

---

# **TIETOJÄRJESTELMÄN VAATIMUSMÄÄRITTELY IT-PALVELUTUOTANNON HALLINTAAN**

Case Oikeusrekisterikeskus



Ammattikorkeakoulun opinnäytetyö

Tietojenkäsittelyn koulutusohjelma

Visamäki, kevät 2014

Jukka Friman

---

## VISAMÄKI

Tietojenkäsittelyn koulutusohjelma

---

<b>Tekijä</b>	Jukka Friman	<b>Vuosi</b> 2014
<b>Työn nimi</b>	Tietojärjestelmän vaatimusmäärittely IT-palvelutuotannon hallintaan, case Oikeusrekisterikeskus	

---

## TIIVISTELMÄ

Tämän opinnäytetyön toimeksiantajana oli Oikeusrekisterikeskus, joka on oikeusministeriön hallinnonalalla toimiva virasto. Oikeusrekisterikeskuksen yhtenä tehtävänä on ylläpitää ja kehittää hallinnonalan tietojärjestelmiä. Opinnäytetyö on tapaustutkimus, jonka tarkoituksena oli selvittää toimeksiantajan tarpeet ja vaatimukset IT-palvelutuotannon hallintaa tukevalle tietojärjestelmälle.

Työn teoriaosuudessa tutustuttiin vaatimusmäärittelyn tavoitteisiin ja eri vaiheisiin, vaatimustenhallintaan sekä vaatimuksen erilaisiin määritelmiin ja luokitteluihin. Lisäksi kuvattiin tietojärjestelmän hankintaa ja ohjelmistotuotantoa, joihin vaatimusmäärittely oleellisena osana kuuluu sekä tarkasteltiin, mitä oleellisia eroja vaatimusmäärittelyllä on perinteisessä ja ketterässä ohjelmistokehitysmallissa. Teoriaosuuden lähdemateriaalina käytettiin sekä painettua että elektronista kirjallisuutta ja erilaisia tutkimuksia ja dokumentteja.

Käytännön osuudessa kartoitettiin Oikeusrekisterikeskuksen tarpeet järjestelmälle. Kartoitettujen tarpeiden osalta tehtiin priorisointi, jonka perusteella sovittiin tarkemman vaatimusmäärittelyn kohde ja toteutettiin itse vaatimusmäärittely. Tarpeiden ja vaatimusten kerääminen toteutettiin olemassa olevaa dokumentaatiota tutkimalla, haastatteluilla sekä työpajoilla.

Työn lopputuloksena syntyi toimeksiantajan tarpeet ja vaatimukset kuvaava vaatimusmäärittely, joka koostui vaatimusdokumentista ja sitä täydentävistä liitteistä. Lisäksi työn tuloksena tunnistettiin joitakin kehittämiskohteita toimeksiantajan omaan vaatimustenhallintaan.

**Avainsanat** Vaatimusmäärittely, vaatimustenhallinta, ohjelmistokehitys

**Sivut** 33 s. + liitteet 25 s.

VISAMÄKI

Degree Programme in Business Information Technology

---

<b>Author</b>	Jukka Friman	<b>Year</b> 2014
<b>Subject of Bachelor's thesis</b>	Requirements specification for an information system to manage IT service operation, case Legal Register Centre	

---

ABSTRACT

This thesis was commissioned by the Legal Register Centre which is an agency in the administrative sector of the Ministry of Justice. One function of the Legal Register Centre is to maintain and develop information systems for the administrative sector. The thesis is a case study whose purpose was to examine the client's needs and requirements for an information system to support management of IT service operation.

The objectives and different phases of requirements specification, requirements management and different definitions and classifications of requirements were examined. Furthermore, procurement of information systems and software engineering were described as requirements specification is a key element of both processes.


The main differences in requirements specification between the traditional and agile approaches in software engineering were also explored. Both printed and electronic literature and various studies and documents were utilized as the source material for conducting the theoretical part.

The needs of the Legal Register Centre for the system were elicited. After elicitation, the needs were prioritized to agree with the scope of the more detailed requirements specification. The elicitation of the needs and the requirements were carried out by studying the existing documentation and by interviews and workshops.

As a result of the thesis a requirements specification describing the needs and the requirements of the commissioner was produced. The specification consisted of the requirements document and its additional attachments. A set of possible improvements in managing requirements was also recognized and documented.

**Keywords** Requirements specification, requirements management, software development

**Pages** 33 p. + appendices 25 p.



# SISÄLLYS

1	JOHDANTO .....	1
2	TUTKIMUSMENETELMÄT .....	2
3	TIETOJÄRJESTELMÄN HANKINTA .....	2
3.1	Valmistelu .....	3
3.2	Toteutus .....	4
4	OHJELMISTOTUOTANTO ASIAKKAAN NÄKÖKULMASTA .....	5
4.1	Määrittely .....	5
4.2	Suunnittelu .....	6
4.3	Toteutus .....	7
4.4	Testaus .....	7
4.5	Ylläpito .....	8
4.6	Prosessimalleja .....	9
4.6.1	Perinteiset prosessimallit .....	9
4.6.2	Iteratiiviset prosessimallit .....	11
5	VAATIMUSMÄÄRITTELY .....	12
5.1	Vaatimus .....	13
5.1.1	Toiminnalliset vaatimukset .....	14
5.1.2	Ei-toiminnalliset vaatimukset .....	15
5.1.3	Rajoitteet .....	15
5.2	Kartoitus .....	16
5.3	Analysointi .....	17
5.4	Määrittely .....	18
5.5	Validointi .....	19
5.6	Vaatimusten hallinta .....	20
5.7	Vaatimusmäärittelyn prosessi .....	20
5.8	Vaatimusmäärittely ketterässä ohjelmistokehityksessä .....	21
6	CASE OIKEUSREKISTERIKESKUS .....	23
6.1	Tarve toimintaa tukevalle järjestelmälle .....	24
6.2	Tarpeiden tunnistaminen ja priorisointi .....	25
6.3	Vaatimusten kartoitus .....	26
6.4	Vaatimusten analysointi ja määrittely .....	27
7	KEHITTÄMISEHDOTUKSIA VAATIMUSTEN HALLINTAAN .....	29
8	YHTEENVETO .....	30
	LÄHTEET .....	31

