisääminen käyttöliittymään muuttaa tyypillisesti myös jo olemassa olevaa käyttöliittymää. [30 s. 2-4.]

Käyttöliittymän suunnitteluprosessi alkaa valitsemalla korkean tason käyttötapauksen ja laatimalla sen suorittamiseksi tarvittavat toiminnot. Kuitenkin välttämällä lisäämstä järjestelmään mitään toimintoa tai tietoa, jota ei käyttötapauksen suorittamiseen tarvita. Suunniteltavaan järjestelmään ei voi syntyä toimintoja, joita ei tarvita käyttötapauksen suorittamiseen. [30 s. 8.]

Käytettävyyssuunnittelu tulisi aloittaa aina ajoissa, koska se tuo ryhtiä kehityshankkeeseen. Samalla tulee muistaa ottaa loppukäyttäjät mukaan suunnitteluun, jotta saadaan tietoa käyttökonteksteista, tavoitteista ja tehtävistä sekä palautetta suunnitelluista ratkaisuista. Loppukäyttäjien mukana ololla on myös ongelmansa, jos järjestelmä suunnitellaan käyttäjien toiveiden mukaan, on tuloksena usein sovellus, johon käyttäjät eivät ole tyytyväisiä. [29 s. 10–11.]

Suunnittelija tekee aina viimeiset ratkaisut ja päätökset käyttöliittymästä, ja on avain asemassa järjestelmän käytettävyyden kannalta. Prosessin lopputulokseen vaikuttaa, missä määrin suunnittelija huomioi käytettävyyssuunnittelun tuloksia ja käyttöliittymän suunnittelun yleisiä periaatteita. Ensimmäinen suunnittelu ei ole oikeastaan koskaan hyvä, vaan tarvitaan uudelleen arviointeja ja suunnittelua. Suunnittelijoiksi määritellään kaikki, jotka ovat mukana suunnittelupäätöksissä. [29 s. 11–12.]

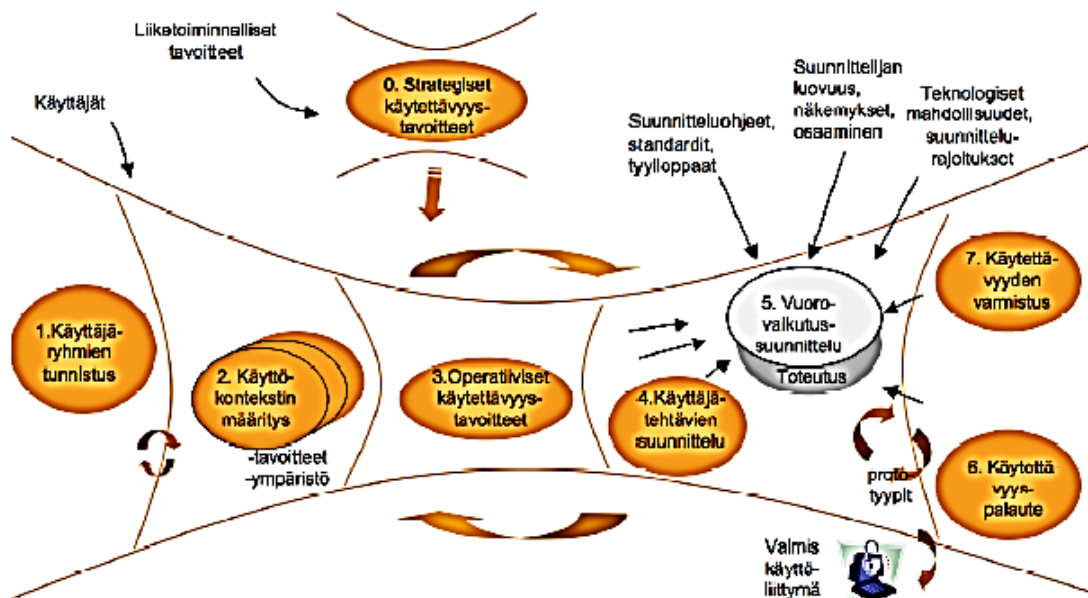
Käytettävyyssuunnitelmia ei yleisesti ole mahdollista toteuttaa siinä laajuudessa ja tarkkuudessa, miten ne kirjallisuudessa esitetään eli käytännössä menetelmiä joudutaan soveltamaan ja etsimään innovatiivisia ratkaisuja toteutukseen. Käytettävyyden suunnittelu ei siis ole mekaanista tekemistä, eikä käytettävyyden suunnittelussa ole mahdollista päästä täydellisyyteen. [29 s. 12-3.]

## JFunnel-käytettävyysohjatun vuorovaikutussuunnittelun malli

JFunnel on Suomessa kehitetty-käytettävyysohjatun vuorovaikutussuunnittelun malli (kuva 15). Mallin ajatus on, että hyvää käytettävyyttä ei voida suunnitella ilman, että tunnetaan käyttäjän työ. Lisäksi mallin ajatuksena on, että suunnittelua tulisi ohjata standardit ja yleiset suunnitteluohjeet.

Malli on jaettu kahdeksaan kohtaan ja mallin keskellä puristuksessa olevaa operatiivisten käytettävyystavoitteiden määrittämistä pidetään yhtenä keskeisimpänä kohtana. Suppilolla tässä kohdassa pyritään kuvaamaan, että operatiiviset käytettävyystavoitteet tiivistävät käyttäjädatan.

Ennen operatiivisia käytettävyystavoitteita käyttäjäryhmien tunnistamisessa ja käyttökontekstien määrittämisessä voidaan käyttää useita erilaisia mittareita, jonka takia kohdat on kuvattu laajemmalle alueelle. Samalla ajatuksella operatiivisten käytettävyystavoitteiden jälkeen potentiaali suunnitteluratkaisuille ja toteutuksille on laajempi kuin operatiivisten käytettävyystavoitteiden määrittäminen. [29 s. 4-5.]



Kuva 15. JFunnel-suppilomalli [29 s. 5].

Suppilomallin kohtien kuvaukset:

0. Kuvaa strategisten käytettävyystavotteiden määrittämistä, jossa on tarkoitus suunnitella, miten sovellus tukee liiketoimintaa. Tässä kohdan mittareina toimivat yleisesti koulutus- ja käyttäjätukiresurssit, hyväksyttävyyden, työprosesseissa saavutettavat säästöt ja käyttäjätyytyväisyys. [29 s. 6.]
1. Käyttäjärühmien tunnistus kohdassa on tarkoitus tunnistaa sovelluksen käyttäjät. Käyttäjät tulee kategorisoida sopiviin käyttäjärühmiin ja kuvata käyttäjärühmät. Kategorisoinnin yleisiä perusteita ovat esimerkiksi ikä, sukupuoli, koulutus, työrooli, sovelluksen käyttökokemus ja käyttöympäristö. [29 s. 7.]
2. Käyttökontekstin määrittäminen kohdassa analysoidaan käyttäjän työ. Yleensä analyysi tehdään käyttäjärühmittäin, koska tyypillisesti käyttäjärühmien käyttökontekstit ovat erilaisia. Määrittämisen edellytyksenä on usein keskinäinen iterointi, koska käyttäjärühmien tunnistus ja käyttökontekstin määrittäminen on järkevää tehdä yhdessä. [29 s. 6.]
3. Operatiivisten käytettävyystavotteissa tehdään suunnittelu ja asetetaan tavoitteet käytettävyydelle. Käytettävyydemäärittämisessä käytetään tyypillisesti ISO 9241-11 -protokollaa sekä sovelletaan tuloksellisuutta, tehokkuutta ja käyttäjätyytyväisyyttä. Jotta käytettävyyden tavoitteet pysyisivät kohtuullisina, pitää pystyä valitsemaan oleelliset käytettävyystavotteet kaikkien tavoitteiden joukosta. [29 s. 7-8.]
4. Käyttäjätehtävien suunnittelu -kohdassa suunnitellaan, kuinka käyttäjät suorittavat tehtävänsä kehitettävällä sovelluksella eli suunnittelukohteena on käyttäjien tehtävät. Käyttötehtävissä otetaan kantaa esimerkiksi käyttäjän ja tietokoneen väliseen työnjakoon. [29 s. 8.]
5. Vuorovaikutus suunnittelussa suunnitellaan ratkaisut muun muassa aiempien prosessien tuottamaan ohjaustietoon, huomioidaan teknologian mahdollisuudet ja toisaalta huomioidaan suunnittelurajoitukset. Tyypillistä tämän vaiheen suunnittelulle on iteraattisuus. [29 s. 9.]
6. Käytettävyyden varmistus kohdassa testataan päästäänkö operatiivisiin käytettävyyden vaatimusten tavoitteisiin eli mitataan sovelluksen käytettävyyden tasoa soveltuvilla mittareilla [29 s. 9-10].

7. Käytettävyysspalautteet kohdassa suunnitteluratkaisuille annetaan laadullista palautetta käytettävyydestä. Palautteista on tarkoitus saada vastauksia, mikä toimii ja mikä ei. Yleensä palautteen kohteena on valmis, olemassa oleva sovellus tai prototyyppi. [29 s. 10.]

### 3.6 Toteutus ja integrointi

Toteutus on ohjelmointia, jota toteutetaan siihen saakka, että saadaan ensimmäinen toimiva käännös. Ensimmäisessä käännöksessä on kuitenkin yleensä suhteellisen paljon virheitä. Loppu vaiheessa toteutusta ohjelmamoduulit integroidaan toimivaksi kokonaisuudeksi. Yleensä ohjelmoija myös yksikkötestaa oman ohjelmointinsa ennen varsinaista testaus-vaihetta. Toteutus-vaiheessa syntyy myös yksikkötestaussuunnitelma ja tekniset dokumentit. [12.]