---

Liite 1	Vaatimusdokumentti
Liite 2	Vaatimustaulukko
Liite 3	Raportointi

## 1 JOHDANTO

Julkisen hallinnon tietohallinnon neuvottelukunta, myöhemmin JUHTA, on viitannut vaatimusmäärittelyä koskevassa suosituksessaan tutkimuksiin, joiden mukaan jopa 75 % ohjelmistoprojekteista epäonnistuu puutteellisen vaatimusmäärittelyn takia. Don Reinertsen taas kertoo luvun olevan 80 – 85 %. On siis selvää, että vaatimusmäärittelyn onnistumisella on merkittävä vaikutus ohjelmistoprojektin onnistumiselle. (JUHTA 2009b, 9; Leffingwell 2011, xxiii.)

Tämän opinnäytetyön tarkoituksena on tuottaa vaatimusmäärittely Oikeusrekisterikeskuksen IT-palvelutuotannon hallintaa tukevalle tietojärjestelmälle. Oikeusrekisterikeskus on oikeusministeriön hallinnonalalla toimiva virasto, jossa työskentelee noin 120 henkilöä. Vuonna 2013 oikeusministeriön hallinnonalan IT-palvelutuotantoa organisoitiin uudelleen, ja tämän myötä Oikeusrekisterikeskuksen yhdeksi tehtäväksi tuli huolehtia oikeusministeriön hallinnonalan tietojärjestelmien ylläpidosta ja kehittämisestä. Oikeusrekisterikeskuksen IT-toiminta perustuu asiakkaiden kanssa tehtäviin palvelusopimuksiin, joilla sovitaan tietojärjestelmiin liittyvistä tehtävistä. Sopimusvelvoitteiden täyttymisen seuraamista ja niihin liittyvien tehtävien priorisoimista helpottamaan on tunnistettu tarve uudelle tietojärjestelmälle.

Työn alkuosassa paneudutaan vaatimusmäärittelyn teoriaan perehtyen muun muassa määrittelyn tavoitteeseen ja vaiheisiin sekä käydään läpi tietojärjestelmän hankinnan kulkua ja ohjelmistotuotannon menetelmiä hahmottamaan vaatimusmäärittelyn roolia prosesseissa. Lähdemateriaalina työssä käytetään sekä painettua että elektronista, ohjelmistotuotannon menetelmiä ja vaatimustenhallintaa käsittelevää kirjallisuutta, aiempia tutkimuksia sekä standardeja ja suosituksia. Tarpeiden ja vaatimusten keräämiseksi tutkitaan olemassa olevaa dokumentaatiota sekä pidetään haastatteluja ja työpajoja.

Opinnäytetyön tutkimuskysymykset ovat:

1. Miten vaatimusmäärittely toteutetaan ketterässä ohjelmistokehityksessä?
2. Mitkä ovat tulevan tietojärjestelmän vaatimukset?
3. Miten Oikeusrekisterikeskuksen vaatimusten hallintaa voitaisiin kehittää?

Opinnäytetyön tietoperusta toimii myös pohjana Oikeusrekisterikeskuksen oman vaatimusmäärittelymenetelmän kehittämiselle.

## 2 TUTKIMUSMENETELMÄT

Opinnäytetyön tutkimusmenetelmänä käytetään tapaustutkimusta, koska tapaustutkimus soveltuu hyvin käytännönläheisiin tilanteisiin. Tietojärjestelmälle asetettavien tarpeiden ja vaatimusten kartoittaminen on yhdessä tulevan järjestelmän sidosryhmien kanssa tehtävää iteratiivista työtä. Tapaustutkimusta käytetäänkin tyypillisesti kehittämissuunnitelmissa muun muassa esitutkimuksen menetelmänä (Eriksson & Koistinen 2005, 3).

Tapaustutkimuksen tarkoituksena on keskittyä yhteen tai useampaan ennalta määriteltävään tapaukseen ja tutkia intensiivisesti kohdetta. Intensiivisenä tutkimusmenetelmänä tapaustutkimus soveltuu hyvin muun muassa organisaatioiden ja prosessien tutkimiseen. Koska tapaustutkimuksessa keskitytään tiettyyn kiinnostavaan kohteeseen, ei sillä näin ollen missään nimessä pyritä tutkimustulosten yleistettävyyteen. (Virtuaaliammattikorkeakoulu 2007.)

Tapaustutkimuksissa käytetään monia erilaisia aineistolähteitä ja aineiston keruumenetelmiä ja yhdistellään niiden tuloksia. Vaikka tapaustutkimus luokitellaan laadulliseksi tutkimukseksi, voidaan tapaustutkimuksen menetelminä käyttää myös määrällisiä menetelmiä kuten kyselyjä. Tyypillisiä aineiston keruumenetelmiä ovat esimerkiksi sekä avoimet että strukturoidut haastattelut, kyselyt, havainnointi, erilaiset työpajat sekä olemassa olevan dokumentaation, kuten muistioiden tai pöytä- ja päiväkirjojen tutkiminen. (Eriksson & Koistinen 2005, 27–28.)

Yinin (1989) mukaan monien lähteiden käyttö onkin yksi kolmesta tapaustutkimuksen periaatteesta. Näin voidaan paremmin selvittää ristiriitaiset ja jopa vastakkaiset väitteet. Kaksi muuta tapaustutkimuksen periaatetta on Yinin mukaan tutkimuksen tietokanta sekä perusteluketjujen ylläpito tutkimuksen luotettavuuden lisäämiseksi. (Järvinen & Järvinen 1995, 56.)

## 3 TIETOJÄRJESTELMÄN HANKINTA

Tietojärjestelmähankkeiden epäonnistumisista on puhuttu jo pitkään, ja esimerkiksi Standish Groupin vuodesta 1994 julkaiseman CHAOS raportin tulokset ovat murskaavia. Vuoden 1994 raportin mukaan vain 16 % tietojärjestelmähankkeista onnistui, ja vuoden 2013 raportin mukaan luku oli 39 %. Samaa kertoo myös keväällä 2013 julkaistu suomalainen hankintatutkimus, Tietojärjestelmien hankinta Suomessa 2013, jonka mukaan edelleen alle puolet tietojärjestelmähankkeista katsotaan onnistuneeksi. Onnistumisen kriteereistä tärkeimpinä pidettiin aikataulun ja budjetin pitämistä sekä liiketoiminnallisen arvon toteutumista. (Forselius 2013, 14; Tietojärjestelmien hankinta Suomessa 2013, 21–22; Standish Group 1995, 3; Standish Group 2013, 1.)

Jokaisella organisaatiolla on syynsä tietojärjestelmien käyttöön. Syitä on erilaisia, ja esimerkiksi Pohjonen (2002, 10) on jakanut nämä tarpeet perus-, liike- ja operatiivisten toimintojen tukemiseen, päätöksenteon tukemiseen sekä strategisen kilpailuedun saavuttamiseen. Toisinaan tarve tietojärjestelmän hankinnalle tulee sidosryhmien tai esimerkiksi lainsäädännön vaatimuksista. Myös tietojärjestelmän hankinnan tulee aina lähteä toiminnan tarpeista. Tietojärjestelmän hankinta onkin yleensä osa suurempaa kokonaisuutta, johon liittyy myös toiminnan kehittämishankkeita. (Forselius 2013, 19–27.)

JUHTA on kuvannut ICT-palveluiden kehittämistä koskevassa suositussarjassaan palveluiden kehittämisen lähtevän liikkeelle organisaation kokonaisarkkitehtuurin kuvaamisella, ja sitä kautta tunnistettavien kehittämiskohteiden viemisellä organisaation toiminta- ja taloussuunnitelmaan. Kokonaisarkkitehtuurilähtöisessä kehittämisessä on kyse organisaation normaalista kehittämis-toiminnasta, jota ohjataan arkkitehtuurinäkökulmasta. Kokonaisarkkitehtuuria ei siis tule nähdä omana kehittämisalueenaan, vaan menetelmänä, jolla tuetaan organisaation kehittämistoimintaa. (JUHTA 2011a, 5; JUHTA 2011b, 2.)

Tietojärjestelmän määritelmänä käytetään yleensä yhden tai useamman ohjelmiston, tietovarastojen, laitteiden ja palveluiden muodostamaa kokonaisuutta, jonka tarkoitus on helpottaa tai mahdollistaa jotakin toimintaa. Joidenkin määritelmien mukaan tietojärjestelmä käsittää myös organisationaaliset, sosiaaliset ja inhimilliset ulottuvuudet, kuten ohjelmistoja käyttävät ihmiset. Tietojärjestelmän hankinnalla tarkoitetaan prosessia, joka johtaa hankinnan valmistelusta järjestelmän käyttöönottoon asti. Prosessi voidaan jakaa karkeasti kahteen osioon: hankinnan valmistelu ja hankinnan toteutus. (Forselius 2013, 115; Pohjonen 2002, 5–6.)

### 3.1 Valmistelu

Hankinnan valmisteluvaiheen lähtökohtana on toiminnan tarve ja siihen perustuva alustava investointipäätös. Tietojärjestelmän hankintaan lähdetessä organisaation strategiset tavoitteet sekä toiminnan kehittämistä varten hankittavan tietojärjestelmän ylätasoinen tavoitteet tulisi olla selvillä. Mikäli kaikkia hankinnan aloittamiseksi tarvittavia lähtötietoja ei valmisteluvaiheessa ole vielä käytettävissä, täydennetään tässä vaiheessa myös puuttuvat lähtötiedot. Valmisteluvaiheen aikana muun muassa tarkennetaan toiminnan tarvetta yksityiskohtaisemmiksi tarpeiksi ja vaatimuksiksi, varmistetaan hankinnan yhteys organisaation strategiaan ja arkkitehtuuriin sekä haluttuun toiminnan muutokseen, tunnistetaan hankinnan sidosryhmät sekä suunnitellaan hankinnan organisointia, läpivientiä ja aikataulua. Valmisteluvaiheen keskeisimmät tuotokset ovat vaatimusmäärittely, arkkitehtuurimäärittelyt, investointilaskelma sekä hankintapäätös. (Forselius 2013, 11–27; TIEKE 2005.)