Integroinnilla myös liitetään yhteen tai välitetään kahden eri ohjelman, sovelluksen tai järjestelmän välillä dataa. Integraatio mahdollistaa tietokantojen ympärille tehtyjen ohjelmistojen pääsyn toistensa tietokantoihin. [20.]

### 3.7 Testaus

Testauksessa tavoitteena on havaita ohjelmiston virheet ja häiriöt, jotta virheet voisi korjata ennen ohjelmiston käyttöönottoa. Testaustuloksia verrataan järjestelmän toiminnalliseen määrittelyyn. Yleensä testaukset suorittaa itsenäinen testausryhmä. [21.]

Testaus jakautuu kolmeen vaiheeseen [12]:

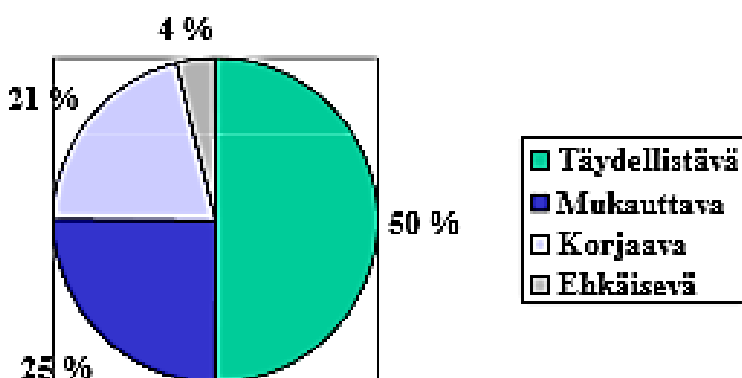
- Moduulitestaukseen, joka suunnitellaan moduulisuunnittelun yhteydessä ja jossa etsitään vikoja yksittäisistä moduuleista.
- Integrointitestaukseen, joka suunnitellaan arkkitehtuurisuunnitelman yhteydessä ja jossa testataan moduulien ja rajapintojen yhteistoimintaa.

- Järjestelmätestaukseen, joka suunnitellaan osana määrittelyä ja toteutetaan vertaamalla määrittelydokumentaatiota ja jossa testataan koko järjestelmän toimivuutta ja suorituskykyä.

### 3.8 Käyttöönotto ja ylläpito

Käyttöönotto tehdään, kun ohjelmiston ominaisuudet toimivat lähes virheettömästi ja kun kaikki kehittävät tahot ovat siihen tyytyväisiä. Käyttöönottoon liittyy asennus ja valmistelu, tietoyhteyksien valmistelu, vanhojen tietojen konvertointi ja käyttäjien perehdyttäminen. Kun uusi ohjelmisto on saatu asennettua, poistetaan vanha järjestelmä käytöstä. [4.]

Ylläpito ei enää kuulu ohjelmistotuotantoon eikä kehitykseen [4]. Ylläpito on kuitenkin tärkein näkökulma ohjelmistotyössä ja koko ohjelma on rakennettava helposti ylläpidettäväksi. Ylläpito tarkoittaa ongelmien ratkaisemista, virheiden korjaamista, uusien piirteiden lisäämistä sekä ohjelman muuttumista vaatimuksien muuttuessa (kuva 16). [12.]



Kuva 16. Ylläpitomuutosten jakauma [22 s. 3].

### 3.9 Elinkaaren loppu

Ohjelma tulee elinkaarensa päähän, kun se muuttuu tarpeettomaksi käyttöympäristön muuttuessa, tulee parempia menetelmiä, ohjelman toimintojen tarve päättyy, toimintoja integroidaan uuteen ohjelmaan tai kun käyttö ja kehittäminen lopput. [4.]

## 4 Työn taustat

Insinööri työ on tehty Neste Markkinointi Oy:n tilauksesta ja työn tarkoituksena on ollut kartoittaa nykyinen perusliikepaikka rekisterinä toimiva ohjelmisto nimeltään Nasta. Kartoituksen taustalla on tarve korvata nykyinen Nasta-järjestelmä uudella. Uudelle järjestelmälle on todellinen tarve, koska nykyinen järjestelmä on tullut elinkaarensa päähän, ja sen tuki on päättynyt. Nykyiselle järjestelmälle ei enää tehdä muuta kuin pakollinen ylläpito. Kaikki muutostoiveet ja lisäominaisuudet saavat odottaa siihen asti, että järjestelmälle saadaan korvaaja.

Järjestelmästä ei ole olemassa dokumentaatiota sen käytöstä, vaan kaikki käyttöön liittyvä tieto on käyttäjien päässä. Tämä on ICT:n näkökulmasta haastava tilanne, koska on vaikea lähteä toteuttamaan uutta järjestelmää, jos ei edes täysin tiedetä, miten nykyinen järjestelmä toimii, mitä se pitää sisällään, ketkä käyttävät, miten tietoa tallennetaan, mitkä nykyisen järjestelmän tiedoista ovat säilytettäviä sekä mitä puutteita nykyisessä järjestelmässä on. Uutta järjestelmää ei pystytä tilaamaan ennen kuin tiedetään, mitä pitää tilata.

### 4.1 Työn tarkoitus

Järjestelmän uusimisprojekti on suunniteltu käynnistettäväksi kunnolla vuonna 2016. Kokonaan uutta korvaavaa järjestelmää tuskin tulee, vaan järjestelmän toiminnallisuudet hyvinkin todennäköisesti liitetään jo olemassa olevaan järjestelmään jatkokehityksellä. Tehtävän esiselvityksen on tarkoitus tukea projektin aloitusta ja toimia esiselvityksenä nykytilanteesta.

Alkuperäisesti työn tarkoituksena ei ole ollut ottaa kantaa, minkälainen uuden järjestelmän tulisi olla, miten sen tulisi toimia tai minkä jo olemassa olevan järjestelmän yhteyteen järjestelmä tulisi rakentaa. Työ kuitenkin laajeni alkuperäisestä tarkoituksesta, koska työn aikana heräsi ajatuksia tulevaa järjestelmää kohtaan.

Nykytilannekartoitusta tehdessä syntyi ajatuksia siitä, miten tuleva järjestelmä voisi toimia. Näistä ajatuksista sekä sidosryhmien jäsenten toiveista ja tarpeista syntyi sivutuotteena korvaavalle järjestelmälle alustavaa vaatimusmäärittelyä sisältäen käyttötapauskuvauksia. Käyttötapauskuvaukset ovat oma dokumenttinsa ja käyttötapauskuvausten



lisäksi syntyi dokumentti ”Nasta-ohjelmiston haasteet ja ehdotukset”. Nasta-ohjelmiston haasteet ja ehdotukset-dokumentissa käydään läpi nykyjärjestelmän tuomia haasteita ja ehdotuksia niiden korjaamiseksi/parantamiseksi. Kyseisiin dokumentteihin palataan tarkemmin myöhemmin.

## 4.2 Työn toteutus

Ongelmakentän kartoitus suoritettiin tutustumalla nykyisin käytössä olevaan järjestelmään. Tämän lisäksi haastateltiin järjestelmän sidosryhmien pääkäyttäjiä ja peruskäyttäjiä, joiden tiedoista muodostui manuaali nykyjärjestelmään. Haastatteluilla oli tarkoitus myös selvittää käyttäjien tarpeet ja toiveet sekä saada selvitettyä, mikä on ajantasaista tietoa nykyjärjestelmässä, mitä järjestelmästä puuttuu ja mistä toiminnoista ja tietojen tallennuspaikoista on vuosien varrella tullut tarpeetonta.

Manuaalin eli nykytilanne kartoituksen tekeminen järjestelmästä on ollut ensisijaisen tärkeää, koska käytöstä ei ole ollut olemassa minkäänlaista dokumentaatiota. Dokumentaatiota on olemassa vain teknisten toiminnallisuuksien osalta. Dokumentaation puutteesta on syntynyt aika ajoin ongelmia, kun henkilöitä on jäänyt pois sidosryhmistä. Osa käyttöön liittyvistä tiedoista sekä joidenkin tietojen tallennuksien vaikutuksista muissa järjestelmissä on poistunut talosta heidän mukanaan.

### 4.2.1 Haastattelut

Nykyisen järjestelmän kartoitus alkoi haastatteluilla. Haastatteleamalla käyttäjiä eri liiketoimintaryhmistä oli tarkoituksena saada vastauksia, miten ja mihin käyttäjät liiketoimintaryhmissä käyttävät järjestelmää. Lisäksi pyrkimys oli saada tietoa käyttäjiltä, mitä hyvää ja huonoa on nykyisessä järjestelmässä sekä mitä toiveita ja tarpeita käyttäjillä olisi tulevaa järjestelmää kohtaan.

Ensimmäisen kierroksen haastattelutulokset olivat kohtuullisia ja niistä sai vain hyvän yleiskuvan, mutta tarkat käyttötapaukset jäivät vielä saamatta. Käyttäjillä oli lisäksi ensimmäisellä haastattelu kierroksella suhteellisen vähän toiveita ja tarpeita uudelle järjestelmälle. Syitä siihen, miksi ensimmäisen haastattelukierroksen tulokset olivat vain koh-

tuullisia ja suuntaa antavia, olivat ainakin oma tietämättömyys nykyisen järjestelmän kai- kista osa-alueista, kokemattomuus haastattelutilanteista ja sidosryhmille esitettyjen ky- symyksien epätarkkuudesta.