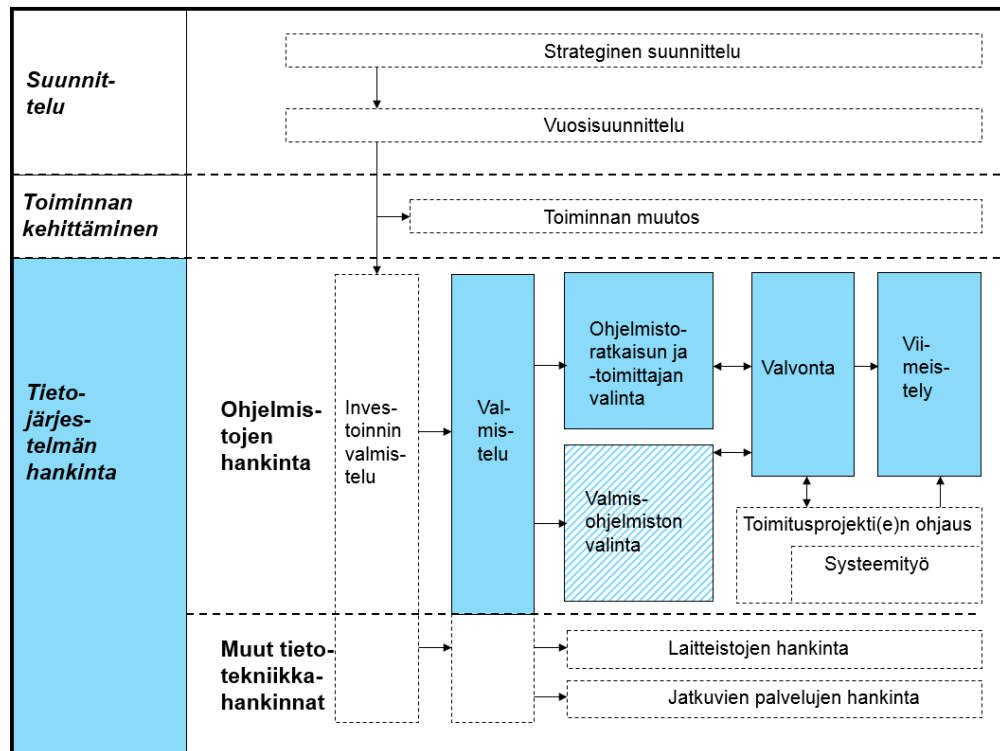
JUHTA on jakanut hankinnan valmisteluvaiheen kahteen pääosaan: esiselvitykseen ja vaatimusmäärittelyyn. Esiselvityksessä on tarkoitus selvittää tietojärjestelmän hankinnan edellytykset sekä kerätä tietoa tietojärjestelmän han-



kinnasta päättävälle tahoille. Tässä vaiheessa tarkennetaan muun muassa sitä, miksi järjestelmää ylipäänsä ollaan hankkimassa ja mitkä ovat keskeisimmät tarpeet, jotka järjestelmällä tulisi täyttää. Lisäksi selvitetään, onko hankinta tai ohjelmiston tuottaminen ylipäänsä mahdollista ja onko se järkevää. Vaatimusmäärittelyvaiheessa aiemmin tunnistettujen tarpeiden ja korkean tason vaatimusten perusteella kartoitetaan järjestelmälle selkeä ja kattava vaatimusmäärittely, joka kuvaa hankittavan tietojärjestelmän toiminnallisuuden. Vaatimusmäärittelydokumentti on tyyppillisesti hankintaan liittyvän tarjouspyynnön liite, joka toimii tilaajan ja toimittajan välisen kommunikoinnin perustana. (JUHTA 2009a, 3; JUHTA 2009b 10–11.)

### 3.2 Toteutus

Toteutusvaiheessa hankinta toteutetaan valmisteluvaiheessa hyväksytyin hankintasuunnitelman sekä muun tuotetun materiaalin perusteella. Tässä vaiheessa haetaan hankinnalle sopiva toimittaja, toteutetaan tietojärjestelmä ja otetaan se käyttöön sekä päätetään hankintaprosessi. Kuvassa 1 esitetty 4V-malli jakaa toteutusvaiheen kolmeen osaan: ohjelmistoratkaisun ja toimittajan valinta, valvonta ja viimeistely.



Kuva 1. Hankinnan kokonaiskuva (Forselius 2013, 20.)

Valinta-vaiheen tarkoitus on löytää hankinnalle toteuttaja. Toteuttaja voi olla joko organisaation sisäinen yksikkö tai ulkopuolinen toimittaja. Riippuen valitusta toimitustavasta vaiheessa muun muassa laaditaan tarjouspyyntö, järjestetään tarjouskilpailu parhaan toimittajan valitsemiseksi, tehdään hankintapäätös ja sopimukset sekä aloitetaan toimitusprojektin suunnittelu. Vaiheen tärkeimpänä lopputuotoksena on hankintapäätös.

Valvonta-vaiheen tarkoitus on varmistaa toimitusprojektin eteneminen ja tulosten laatu. Vaiheessa tuotetaan itse hankinnan kohde eli tietojärjestelmä siihen liittyvine dokumentaatioineen. Vaiheen päättyessä esitetään hankkeen ohjausryhmälle hankinnan päättämistä.

Viimeistely-vaiheen tarkoitus on todeta, että hankinta on kokonaisuudessaan toteutettu sekä kerätä kokemuksia tulevia hankintoja ja toiminnan jatkuvaa kehittämistä varten. (Forselius 2013, 11–12.)

## 4 OHJELMISTOTUOTANTO ASIAKKAAN NÄKÖKULMASTA

Tietojärjestelmän hankinta lähtee tavallisesti organisaation toiminnan tarpeista. Koska asiakkaalla on ohjelmiston tuottamiselle liiketoiminnallinen syy, näkee asiakas ohjelmistoprojektin yleensä toimittajaa laajempuna kokonaisuutena, jolla pyritään kehittämään organisaation toimintaa. (Haikala & Mikkonen 2011, 19; Immonen 2002.)

Haikalan ja Mikkosen (2011, 12) mukaan ohjelmistotuotannolla tarkoitetaan systemaattista menetelmää ohjelmiston tuottamiseksi, niin että tuotokset täyttävät niille asetetut kohtuulliset odotukset laadittujen kustannusarvioiden ja aikataulujen puitteissa. Ohjelmistotuotanto käsittää kaikki ohjelmistotyöhön liittyvät osa-alueet: määrittely, suunnittelu, toteutus, testaus, käyttöönotto ja ylläpito. Lisäksi ohjelmistotuotantoon liittyy tukiprosesseja, kuten projektinhallinta, laadunhallinta, muutostenhallinta, dokumentointi sekä vaatimustenhallinta.

### 4.1 Määrittely

Määrittelyvaiheen tarkoituksena on muodostaa käsitys ohjelmiston tarpeellisuudesta ja toteuttamiskelpoisuudesta, selvittää tavoitteet ja tarpeet sekä laatia ratkaisumalli. Määrittelyvaihe voidaan jakaa kahteen osaan: asiakasvaatimusten kartoittamiseen ja ohjelmistovaatimusten määrittämiseen. (Haikala & Märijärvi 2000, 66.)

Asiakasvaatimusten kartoittaminen alkaa jo esitutkimusvaiheessa, ja toisinaan sitä kutsutaankin esitutkimukseksi tai tarvekartoitukseksi. Asiakasvaatimukset kuvaavat ohjelmistolta vaadittavaa toiminnallisuutta eli sitä, mitä ohjelmiston pitää tehdä. Lisäksi asiakasvaatimuksilla voidaan kuvata rajoitteita, joiden

puitteissa ohjelmiston tulee toimia. Esitutkimusvaihe toteutetaan tyypillisesti asiakastoteutuksissa varsinaisen ohjelmistokehitysprojektin ulkopuolella, varsinkin jos ohjelmiston toteuttavaa kumppania ei ole vielä valittu. (Sommerville 2011, 83.)

Asiakasvaatimusten pohjalta voidaan määrittellä ohjelmiston toiminta ja muodostaa ohjelmistovaatimukset, joilla asiakasvaatimukset täytetään. Ohjelmistovaatimukset ovat tarkkoja kuvauksia palveluista ja toiminnallisuudesta, jotka ohjelmiston tulee tarjota, sekä rajoitteista, joiden puitteissa sen tulee toimia. Tyypillisesti yhden asiakasvaatimuksen täyttämiseksi joudutaan kuvaamaan useita ohjelmistovaatimuksia. Myös ohjelmistovaatimusten määrittely voidaan toteuttaa varsinaisen ohjelmistokehitysprojektin ulkopuolella, mutta määrittelyä joudutaan usein ainakin tarkentamaan projektin aikana. (Sommerville 2011, 83; Forselius 2013, 26.)

Määrittelyvaiheen lopputuotoksena syntyy tavallisesti toiminnalliseksi määrittelyksi kutsuttu dokumentti. Toiminnallinen määrittely sisältää sekä asiakasvaatimukset että ohjelmistovaatimukset. Määrittelyvaiheessa voidaan tuottaa myös ohjelmiston testaussuunnitelma, koska sen tulisi perustua määrittelydokumentaatioon. (Haikala & Mikkonen 2011, 62–68.)

### 4.2 Suunnittelu

Suunnitteluvaiheessa suunnitellaan, miten määrittelyvaiheessa määritellyt toiminnot toteutetaan. Suunnitteluvaihe voidaan jakaa kahteen osaan: arkkitehtuurisuunnitteluun sekä moduulisuunnitteluun. Arkkitehtuurisuunnittelussa ohjelmisto jaetaan pienemmiksi toiminnallisiksi kokonaisuuksiksi ja moduulisuunnittelussa suunnitellaan nämä pienemmät kokonaisuudet. Toisinaan suunnittelu- ja määrittelyvaiheiden eroa kuvataan niin, että määrittelyvaiheessa kerrotaan, mitä ohjelmiston pitää tehdä, ja suunnitteluvaiheessa kerrotaan, miten se toteutetaan. (Haikala & Märijärvi 2000, 28.)

Arkkitehtuurisuunnittelun avulla varmistetaan, että asiakkaan vaatimukset sekä teknologia saadaan yhdeksi toimivaksi kokonaisuudeksi. Arkkitehtuurisuunnittelussa ohjelmisto jaetaan pienemmiksi itsenäisiksi komponenteiksi, joista jokaisella voi olla myös oma sisäinen arkkitehtuurinsa. Lisäksi arkkitehtuurisuunnittelussa suunnitellaan näiden komponenttien väliset rajapinnat, niin että yksittäisen moduulin suunnittelussa ja toteutuksessa voidaan keskittyä vain kyseiseen moduuliin koko ohjelmiston sijaan. Itsenäiset ja mahdollisimman vähän toisistaan riippuvat komponentit helpottavat myös ohjelmiston ylläpitoa. (Haikala & Mikkonen 2011, 177–179.)

### 4.3 Toteutus

Toteutusvaiheessa suunnittelun mukaiset ohjelmistokomponentit, rajapinnat ja käyttöliittymät toteutetaan ja integroidaan ohjelmistoksi. Mikäli ohjelmiston rakennetta ja toiminnallisuutta koskevat ratkaisut on tehty huolellisesti jo aiemmissa vaiheissa, pitäisi itse ohjelmointivaiheen olla melko suoraviivaista. Myös ohjelmiston dokumentointi on merkittävä osa toteutusvaihetta. (Pohjonen 2002, 34; Haikala & Mikkonen 2011, 195.)