Toisella kierroksella edellisen haastattelu kierroksen tuloksia tarkennettiin täsmällisim- millä kysymyksillä. Lisäksi pääkäyttäjiltä alettiin keräämään tarkkaa kuvausta siitä, miten he tallentavat ja ylläpitävät tietoa järjestelmässä. Samalla oli pyrkimys selvittää, mitä merkityksiä tallennettavilla tiedoilla on muihin järjestelmiin ja mitkä tiedot ovat nykyisin tarpeellisia.

Toisen haastattelukierroksen aikana alkoi syntyä oikeasti nykytilannekuvausta. Samalla sidosryhmiltä alkoi saamaan tieto asioista, jotka eivät nykyjärjestelmässä toimi kovin- kaan yksinkertaisesti ja helposti. Lisäksi sidosryhmäläisiltä tuli tietoa siitä, mitä tietoja he joutuvat etsimään toisista järjestelmistä ja mitä tietoja he eivät pysty tallentamaan nykyi- seen järjestelmään ollenkaan.

Toisen haastattelukierroksen parempaan tiedon saantiin vaikutti ainakin oman parem- man valmistautumisen lisäksi aika. Ensimmäisen haastattelu kierroksen jälkeen sidos- ryhmäläiset olivat jääneet miettimään nykyjärjestelmää ja saaneet rauhassa koota aja- tuksiaan siitä. Tämä auttoi heitä tarkastelemaan tilannetta ylemmältä tasolta ja näke- mään epäkohtia paremmin.

Joidenkin sidosryhmän jäsenten osalta haastatteluista tehtiin lukuisia kertoja, mutta haas- tattelutilanteet eivät olleet enää niin virallisia. Haastattelut olivat enemmänkin lyhyitä ky- selyitä tarkentamaan ja varmentamaan jo kerättyjä tietoja.

#### 4.2.2 Tutustuminen järjestelmään

Nykyinen järjestelmä ei itsessään ollut täysin vieras työtä aloittaessani. Olin käyttänyt järjestelmää työssäni edellisenä kesänä, joten olin jo valmiiksi suhteellisen tietoinen siitä, mitä järjestelmä pitää sisällään ja kuinka sillä haetaan tietoja. Uutena järjestelmästä tuli tietojen tallentaminen ja ylläpito sekä tallennettavien tietojen vaikutukset muissa järjes- telmissä. Aiempaa tarkkaa tietoa ei ollut järjestelmän liittymärajapinta ohjelmistoista ja järjestelmistä.

Jokaisen haastatteluilla selvitetyn käyttötapauksen jälkeen oli pyrkimys toistaa tapahtuma ainakin jollain tasolla järjestelmässä itse. Toistamisen tarkoituksena oli tapahtumapolun varmentaminen omista muistiinpanoista. Joistain tallennuksiin ja ylläpitoon liittyvistä toimista ei ollut mahdollisuutta saada tietoa käyttäjiltä. Nämä tallennus- ja ylläpitoimet piti selvittää tutkimalla järjestelmää itsenäisesti.

### 4.3 Työn suorittamisen haasteet

#### 4.3.1 Haastattelut

Sidosryhmien haastattelemisen ja heidän Nastan käyttöönsä perehtyminen ei ollut täysin ongelmaton. Syitä tähän löytyy ainakin ymmärtämättömyydestä ohjelmistotekniikkaan, loma-aika, työkiireet ja rutinoituminen.

**Sidosryhmien ymmärtämättömyys ohjelmistotekniikkaan** näkyi heidän ymmärtämättömyytensä siitä, miksi kartoitusta tehdään. Osittain sidosryhmillä oli ajatusta kartoitustyön olevan turhaa ja tarpeetonta, koska uusi järjestelmään vain tilataan. Ei aina löytynyt ymmärrystä sille, että ennen kuin mitään voidaan tilata, pitää selvittää, mitä pitäisi tilata.

**Loma-aika** vaikeutti ihmisten saamista haastatteluihin. Osa haastateltavista oli oman työnsä kannalta ikävään aikaan lomalla, joka hidasti ja hankaloitti työn etenemistä. Joidenkin järjestelmän osien tutkiminen ja dokumentointi pysähtyi aika ajoin, koska käyttäjät eivät olleet tavoitettavissa.

**Työkiireet** vaikuttivat sidosryhmien haastattelujen hankaluuteen. Kun on paljon töitä tehtävänä ja loma-aika, on vaikea löytää aikaa asioiden huolelliseen läpikäymiseen. Varsinkin pääkäyttäjien osalta työkiireet vaikuttivat voimakkaasti, koska heidän kanssaan aikaa haastatteluissa tarvittiin peruskäyttäjää enemmän ja haastatteluita tarvitsi tehdä useita kertoja.

**Rutinoituminen** toi myös ison haasteen haastattelujen osalta. Moni sidosryhmäläinen on käyttänyt nykyistä järjestelmää jopa yli 20 vuotta. Tavot joilla järjestelmästä haetaan tai tallennetaan tietoa, on toistettu lukemattomia kertoja. Monia asioita käytön osalta pidetään itsestään selvyytenä eikä tämän takia muisteta kertoa kaikkea.

Vuosien kokemus käytöstä on aiheuttanut myös käyttäjien omanlaista sokeutumista nykyjärjestelmän heikkouksille. Heikkouksien näkemiseen vaikuttaa myös se, ettei kaikilla sidosryhmäläisillä ollut kunnollista vertauskohdetta.

Käyttäjät, joilla käyttökokemusta on huomattavasti vähemmän, näkivät epäkohtia ja puutteita enemmän kuin kokeneemmat käyttäjät. Heillä oli myös toiveita ja tarpeita huomattavasti enemmän verrattuna kokeneempiin käyttäjiin.

#### 4.3.2 Käyttöoikeudet

Työn suorittamisen haasteena on ollut myös käyttäjäoikeuksien puutteellisuus. Ilman administrator-oikeuksia ei ole ollut mahdollista päästä järjestelmän kaikkiin osiin, jonka seurauksena kaikkea käyttöön liittyvää ei ole ollut mahdollista selvittää. Selvittämättä jääneet osiot järjestelmästä ovat kuitenkin hyvin pieni osa kokonaisuudesta.

Käyttöoikeuksien puutteellisuuden vuoksi jäi muun muassa maayhtiöiden nykyjärjestelmän käyttö vain noin tietoisuuteen. Käyttöoikeuksillani ei päässyt näkemään maayhtiöiden järjestelmään tallentamia tietoja ollenkaan. Tähän perustuen työssä ei ole huomioitu maayhtiöiden toiveita ja tarpeita.

#### 4.3.3 Ylläpito ja päivittäminen

Järjestelmän ylläpito on jäänyt viime vuosina vähäiselle. Järjestelmästä löytyy vanhentunutta tietoa ja sellaisia ominaisuuksia, joita ei enää käytetä ja joille ei ole enää tarvetta. Järjestelmästä ei oikeastaan ole koskaan poistettu mitään, vaan sinne on vain lisätty tietoa. Koska mitään ei ole koskaan poistettu, on vaikea karsia siirrettäväksi tiedoksi vain oleellinen tieto kaiken tiedon joukosta. Toisaalta taas järjestelmästä puuttuu mahdollisuuksia tallentaa lisää erilaisia tietoja liikepaikoista. Näiden kahden välille tasapainon löytäminen on haastavaa, koska eri sidosryhmien tarpeet ja toiveet ovat hyvinkin eriäviä.

## 5 Nasta-ohjelmisto

### 5.1 Järjestelmän tarkoitus ja liiketoimintaprosessit

Nasta nimitys ohjelmistolle tulee sanoista Neste ASema daTA. Nasta toimii Nesteen markkinoinnin liikepaikkojen perustietorekisterinä, jonka päätehtävinä on toimia liikepaikkoihin liittyvän tiedon keskitettynä ylläpitopaikkana, levittää tarpeellista tietoa jatkojärjestelmiin ja toimia tietojen raportointi välineenä. Ohjelmisto on ollut käytössä jo 90-luvun alusta alkaen. Nastalla on nykyisellään noin kymmenen rajapintaohjelmistoa ja sovellusta.

### 5.2 Järjestelmän sisältämät liikepaikkatiedot

Järjestelmään tallennetaan kaikkien öljyn vähittäismyynnin asemien tiedot. Vähittäismyynnissä on liikenneasemia, automaattiasemia sekä raskaankaluston asemia. Nämä jakautuvat vielä eri tyypeihin riippuen esimerkiksi siitä, onko polttoainemyynti asemalla vähittäismyynnin omaa myyntiä vai kauppiaan myyntiä. Jokaisella asemalla on oma asemanumero, jota käytetään muun muassa liikepaikan seurantaan.

Nastan sisältämät liikepaikkatiedot ovat:

- mittarikenttätiedot
- mittaritiedot
- automaattitiedot
- säiliökenttätiedot
- säiliötiedot
- myymälätiedot
- pesutiedot
- kahvilatiedot
- palvelutiedot
- huoltotiedot
- sopimusosapuolitiedot

- sopimustiedot
- vastuuhenkilötiedot
- kuvat
- markkinointitiedot
- tietoliikennetiedot
- sovellusliittymät
- polttonestetiedot
- vuokraajan tiedot
- yrittäjätiedot
- asematiedot.

### 5.3 Järjestelmän käyttäjät

Nastaa käytetään Suomessa, Baltiassa ja Venäjällä. Muualla käytettävä versio on sama kuin Suomessa käytettävä. Nesteen Markkinoinnin lisäksi Nastaa käyttää etäyhteyksillä myös muutamat yhteistyökumppanit.