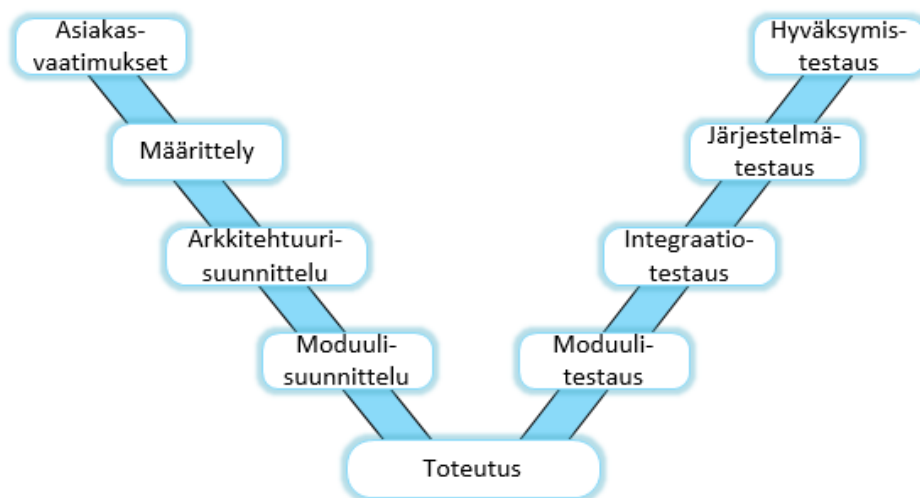
Tyypillisesti komponenttitason suunnittelu- ja toteutusvaiheet limittyvät, mutta esimerkiksi turvallisuuskriittisissä komponenteissa suunnittelu on hyvä tehdä valmiiksi ennen toteutuksen aloittamista. Toteutusvaiheessa tavallisesti tehdään myös jo moduulitestausta, jossa ohjelmistokomponentin ohjelmoija testaa jatkuvasti toteuttamaansa komponenttia, todetakseen sen toimivan suunnitellusti. (Sommerville 2011, 40–42.)

### 4.4 Testaus

Testausvaiheen aikana varmistetaan että ohjelmisto toimii, kuten sen on tarkoitettu toimivan sekä pyritään löytämään siitä virheet ennen julkaisua. Sommervillen mukaan testausvaiheella on kaksi tavoitetta: varmistaa, että ohjelmisto täyttää sille asetetut vaatimukset ja tunnistaa mahdolliset virheet tai puutteet. Näiden tavoitteiden saavuttamiseksi testausta tehdään useissa ohjelmistokehityksen vaiheissa erilaisin metodein ja erilaisin tavoittein.

Testaus voidaan jakaa kehittäjän tekemään testaukseen sekä asiakkaan tekemään testaukseen. Kehittäessä ohjelmistoa suoraan asiakkaalle, kehittäjän tekemä testaus kuuluu tyypillisesti ohjelmistotoimittajan vastuulle. Kehittäjän tekemän testauksen tarkoitus on varmistaa, että toteutettu ohjelmisto täyttää sille asetetut vaatimukset sekä etsiä ohjelmistosta virheitä. Asiakkaan tekemän hyväksymistestauksen tavoite on selvittää, täyttääkö ohjelmisto asiakkaan sille asettamat odotukset. Hyväksymistestauksen lisäksi asiakas osallistuu toisinaan myös aiempien testausvaiheiden toteutukseen. Tämä on yleistä esimerkiksi iteratiivisissa ohjelmistokehitysmalleissa, joista kerrotaan alaluvussa 4.6.2. (Sommerville 2011, 206, 210–229.)

Yksi tapa kuvata ohjelmistotestauksen vaiheita on kuvassa 2 esitetty V-malli. Mallin mukaan testaus on jaettu vähintään kolmeen eri tasoon. Määrittelyvaiheessa tuotettujen määritysten mukaisuus todetaan järjestelmätestauksella. Arkkitehtuurisuunnittelun mukaisuus varmistetaan integraatiotestauksella ja moduulisuunnittelun mukaisuus moduulitestauksella. Järjestelmätestaussuunnitelma ja testitapaukset tulisi suunnitella jo määrittelyvaiheessa, integraatiotestaus arkkitehtuurisuunnitteluvaiheessa ja moduulitestausta moduulisuunnitteluvaiheessa. (Haikala & Märijärvi 2000, 28.)



Kuva 2. V-malli

Hyväksymistestauksella varmistetaan, että toteutettu ohjelmisto täyttää sille asetetut tarpeet ja toiveet, ja että asiakas on tyytyväinen lopputulokseen. Hyväksymiskriteerit tulisi määrittellä jo hankintasopimukseen, mutta todellisuudessa tämä voi olla vaikeaa, koska vaatimukset tavallisesti tarkentuvat projektin aikana. Hyväksymistestausprosessi jatkuu, kunnes ohjelmiston toteuttaja sekä asiakas ovat yhtä mieltä siitä, että lopputulos on hyväksyttävissä. (Sommerville 2011, 42, 229.)

### 4.5 Ylläpito

Ohjelmiston elinkaaren katsotaan olevan vasta aluillaan käyttöönottovaiheessa. Ylläpitovaihe onkin ohjelmiston elinkaaren pisin yksittäinen vaihe, joka kestää ohjelmiston käyttöönotosta sen käytöstä poistamiseen asti. Ylläpitovaiheessa ratkotaan asiakkaan ongelmia, tuotetaan ohjelmistoon uusia ominaisuuksia tai muutetaan olemassa olevia ominaisuuksia sekä korjataan virheitä. (Pohjonen 2002, 37–38; Immonen 2002.)

Ylläpitovaiheessa ohjelmistoon tehtävien muutosten arviointi ja hyväksyntä hoidetaan muutoksenhallinnan avulla. Muutoksenhallintaprosessin kulku vaihtelee organisaatioittain, eikä prosessille ole yhtä oikeaa mallia. Muutospyyntöjä tai -ehdotuksia voi esiintyä esimerkiksi ohjelmistosta löytyvän virheen, toimintaympäristön tai lainsäädännön muutoksen, tai sidosryhmien uusien tarpeiden vuoksi. Ennen muutospyyntöjen hyväksymistä tulisi aina arvioida muutoksen kustannukset sekä analysoida sen vaikutukset. Hyväksytyt muutokset toteutetaan ja otetaan käyttöön jossain tulevassa ohjelmistoversiossa. (Sommerville 2011, 237–238.)

Tyypillisesti ohjelmiston ylläpitovaiheen osuus on noin 70 – 80 % koko elinkaaren kustannuksista. Ylläpitovaihetta yleisimmin hankaloittava tekijä on puutteellinen dokumentaatio. Määrittely-, suunnittelu- ja toteutusvaiheissa tehdyt ratkaisut esimerkiksi arkkitehtuuri-suunnittelussa, ylläpidettävyyden

huomioinnissa vaatimuksia määriteltäessä sekä dokumentaation laadussa ovat siis tärkeässä asemassa ylläpitovaiheen tukemisessa. (Sommerville 2011, 242–243; Pohjonen 2002, 37–38; Immonen 2002.)

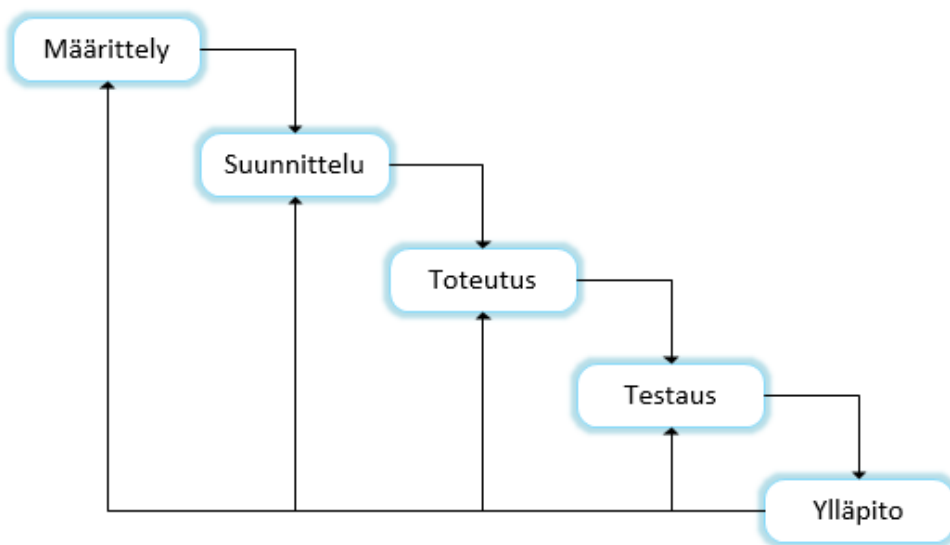
### 4.6 Prosessimalleja

Prosessilla tarkoitetaan usein toistuvia tapahtumasarjoja, joiden avulla syötteet muutetaan tuotoksiksi. Prosessi koostuu useista toisiinsa liittyvistä toiminnoista, ja ne ovat mukana jokapäiväisessä elämässämme, vaikkamme niitä välttämättä prosesseina tunnistakaan. Prosesseja kuvataan helpottamaan tapahtumasarjan toistamista samalla tavalla myöhemmin. Kun joku esimerkiksi keksii ratkaisun tiettyyn ongelmaan, voidaan ratkaisu kuvata prosessiksi, jonka toistamalla muut pystyvät ratkaisemaan saman ongelman. Esimerkiksi pesukoneen käyttöohjeessa tyypillisesti kuvataan prosessi, kuinka tulee toimia pestäkseen koneellisen pyykin. (Kotonya & Sommerville 1998, 25–26; JUHTA 2002, 2.)

Ohjelmistotuotantoa varten on kehitetty erilaisia malleja, joita kutsutaan esimerkiksi prosessimalleiksi (Sommerville 2011), ohjelmistoprojektimalleiksi (Haikala & Mikkonen 2011), elinkaarimalleiksi (Haikala & Märijärvi 2000; Pohjonen 2002) tai systeemityömalleiksi (TTL 2005). Nämä mallit eivät ole tarkkoja kuvauksia prosessista, vaan abstrakteja malleja, jotka kuvaavat erilaisia lähestymistapoja ohjelmistoprosessille. Prosessimallit voidaan jakaa suunnitelmaohjautuviin malleihin ja iteratiivisiin malleihin. Suunnitelmaohjautuvia malleja kutsutaan monesti myös perinteisiksi malleiksi. (Sommerville 2011, 29.)

#### 4.6.1 Perinteiset prosessimallit

Ensimmäinen, ja luultavasti tunnetuin ohjelmistoprosessimalli, on vuonna 1970 julkaistu vesiputousmalli. Malli on saanut nimensä vesiputousta muistuttavasta kuvauksesta lineaarisesti etenevästä prosessista. Kuvassa 3 esitetty vesiputousmalli kuvaa kaikki alaluvuissa 4.1 – 4.5 kuvatut vaiheet toisiaan seuraavina, peräkkäisinä vaiheina.



Kuva 3. Vesiputousmalli (Sommerville 2011, 30.)

Vesiputousmallin ajatus on tuottaa jokaisen vaiheen aikana dokumentaatio, joka hyväksytään ja ojennetaan seuraavalle vaiheelle. Myöhempiä vaiheita ei tulisi aloittaa ennen edellisen hyväksymistä, mutta käytännössä vaiheet kuitenkin limittyvät; suunnittelun yhteydessä huomataan puutteita määrittelyssä, toteutuksessa huomataan suunnitteluvirheitä ja niin edelleen. Perinteisiä prosessimalleja kutsutaan usein myös suunnitelmaohjatutuviksi menetelmiksi, millä viitataan nimenomaan dokumentaation suureen määrään ennen varsinaisen toteutusvaiheen aloittamista. (Sommerville 2011, 30–31.)