Järjestelmällä on noin 200 käyttäjätunnusta, joista noin sadalla käyttäjätunnuksella on kirjauduttu järjestelmään viimeisen vuoden aikana. Aktiivisia käyttäjiä järjestelmällä on noin 50. Suhteessa koko Neste Oyj:n henkilöstömäärään on Nastan käyttäjämäärät ovat suhteellisen pieniä. Nastan pääsääntöiset sidosryhmien jäsenet löytyvät seuraavista liiketoimintaryhmistä:

- web
- rakentaminen ja kunnossapito
- markkinointi
- kenttämyynti
- asiakaspalvelu
- laskutus ja hallinto
- liikennepalvelu Suomi-Liikepaikat

- kortti- ja maksujärjestelmät
- varastonvalvonta.

#### 5.4 Ohjelmiston nykytilanne

Ohjelmiston nykytilanteesta syntyi reilu 100 sivuinen manuaali, joka jää Neste Markkinointi Oy:n käyttöön.

#### **Manuaali**

Manuaalissa on pyritty huomioimaan ajatus, että manuaalin käyttäjällä ei olisi mitään tietoa järjestelmästä, eikä ymmärrystä tiedoista, joita järjestelmään tallennetaan. Tämän ajatuksen mukana kulkeminen koko dokumentin ajan on ollut haastavaa. Monia asioita on alkanut itsekin pitämään itsestään selvyytenä ja niiden ymmärrettävyyttä yksiselitteisinä työn aikana.

Työssä on ollut vaikea muistaa avata termejä ja selittää, miksi näin toimitaan ja mihin jonkin tiedon tallennus vaikuttaa. Tässä on auttanut paljon manuaalin luetuttaminen henkilöillä, joilla ei itsellään ole juurikaan kokemusta Nastan käytöstä. Heiltä saamien kommenttien avulla manuaalista on saatu käyttäjäystävällisempi ja käyttäjän kannalta helpommin ymmärrettävä toiminnan ja tarkoitusten osalta.

## 6 Nasta-järjestelmän haasteiden ja ehdotuksien ylätasoin tiivistelmä

### 6.1 Toiminnallisia vaatimuksia järjestelmälle

Vaikka työssä ei ole ollut tarkoitus ottaa kantaa tulevan järjestelmän toiminnallisuuteen, on työn aikana kuitenkin herännyt ajatuksia, miten asiat voisivat toimia. Näistä ajatuksista syntyi noin 50-sivuinen dokumentti ” Nasta-ohjelmiston haasteet ja ehdotukset”, joka jää Neste Markkinoinnin käyttöön.

Tässä luvussa käydään Nasta-ohjelmiston haasteet ja ehdotukset-dokumentin sisältöä tiivistettynä sekä nostamalla muutamia kohtia tarkemmin esiin. Esiin nostetut kohdat ovat valikoituneet käyttäjiltä tulleiden toiveiden ja tarpeiden pohjalta.

#### **Tulevien tarpeiden huomioiminen**

Ehdottoman tärkeää tulevaan järjestelmään on huomioida sen joustavuus, jotta järjestelmä kestäisi aikaa ja on muuntautumiskykyinen tulevaisuuden tarpeita ajatellen. Lisäksi järjestelmään pitää helposti saada uusia tallennuspaikkoja tulevillekin tiedoille. Jo nyt on tarpeita uusille tallennuspaikoille joita nykyisessä järjestelmässä ei ole, esimerkiksi PEK, PEK-purku jne.

Reaaliaikaisempi tiedon siirto on tärkeää saada myös mukaan. Tällöin esimerkiksi Neste.fi:hin saataisiin asiakkaille tieto, jos asemalla on lyhempiäkin toimintakatkoksia. Tulevan järjestelmän rajapinnat tulee olla kunnossa, koska kolmannetkin osapuolet haluavat hyödyntää asemien tietoja.

Samalla järjestelmään tulee ottaa voimakkaasti mukaan sähköiset kanavat sekä web-kanavat. Käyttöliittymän tulee olla selkeä ja helppokäyttöinen, jotta perus ATK-aidot omaava henkilö oppii järjestelmän käytön helposti ja nopeasti. Lisäksi käyttöliittymässä on huomioitava mobiililaitteet.



## **Vanhentuneen tiedon siivoaminen**

Suurimpana haasteena on karsia nykyisestä järjestelmästä vain tärkeä tieto siirrettäväksi uuteen järjestelmään. Eli pitää saada karsittua sellainen tieto pois, jolla ei ole enää tarkoitusta eikä se palvele nykyisiä käyttäjiä.

Nastalla on monta pääkäyttäjää, ja heidän näkemyksensä tärkeästä ja turhasta tiedosta ovat eriäviä. Joidenkin pääkäyttäjien mielestä mitään tietoja ei saisi poistaa ja joidenkin mielestä olisi ehdottoman tärkeää poistaa turha ja vanhentunut tieto. Siivoamalla saataisiin järjestelmään selkeyttä, ja suurin osa tavallisista käyttäjistä on haastatteluissa ker-tonut siivoustarpeesta ja tietojen epäluotettavuudesta.

Yksi pääkäyttäjä ei pysty huolehtimaan siivouksesta, vaan kaikkien tulee osallistua siivoukseen. Siivoamista vaikeuttaa se, että pääkäyttäjät eivät aina ole täysin tietoisia, mitkä tiedoista ovat turhaa/vanhentunutta, mitä tietoja ei ole enää ylläpidetty ja onko tieto luotettavaa vai suuntaa antavaa. Ja lisäksi vaikka tietoja siivotaan eikä kaikkea ole enää tarkoituksen mukaista siirtää uuteen järjestelmään, pitää kuitenkin asemien elinkaariajattelu säilyä.

## **Tietojen ylläpito vain yhdessä järjestelmässä**

Näen riskin (dataristiriita ja tehottomuus) ylläpitää sellaisia tietoja uudessa järjestelmässä jotka ovat ylläpidossa jossain muussa järjestelmässä. Jos tällaisille tiedoilla on tulevaisuudessa tarvetta, niin niille tiedoille pitäisi olla olemassa tallennuspaikat uudessa järjestelmässä, ja kyseiset tiedot pitäisi saada toisista järjestelmistä integraatioina. Päälekkäisissä tallennuspaikoissa ei mielestäni ole järkeä varsinkaan, jos tallennukset mo- neen eri järjestelmään tehdään manuaalisesti.

## **Elinkaariajattelun säilyttäminen**

Tulevan järjestelmän pitää ylläpitää asemien elinkaariajattelua eli asemien koko historia pitää olla löydettävissä ”syntymästä hautaan”. Tästä huolimatta tulee uudessa järjestelmässä olla paremmat valmiudet siivota vanhaa tietoa pois ilman, että tietojen siivoami- nen vaikuttaa lopetettujen asemien tietoihin. Esimerkiksi nykyisestä järjestelmästä siivo- tessa vanhoja laitetietoja pois, katoavat nämä tiedot myös lopetetuista asemista.

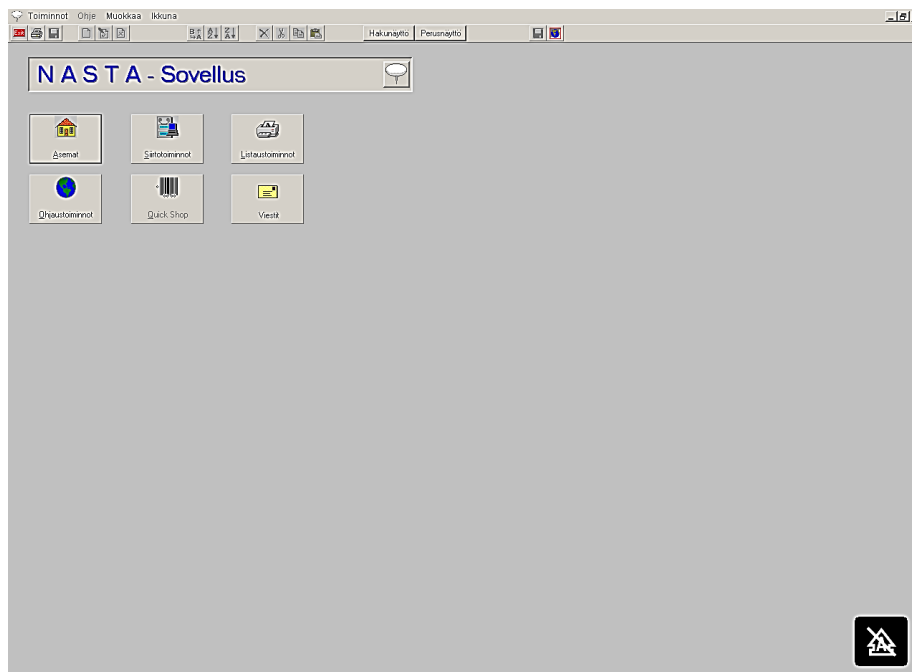
## Pääkäyttäjien roolit ja vastuut

Uudessa järjestelmässä tulee olla selkeät pääkäyttäjät, jotka vastaavat omista osa-alueistaan järjestelmässä. Tällä minimoidaan riskiä siitä, ettei uuden järjestelmän kanssa tapahdu samaa kuin Nastan kanssa. Tietoja on tallennettu järjestelmään, mutta kukaan ei ole ylläpitänyt niitä. Nastassa tietojen luottavuus ja eheys ovat kärsineet asiasta. Ei ole tarkoituksenmukaista säilöä tietoa, jolla ei ole luotettavuusarvoa.

Pääkäyttäjillä tulisi olla myöskin vastuu tietojen ylläpidosta ja vanhojen tietojen siivoamisesta. Jotta nämä saataisiin toimimaan, tulee uudessa järjestelmässä tietojen muokkaaminen ja siivoaminen saada nykyiseen Nasaan verrattuna yksinkertaisemmaksi ja helpommaksi.

## Käyttöliittymän parantaminen

Järjestelmän pitää hyödyntää koko näyttöä (kuva 17), jolloin yhdelle sivulle pystytään tuomaan huomattavasti enemmän tietoa. Tällä saadaan karsittua välilehtien määrää sekä vähennettyä moni steppisiä kulkuja tietoihin. Tiedot olisivat täten myös helpommin löydettävissä, eikä tarvitsisi hyppiä sivulta sivulle.



Kuva 17. Nasta-järjestelmän aloitusnäyttö, josta näkee, kuinka järjestelmä hyödyntää näyttöä vain vasenta yläkulmaa.

Käyttöliittymässä tulisi ottaa huomioon maailman mobilisoituminen, joten ainakin osioita järjestelmästä pitää pystyä käyttämään mobiililaitteilla. Mobiiliversion pitää olla huomattavasti kevyempi kuin kokoversion sekä toimintaa yksinkertaistettava. Lisäksi mobiiliversiona olevat painikkeet pitää olla riittävän suuret, jotta käyttäjän on mahdollista painaa niitä kosketusnäytöllisestä laitteesta.

Mobiilisovelluksella pitäisi pystyä ainakin löytämään ja päivittämään asemien perustietoja, lisäämään valokuvia arkistoon, muokkaamaan aseman tilakoodia sekä saamaan asema kartalle ja paikannuksella ajo-ohjeet asemalle.

## 6.2 Järjestelmän datan hallinta

### **Järjestelmien välillä liikkuminen**

Uudesta järjestelmästä olisi hyvä päästä siirtymään suoraan vikailmoitusjärjestelmään tekemään vikailmoituksia, sekä kontrollinauha-järjestelmään katsomaan aseman kontrollinauhaa. Toiminto palvelisi useampia käyttäjiä ja helpottaisi heidän työtään ohjelmistojen välillä.

### **Asemien ryhmittely**

Nykyisessä järjestelmässä määritellyt asematyytit pitäisi miettiä uudelleen uudessa järjestelmässä, koska ilmeisesti asematyyppi vaikuttaa siihen, miten aseman tiedot siirtyvät Neste.fi:hin. Tänne esimerkiksi kauppiaautomaatti ja liikenneasema siirtyvät lähes identtisinä, eivätkä asiakkaat ymmärrä eroa.

### **Hakemisen haasteet ja ehdotukset**

Haku paikkoja on nykyisessä järjestelmässä liian monta. Hauissa on päällekkäisyyksiä, eivätkä haut ole yksinkertaisia ja helppoja. Myös hakuparametreja ei ole tarpeeksi, jolloin käyttäjät eivät saa kaikkia tarvitsemiaan tietoja ulos järjestelmästä. Käyttäjät joutuvat tekemään paljon manuaalista työtä, jotta heidän ottamansa listaukset pitävät sisällään vain sen mitä haluttiin. Hakuihin pitää siis saada lisää parametreja, jotta hakuja saadaan täsmennettyä ja kohdistettua oikeille asemille (olettaen, että asemien tiedot saadaan kuntoon).

Asemien listaustoiminnot ja raportoinnit toimivat Nastassa eri paikasta kuin muut asemien haut. Näissä toiminnoissa on paljon samoja piirteitä, joten haut ja listaustoiminnot pitäisi yhdistää samaan hakutoimintoon selkeyttämisen vuoksi.

Asemista otetaan paljon listauksia/raportointeja eli asemien tietoja on tarvetta koostaa listauksiin. Nykyisessä järjestelmässä kaikkia listauksia ei pystytä ottamaan, eikä asemien tietoja pystytä koostamaan halutuilla tavoilla. Ongelmat johtuvat siitä, että listaustoiminnoissa ei ole hakuoptioita toimintoihin. Lisäksi listauksien otto on hyvin kankeaa ja sekavaa. Listaustoiminnoissa olevat hakuehdot eivät ole yksiselitteisiä. Otettaviin listauksiin ei myöskään ole mahdollista karsia tietomäärää valmiiksi ja/tai valita vain haluaansa tietoa. Raportteja ja listauksia joudutaan koostamaan tulostuksen jälkeen, jotta ne palvelisivat käyttäjää halutulla tavalla.

Listauksia/raportteja tulisi pystyä ottamaan kaikista asemiin liittyvistä tiedoista. Monet käyttäjät ottavat usein samoja listauksia ja käyttäjien etu olisi, jos uuteen järjestelmään pystyisi tallentamaan omia ns. suosikkeja. Listaukset/raportit pitää saada suoraan Exceliin, tulostukseen, sähköpostiin ja tallennettua lokaalisti omalle koneelle.

Asemat tulee saada näkyviin myös kartalle, ja kartalta tulisi päästä klikkaamalla aseman/asemien tietoihin.

### **Dokumenttien käsittelyn (sisältää sopimukset) haasteet ja ehdotukset**

Kaikki dokumentit pitäisi saada yhteen paikkaan, eikä niitä saisi olla ripoteltuna eri järjestelmiin. Tällöin kaikki dokumentit olisivat helposti saatavilla ja löydettävissä. Tallennuksessa käytettäviä dokumenttityyppejä tulisi lisätä tulevaan järjestelmään, jotta dokumentit olisi helpompi erotella toisistaan.

Kaikkien dokumenttien tallentamiseen pitää olla samat käsittelysäännöt ja tallentamisen pitää olla mahdollista kaikilta levyasemilta. Lisäksi tallentamisen pitää olla mahdollista myös muissa muodoissa kuin pdf-muodossa. Dokumenttien lisäämiseen on myös olemassa erilaisia käytäntöjä, mikä tekee dokumenttien tallentamisesta hankalaa sekä lisää muistin varassa toimimista.

Kun Nastaan tallennetaan sopimuksia, ei Nastassa ole valmiiksi tallennettuna tyhjiä sopimus pohjia, vaan ne pitää hakea erikseen toisesta järjestelmästä. Sopimus pohjat pitää

saada uuteen järjestelmään ja järjestelmän tulisi antaa oikea sopimus pohja automaattisesti kauppiastyypin mukaan.

Sopimusten päättymispäivän lähentyessä tulisi järjestelmän antaa automaattinen hälytys asiasta. Nykyisin ei sellaista toimintoa ole ja päättyviä sopimuksia joudutaan käymään läpi manuaalisesti.

### **Tallentamisen haasteet ja ehdotukset**

Tietojen tallennuksessa tulisi mahdollisimman paljon pystyä käyttämään täppiä ja/tai vetovalikkoja, joista valitaan tallennettavia tietoja. Tällöin nimeämiset pysyisivät yhdenmu-kaisina ja kirjoitusvirheet saataisiin minimoitua. Kirjoitettuihin tietoihin järjestelmän tulisi tehdä tarkistuksia syötteisiin ennen kuin järjestelmä hyväksyy tallennuksen (esimerkiksi koordinaatit). Tällä varmistettaisiin tietojen eheyttä ja luotettavuutta.

Monissa tietojen tallennuksissa pitää tallennuksia käydä tekemässä samaan asiaan liit-tyen monessa eri paikassa. Esimerkiksi uuden aseman perustamiseen liittyvät kohdat tulisi kaikki saada samalle sivulle, ettei tarvitsisi käydä monessa paikassa lisäämässä ja tallentamassa tietoa. Tällöin kaikki tieto tulisi tallennettua kerralla niin kuin kuuluu, eikä inhimillisiä unohduksia pääsisi tapahtumaan niin helposti.

Tallentaminen voisi myös uudessa järjestelmässä toimia siten, että tietojen lisäämisen jälkeen painetaan tallenna-nappulaa. Myös nykyisessä järjestelmässä pitää painaa Tal-leta ennen kuin tiedot tallentuvat, jolloin tiedot eivät vahingossa muutu niitä tarkastelta-essa, koska käytössä ei ole automaattista tallennusta.

### **Integraatioiden haasteet ja ehdotukset**

Integraatioiden määrää tulisi lisätä ohjelmistojen välillä molempiin suuntiin. Nykyisin ase-man perustietoja tallennetaan moneen järjestelmään manuaalisesti, mikä tuo eroavai-suuksia nimeämisiin järjestelmien välillä. Lisäksi moneen paikkaan manuaalinen tallen-taminen aiheuttaa moninkertaista työtä ja vie työaikaa muilta töiltä.

Nastassa on aikaisemmin ollut mm. volyymitietoja (myynti volyymit), ja käyttäjät toivoisi-vat tietoja takaisin. Tällaiset toiveet ja tarpeet olisi mahdollista toteuttaa integraatioilla muista järjestelmistä.

Tietojen luotettavuutta pystyttäisiin parantamaan integraatioilla. Esimerkiksi asemien mittarikenttä- ja säiliötiedot tallennetaan manuaalisesti järjestelmään, vaikka ne voisi mahdollisesti saada integraatiolla automaattisesti toimittajan rekisteristä (Taipuuko toimittajan järjestelmä liittymiin? Entä tietoturva?). Tällä parannettaisiin sekä oman järjestelmän että oman järjestelmän kautta tietojen luotettavuutta ja eheyttä.

### **Palvelutietojen ylläpidon haasteet ja ehdotukset**

Aseman palvelutietoihin sisältyy Nastasta myymälä-, pesu- ja kahvilatiedot sekä Palvelu/tuotteet ja TI/Huolto. Haaste näiden tietojen ylläpidossa ja olemassa olossa on, että tietoja ei ole ylläpidetty. Nykyiset tiedot ovat puutteellisia ja vanhentuneita, mikä vaikuttaa myös asemien hakujen vääristymään sekä puutteellisiin ja väriin tietoihin, joita siirretty Neste.fi:hin.

Tietojen ylläpito pitäisi ehdottomasti tulla kauppiailta itseltään KauppiasExtranetin kautta, koska tämä on heille teknisesti mahdollista. Kauppiaille tulisi painottaa, että tietojen ylläpito KauppiasExtranetin kautta on heille mainosta siinä, missä heidän omat nettisivunsaakin.

Kauppiaita tulisi aktivoida KauppiasExtranetin kautta tietojen tarkastamisen ja päivittämisen osalta ainakin pari kertaa vuodessa muistutuksella. Voitaisiin esimerkiksi lähettää sähköpostia asiasta ja/tai kauppiaan kirjautuessa KauppiasExtranettiin extranet voisi hälyttää asiasta. Tällä tavalla palvelutiedot saataisiin ylläpidettyä ainakin nykyistä paremmin ja asemista saataisiin enemmän/luotettavampaa tietoa siirtymään myös Neste.fi:hin. Lisäksi järjestelmästä tehtävät haut ja listaukset antaisivat parempia/oikeellisempia tuloksia, jos palvelutiedot olisivat paremmin ylläpidettyjä.

Asiakkaille tarkoitettujen palvelutietojen päivityksen ohella kauppiaiden tulisi itse ilmoittaa KauppiasExtranetin kautta omat markkinointitietonsa (tarpeet markkinointimateriaalia varten), jotta nämäkin tiedot saataisiin kuntoon ja myös pysymään kunnossa.

## 7 Käyttötapauskuvaukset

Käyttötapauskuvauksia syntyi 30 kappaletta ja niistä tuli noin 70-sivuinen dokumentti. Käyttötapauskuvaukset löytyvät tekstimuotoisina sekä käyttötapauskaavioina. Käyttötapauskuvaukset jäävät Neste Markkinointi Oy:n käyttöön.

Käyttötapauskuvaukset eivät olleet työn alkuperäisessä suunnitelmassa, mutta niitä syntyi huomaamatta samalla kun syntyi ”Nasta-ohjelmiston haasteet ja ehdotukset”-dokumentti. Kun tietoisuus nykyjärjestelmän ongelmakentistä sekä sidosryhmien toiveista ja tarpeista kasvoi, syntyi ajatuksia siitä, miten korvaava järjestelmä voisi toimia.

Käyttötapauskuvaukset eivät ole täydellisiä, koska käyttötapauskuvaukset on peilattu nykyiseen Nasta-järjestelmään. Lisäksi tässä vaiheessa ei ole vielä olemassa tarkempia ajatuksia ja suunnitelmia siitä, mihin korvaavan järjestelmän kanssa ollaan menossa.

Käyttötapauskuvauksissa on pyritty huomioimaan, miten tapahtumien kulut voisi mennä tulevassa järjestelmässä. Kuvauksissa ei myöskään ole liian tarkasti määritelty, kuinka järjestelmän tulisi toimia. Lähinnä kuvauksissa on pyritty huomioimaan, mitä erikoisvaatimuksia kussakin käyttötapauskuvauksessa on. Erikoisvaatimukset pohjautuvat sidosryhmiltä tulleisiin kommentteihin ja omiin ajatuksiin.

## Käyttötapauslista

<b>Tekeminen</b>	<b>ID</b>	<b>Nimi (nykyjärjestelmän toiminnosta)</b>	<b>Lyhyt kuvaus</b>
Haku	HA_01	Asemahaku	Asemien perushaku
Haku	HA_02	Kenttähaku	Useamman aseman haku saman aikaisesti jollakin yhteisellä parametrilla.
Haku	HA_03	Palveluhaku	Tietyn/tiettyjen palveluiden omaavien asemien haku.
Haku	HA_04	Listauksien/raportointien otto	Asemien listausta ja raportointia.
Asemasta haku	AH_01	Aseman dokumenttien haku arkistosta	Yksittäisen aseman dokumenttien haku järjestelmän arkistosta.
Asemasta haku	AH_02	Aseman säiliötietojen haku	Yksittäisen aseman säiliötietojen haku järjestelmästä.
Asemasta haku	AH_03	Aseman palvelutietojen haku	Yksittäisellä asemalla olevien palveluiden haku järjestelmästä.
Asemasta haku	AH_04	Aseman mittarikenttätietojen haku	Yksittäisen aseman mittarikenttätietojen haku.
Asemasta haku	AH_05	Aseman markkinointitietojen haku	Yksittäisen aseman sopimuksien, lupien yms. dokumenttien haku.
Perustaminen	PE_01	Aseman perustaminen	Aseman perustaminen sekä perustietojen tallentaminen järjestelmään.
Perustaminen	PE_02	Aseman perustamis- sopimuksien luonti	Asemaan liittyvien kauppiaas-/hoitajasopimusten tallentaminen järjestelmään.
Perustaminen	PE_03	Aseman henkilön perustaminen	Asemasta huolehtivan henkilön tietojen tallentaminen järjestelmään.
Perustaminen	PE_04	Aseman sijaintitietojen tallennus	Aseman sijaintitietojen tallentaminen, jotta asema saadaan näkymään kartalla.
Perustaminen	PE_05	Aseman dokumenttien tallentaminen arkistoon	Asemaa koskevien lupien, sopimuksien yms. dokumenttien tallentaminen.
Perustaminen	PE_07	Aseman automaattitietojen tallentaminen	Yksittäisen aseman automaattitietojen tallennus järjestelmään.
Perustaminen	PE_08	Aseman mittarikenttätietojen tallentaminen	Yksittäisen aseman mittarikenttätietojen tallentaminen järjestelmään.
Perustaminen	PE_09	Aseman mittaritietojen tallentaminen	Yksittäisen aseman mittaritietojen tallentaminen järjestelmään.



Perustaminen	PE_10	Aseman säiliökenttätietojen tallentaminen	Yksittäisen aseman säiliökenttätietojen tallentaminen.
Perustaminen	PE_11	Aseman säiliötietojen tallentaminen	Yksittäisen aseman säiliötietojen tallentaminen järjestelmään.
Perustaminen	PE_13	Aseman tietoliikenneyhteyksien tallentaminen	Yksittäisen aseman laitteiden tietoliikenneyhteystietojen tallentaminen järjestelmään.
Perustaminen	PE_14	Aseman palvelutietojen tallentaminen	Yksittäisellä asemalla olevien palveluiden tallentaminen järjestelmään.
Perustaminen	PE_15	Aseman tilakoodin muuttaminen	Yksittäisen aseman tilakoodin muuttaminen järjestelmään.
Perustaminen	PE_16	Asematiedotteen lähettäminen	Asematiedotteen lähettäminen järjestelmän kautta sidosryhmien jäsenille.
Perustaminen	PE_17	Aseman muuttaminen lopetetuksi	Aseman tilan muuttuessa lopetetuksi, tulee aseman tallentua järjestelmään eritavalla kuin aktiivisen aseman.
Muut	MU_01	Aseman perustiedot	Aseman perustiedoista vähintäänkin löytyvät tiedot.
Muut	MU_02	Muutosten tarkkailu	Muutosten tarkkailusta näkee mitä tietoja asemasta on muutettu, kuka on muuttanut ja milloin.
Ohjaustoiminnot	OH_01	Ohjaustoimintojen käyttö	Ohjaustoiminnoista pystytään luomaan uusia parametreja valintakenttiin.

## 8 Yhteenveto

Insinööriyössä selvitettiin Nasta-järjestelmän eli liikepaikkarekisterijärjestelmän nykytilaa. Työn tarkoituksena oli saada aikaiseksi nykyjärjestelmästä manuaali, koska sellaista ei ole ollut, vaan kaikki käyttöön liittyvä tieto on ollut pelkästään käyttäjien päässä.

Järjestelmän uusimista varten tehdyllä nykytilan selvityksellä saatiin selvitetty ongelmakentät sidosryhmien jäsenten haastatteluilla ja tutkimalla järjestelmää. Ongelmakenttien ymmärtämisellä ja kirjaamisella syntyi uudelle järjestelmälle alustavaa vaatimusmäärittelyä ja käyttötapauskuvauksia.

Dokumentissa ”Nasta-ohjelmiston haasteet ja ehdotukset” käydään läpi käyttäjiltä tulleita järjestelmässä olevia ongelmia sekä käyttäjien toiveita ja tarpeita. Lisäksi ehdotetaan ratkaisuja ja parannuksia näiden korjaamiseksi tulevassa järjestelmässä. Käyttötapauskuvauksilla haetaan mahdollista kulkua tulevassa järjestelmässä tietojen hakemiseen, ylläpitoon ja tallentamiseen.

Työ oli haasteellista tehdä, ja työtä varten olen joutunut lukemaan ja etsimään tietoa paljon. Verkkotekniikkaan suuntautuneena en ole suorittanut kursseja, jotka olisivat tukeneet työtäni. Vaikka työn suorittamisessa on ollut paljon haasteita, niin työn toteutuksen kuin ymmärtämisenkin kannalta, on työ ollut todella antoisa ja opettava.

## Lähteet

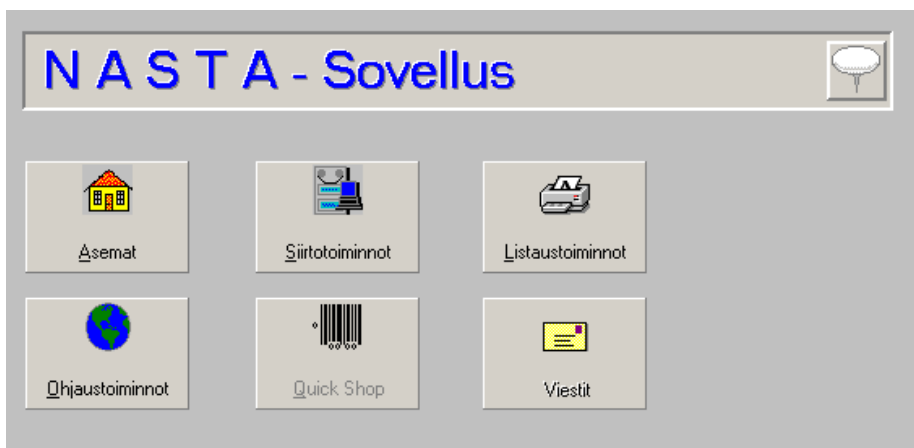
- 1 Neste Oil 2004. Verkkodokumentti. Verkkodokumentti. Saatavissa: <http://nesteoil.fi/d-fault.asp?path=35,52,107,2999,6751>. Luettu 5.5.2015.
- 2 Wikipedia Neste (yritys) 2015. Verkkodokumentti. Saatavissa: [http://fi.wikipedia.org/wiki/Neste\\_%28yritys%29](http://fi.wikipedia.org/wiki/Neste_%28yritys%29). Luettu 10.6.2015.
- 3 Oulun kauppaoppilaitos Johdatus tietojärjestelmiin. Verkkodokumentti. Saatavissa [http://www.okol.org/verkkokurssit/datanomi/tietojarjestelmien\\_kaytto\\_ja\\_kehittaminen/johdatus\\_tietojarjestelmiin/kehittamistyon\\_vaiheet\\_ja\\_elikaarimallit/kehittamistyon\\_vaiheet\\_ja\\_elikaarimallit.htm](http://www.okol.org/verkkokurssit/datanomi/tietojarjestelmien_kaytto_ja_kehittaminen/johdatus_tietojarjestelmiin/kehittamistyon_vaiheet_ja_elikaarimallit/kehittamistyon_vaiheet_ja_elikaarimallit.htm). Luettu 15.6.2015.
- 4 Wikipedia Ohjelmistotuotanto 2015. Verkkodokumentti. Saatavissa <https://fi.wikipedia.org/wiki/Ohjelmistotuotanto>. Luettu 15.6.2015.
- 5 Tampereen teknillinen yliopisto 2012. Verkkodokumentti. Saatavissa <http://www.cs.tut.fi/~otupk/kesa-2012/2012-06-07-kayttotapaukset.pdf>. Luettu 29.7.2015.
- 6 Tietojenkäsittelytieteen laitos–Helsingin yliopisto Ohjelmistotuotanto 2007. Verkkodokumentti. Saatavissa <http://www.cs.helsinki.fi/u/taina/ohtu/k-2007/pdf/Ohjelmistotuotanto6d.pdf>. Luettu 15.6.2015.
- 7 Tietojenkäsittelytieteen laitos–Helsingin yliopisto Ohjelmistotuotannon prosessimallit 2000. Verkkodokumentti. Saatavissa <http://www.cs.helsinki.fi/u/taina/ohtu/s-2000/luennot/prosessi/kaikki.html>. Luettu 15.6.2015.
- 8 Informaatioteknologia -Jyväskylän yliopisto Käytettävän käyttöliittymän suunnittelu 2005. Verkkodokumentti. Saatavissa [http://apro.mit.jyu.fi/itkp103/xhtml/08\\_itkp103\\_kw\\_luento2.ppt.html](http://apro.mit.jyu.fi/itkp103/xhtml/08_itkp103_kw_luento2.ppt.html). Luettu 15.6.2015.
- 9 Julkisen hallinnon suositukset 2012. Verkkodokumentti. Saatavissa <http://docs.jhs-suositukset.fi/jhs-suositukset/JHS173/JHS173.html#H6>. Luettu 29.7.2015.
- 10 SCRUM 2015. Verkkodokumentti. Saatavissa <http://hybridimenetelma.sun-tuubi.com/?cat=16>. Luettu 15.6.2015.
- 11 Esitutkimus. Verkkodokumentti. Saatavissa [elearn.ncp.fi/materiaali/vikevainene/Ohjelmistotuotanto/esitutkimus.ppt](http://elearn.ncp.fi/materiaali/vikevainene/Ohjelmistotuotanto/esitutkimus.ppt). Luettu 15.6.2015.
- 12 Metropolia Ammattikorkeakoulu Ohjelmointitekniikka lyhyesti 2010. Verkkodokumentti. Saatavissa <http://users.metropolia.fi/~karita/2010-3ITProg/Introduction/OhjelmistotekniikkaLyhyesti.pdf>. Luettu 15.6.2015.

- 13 Kansainvälisen viestinnän laitos/Tietojenkäsittelytieteen laitos Joensuun yliopisto Ohjelmistotuotanto 2002. Verkkodokumentti. Saatavissa [http://cs.joensuu.fi/~jimmonen/jot\\_moniste/jot\\_moniste\\_121.html](http://cs.joensuu.fi/~jimmonen/jot_moniste/jot_moniste_121.html). Luettu 15.6.2015.
- 14 Accelerated development. Verkkodokumentti. Saatavissa <http://accelerateddevelopment.ca/blog/tag/user-story/>. Luettu 27.7.2015.
- 15 Tietoyhteiskunnan kehittämiskeskus Vaatimusmäärittely 2005. Verkkodokumentti. Saatavissa <http://www.tieke.fi/pages/viewpage.action?pageId=3441242>. Luettu 15.6.2015.
- 16 Helsingin yliopisto Ohjelmiston vaatimusmäärittely 2010. Verkkodokumentti. Saatavissa <http://www.cs.helsinki.fi/u/paakki/Vaatimus-11-Luentokalvot-1.pdf>. Luettu 15.6.2015.
- 17 Wikipedia Vesiputousmalli 2013. Verkkodokumentti. Saatavissa <https://fi.wikipedia.org/wiki/Vesiputousmalli>. Luettu 2.7.2015.
- 18 Ilkka Haikala, Jukka Märijärvi. 1998 Ohjelmistotuotanto Suomen Atk-kustannus. Jyväskylä. Gummeruksen kirjapaino Oy.
- 19 Wikipedia Ohjelmistotuotanto 2015. Verkkodokumentti. Saatavissa <https://fi.wikipedia.org/wiki/Ohjelmistotuotanto#Vaatimusanalyysi>. Luettu 3.7.2015.
- 20 InfoBuild Integraatio 2015. Verkkodokumentti. Saatavissa [http://www.info-build.fi/termi\\_integraatio.php](http://www.info-build.fi/termi_integraatio.php). Luettu 3.7.2015.
- 21 Wikipedia Ohjelmiston testaaminen 2015. Verkkodokumentti. Saatavissa [https://fi.wikipedia.org/wiki/Ohjelmiston\\_testaaminen](https://fi.wikipedia.org/wiki/Ohjelmiston_testaaminen). Luettu 3.7.2015.
- 22 Tampereen teknillinen yliopisto Ohjelmistotekniikka. Verkkodokumentti. Saatavissa <http://www.cs.tut.fi/~evo/kalvot/johdanto6.pdf>. Luettu 3.7.2015.
- 23 Wikipedia 2015. Verkkodokumentti. Saatavissa [https://fi.wikipedia.org/wiki/Extreme\\_Programming](https://fi.wikipedia.org/wiki/Extreme_Programming). Luettu 27.7.2015.
- 24 Wikipedia 2015. Verkkodokumentti. Saatavissa [https://fi.wikipedia.org/wiki/Ketter%C3%A4\\_ohjelmistokehitys](https://fi.wikipedia.org/wiki/Ketter%C3%A4_ohjelmistokehitys). Luettu 27.7.2015.
- 25 Helsingin yliopisto 2005. Verkkodokumentti. Saatavissa <http://www.cs.helsinki.fi/u/taina/ohtu/k-2005/luku2.pdf>. Luettu 27.7.2015.
- 26 Helsingin yliopisto TKTL 2013. Verkkodokumentti. Saatavissa [http://www.cs.helsinki.fi/u/laine/vaati/vaati\\_s13\\_02.pdf](http://www.cs.helsinki.fi/u/laine/vaati/vaati_s13_02.pdf). Luettu 7.7.2015.

- 27 Helsingin yliopisto 2003. Verkkodokumentti. Saatavissa <http://www.cs.helsinki.fi/u/paakki/ohruk03-luento2.pdf>. Luettu 27.7.2015,
- 28 Jyväskylän yliopisto 2007. Verkkodokumentti. Saatavissa <http://users.jyu.fi/~kolli/OHTU2007/materiaali/vaatimukset.pdf>. Luettu 7.7.2015.
- 29 Joticon Oy JFunnel-malli. Verkkodokumentti. Saatavissa [http://www.joticon.fi/white\\_paper\\_JFunnel.pdf](http://www.joticon.fi/white_paper_JFunnel.pdf). Luettu 17.7.2015.
- 30 Helsingin yliopisto Tietojenkäsittelylaitos. Verkkodokumentti. Saatavissa <http://www.cs.helsinki.fi/u/salaakso/papers/GUIDe-suomeksi.pdf>. Luettu 17.7.2015.
- 31 Neste Oil 2014. Verkkodokumentti. Saatavissa <https://www.neste.com/sites/default/files/674544.pdf>. Luettu 20.7.2015.
- 32 Helsingin yliopisto 2009. Verkkodokumentti. Saatavissa [http://www.cs.helsinki.fi/u/taina/opol/k-2009/pdf/luku-6\\_2.pdf](http://www.cs.helsinki.fi/u/taina/opol/k-2009/pdf/luku-6_2.pdf). Luettu 27.7.2015.
- 33 Software Business and Engineering Institute. Verkkodokumentti. Saatavissa <http://www.soberit.hut.fi/T-76.115/01-02/palautukset/groups/Confuse/lu/docs/vmalli/index.html>. Luettu 27.7.2015.

## Nykytilanne dokumentista päävalikko esittely

Nasta järjestelmän päävalikko



Sovelluksen aloitussivulta aloitetaan Nastan käyttö. Näistä on nykyisin käytössä järjestelmästä kohdat Asemat, Listaustoiminnot ja Ohjaustoiminnot.

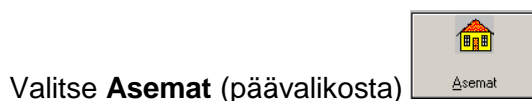
- **Asemat** kohdasta päästään hallinnoimaan ja tarkastelemaan asemien tietoja.
- **Listaustoiminnot** kohdasta päästään ottamaan listauksia ja raportteja asemista.
- **Ohjaustoiminnot** kohdasta päästään luomaan mm. uusi käyttäjä ja hallinnoimaan asetuksia.

Aloitussivulla olevat **Siirtotoiminnot** ja **Quick Shop** ovat peruja 90-luvun myynninseurannan järjestelmästä. Nämä eivät ole enää käytössä.

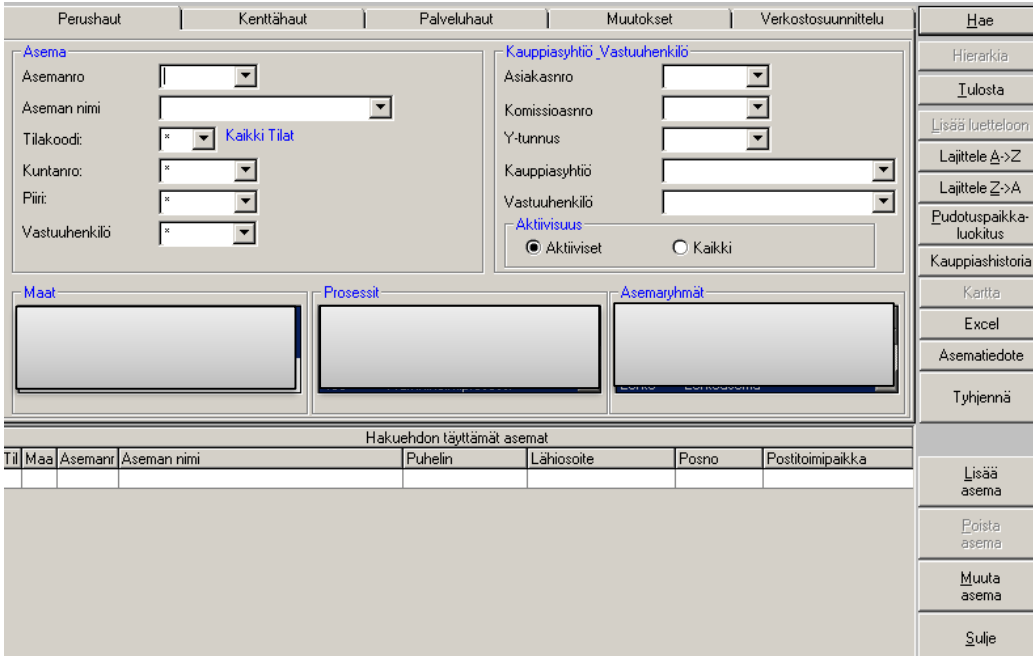
**Viestit** toiminnolla on viestitetty Nastaan liittyvistä päivityksistä ja muista muutoksista, mutta tätä ei enää käytetä.

## Nykytilanne dokumentista vapaan asemanumeron haku

Vapaa asemanumero etsitään järjestelmästä ja käytettävyys varmistetaan kirjanpitäjältä ennen kuin asemanumerolle voidaan perustaa asema.



Valitse **Perushaut** – välilehti



Tilakoodissa tulee olla **Kaikki tilat**


**Asemanumero**-kohdassa käytetään vetopainiketta, josta valitaan käyttämättömiä numeroita.

Valittu asemanumero/-numerot varmistetaan kirjanpitäjältä.

Kirjanpitäjältä tulee tieto siitä, voiko asemanumeroa käyttää vai onko kyseisellä numerolla jotain kesken.

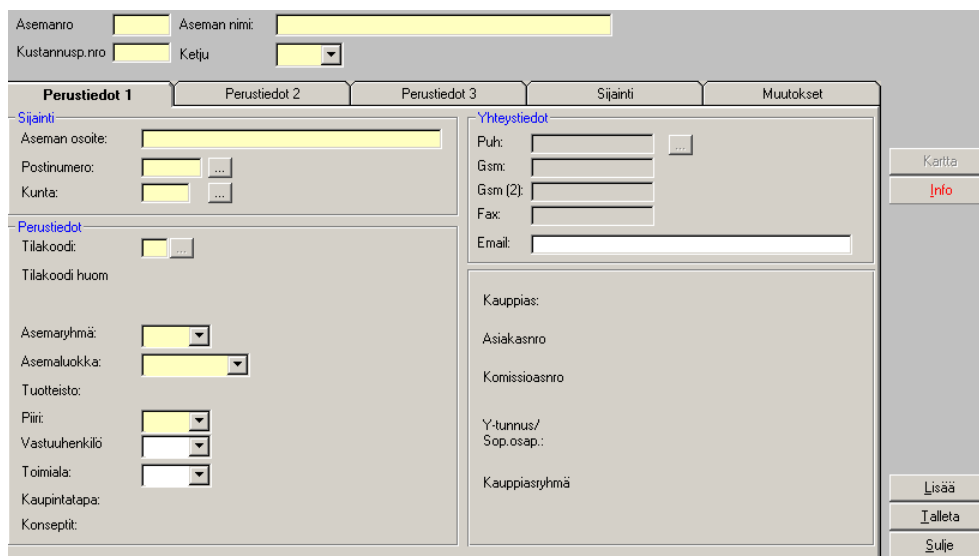
## Nykytilanne dokumentista aseman perustaminen

Aseman perustamisesta alkaa koko asematietojen hallinta ja käyttö. Operaatiolla saadaan asema tallennettua järjestelmään ja siihen pystytään lisäämään lisäosia.

Valitse **Asemat** (päävalikosta) 

Valitse **Perushaut**-välilehti

Valitse **Lisää asema** (tyhjä Perustiedot 1 -pohja)



Täytetään ainakin seuraavat tiedot (merkitty keltaisella) vrt. asemahaun antamat perustieto 1-välilehdeltä:

- **Asemanumero**
- **Aseman nimi**
- **Kustannuspaikkanumero** (sama kuin asemanumero)
- **Ketju** (mihin ketjuun asema kuuluu, esimerkiksi kuuluko asema Suomen vai Eestin asemaketjun.)
- **Aseman osoite**



- **Postinumero**
- **Kunta**
- **Tilakoodi** (laitetaan sen mukaan, missä tilassa asema sillä hetkellä on, esimerkiksi aktiivinen, jos asema on toiminnassa).
- **Asemaryhmä** (kertoo onko asema esimerkiksi liikenneasema vai express)
- **Asemaluokka** (kertoo onko asema esim. konseptoitu vai automaatti)
- **Piiri** (aluepäällikkötieto)
- **Vastuuhenkilö**
- **Toimiala**

Painetaan **Talleta**

**Käyttötapaus: Aseman perustaminen PE\_01**

ID	PE_01		
Nimi	Uuden aseman perustaminen		
Lyhyt kuvaus	Uuden aseman perustaminen rekisteriin tehdään kun taloushallintoon tulee tieto uudesta asemasta tai esimerkiksi aseman mittarikentän omistuksen muuttumisesta kauppiaalta yhtiölle. Tällöin mittarikentälle perustetaan uusi asemanumero.		
Normaali tapahtumien kulku	<ol style="list-style-type: none"> <li>1. Aloitetaan vapaan asemanumeron haku</li> <li>2. Varmistetaan asemanumeron käytettävyys</li> <li>3. Lisätään tarvittavat tiedot asemasta</li> <li>4. Lisätään Kauppias/ Hoitaja asemaan</li> <li>5. Päätetään tapahtuma</li> </ol>		
Käsittelysäännöt	<ul style="list-style-type: none"> <li>• Uusi asema pitää pystyä perustamaan järjestelmään helposti</li> <li>• Järjestelmään on määriteltävä pakollisia tietoja, jotka täytyy vähintäänkin täyttää uuden asemanumeron kohdalla <ul style="list-style-type: none"> <li>○ aseman perustiedot</li> </ul> </li> <li>• Asemaa perustettaessa pitää kaikkien täytettävien tietojen olla samalla sivulla.</li> <li>• Pääkäyttäjän on pystyttävä ylläpitämään oletustietoja helposti (lisäämään/muuttamaan/poistamaan)</li> </ul>		
Erikoiskäsittelysäännöt			
Alkuehto	Asema ei ole järjestelmässä.		
Lopputulos	Asema löytyy järjestelmästä.		
Vaihtoehtoinen kulku ja/tai lopputulos	<p>Asema on jo olemassa, mutta väärässä tilakoodissa. Tällöin asemaa ei tarvitse perustaa, vaan tilakoodi muuttaa oikeaksi.</p> <p>Lisättävissä tiedoissa ei ole tarvittavaa valintaparametria, jolloin tieto käydään perustamassa ohjaustoiminnoista. Tämän jälkeen jatketaan tietojen tallentamista.</p>		

Poikkeustilanteet	Aseman perustaminen epäonnistuu
Erikoisvaatimukset	Tietoja on ylläpidettävä
Jäljitettävyys	
Käyttäjät	Kaikki
Liiketoiminnan vastuuhenkilö/ (kirjaaja)	Taloushallinto
Muutoshistoria	
Katselmointi	
Tila	

### Aseman perustamisen käyttötapauskaavio