Toinen tunnettu suunnitelmaohjattu prosessimalli on V-malli, jonka alkuperäinen versio kuvasi samat vaiheet kuin vesiputousmalli. V-malli on esitetty kuvassa 2. V-mallissa vaiheet on kuvattu V-kirjaimen muotoon, niin että mallin risteyskohdassa olevan toteutusvaiheen vasemmalla puolella ovat sitä edeltävät vaiheet, ja oikealla puolella vastaavat testaus ja todentamisvaiheet. (Salo 2006, 19–20.)

Perinteisten prosessimallien suurimpana heikkoutena pidetään liian etupainotteista määrittelyä. Laajojen tietojärjestelmien kehittäminen, asiakasvaatimusten kartoittamisesta valmiiseen tuotteeseen, saattaa viedä vuosia, ja aikaisessa vaiheessa määritellyt vaatimukset usein muuttuvat hankkeen aikana. Koska lopputulos nähdään vasta toteutusvaiheen jälkeen, saattavat asiakkaan tarpeet jäädä täyttymättä. (Sommerville 2011, 31–32.)

### 4.6.2 Iteratiiviset prosessimallit

Suunnitelmaohjautuvien mallien haasteista johtuen ohjelmistotuotannon prosessimallien kehitys on suuntautunut enemmän vaatimusten evoluutioon tähtääviin prosessimalleihin. Iteratiivisten prosessimallien perimmäinen ajatus on pilkkoa toteutus pienempiin osiin, niin että toteutettava ohjelmisto valmistuu paloissa. Iteraatioita on kuvattu jopa omiksi mini-projekteikseen, jotka pitävät sisällään vaatimusmäärittely-, suunnittelu-, toteutus- ja testausvaiheet. Iteraation sisältö ei välttämättä ole kokonainen toimiva sovelluksen osa, joten tyyppillisesti iteraatioita ei julkaista tuotantokäyttöön. Iteratiivisia malleja kutsutaan myös iteratiivisiksi ja inkrementaaliseksi malleiksi. Inkrementaalilla kehittämisellä tarkoitetaan ohjelmiston toiminnallisuuden kasvattamista vaiheittain. Inkrementti koostuu tyyppillisesti yhdestä tai useammasta iteraatiosta, ja inkrementin valmistuttua se voidaan ottaa tuotantokäyttöön. (Salo 2006, 21.)

Iteratiivisilla malleilla on monia hyötyjä perinteisiin malleihin verrattuna. Koska ohjelmistoa toteutetaan pienemmissä paloissa, voidaan muuttuvia vaatimuksia huomioida paremmin ja nopeammin kuin perinteisissä malleissa. Asiakas pääsee myös näkemään nopeammin valmistuvan ohjelmiston toimintaa ja ehdottamaan muutoksia mahdollisiin puutteisiin. Inkrementtien suunnittelussa huomioidaan tavallisesti myös toiminnallisuuksien liiketoiminnallinen arvo eli toteutetaan ensimmäisenä sellaiset toiminnallisuudet, joista asiakas saa suurimman hyödyn. (Sommerville 2011, 33.)

Iteratiiviset prosessimallit ovat myös ketterän ohjelmistokehityksen perusta. Ketterillä menetelmillä viitataan vuonna 2001 perustetun Agile Alliancen julistuksen periaatteiden mukaiseen ohjelmistokehitykseen. Julistuksen keskeisin viesti on korostaa yksilöiden ja yhteistyön, muutoksiin vastaamisen sekä toimivan ohjelmiston merkitystä prosessien, dokumentaation, sopimusten ja suunnitelmien sijaan. (Agile Alliance 2001.)

Ketterää ohjelmistokehitystä on viime vuosina tarjottu ratkaisuksi lähes kaikkiin tietojärjestelmien kehittämisen ongelmiin. Oikein käytettynä ketteristä menetelmistä onkin saatu hyviä tuloksia, mutta toisinaan ne ovat tuottaneet myös pettymyksiä. Tunnetuimpia ketteriä menetelmiä ovat esimerkiksi eXtreme Programming eli XP ja Scrum. (Forselius 2013, 16; Sommerville 2011, 59.)

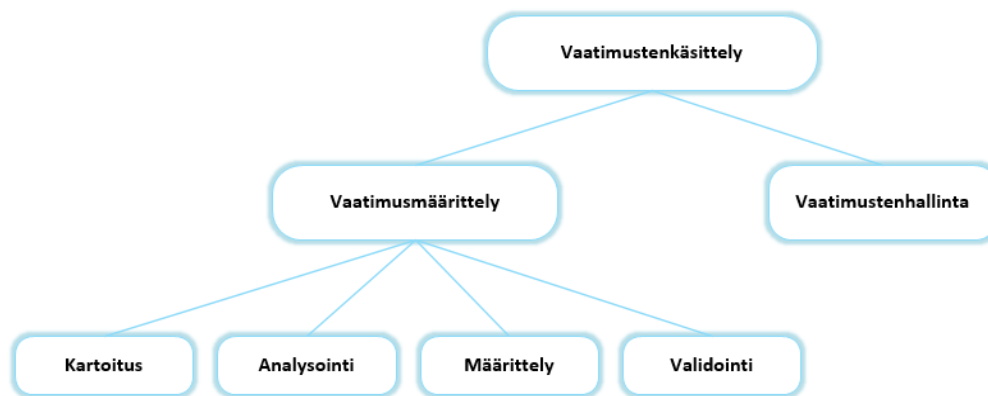


## 5 VAATIMUSMÄÄRITTELY

Tietojärjestelmän hankinnan valmisteluvaiheen tehtävistä vaatimusten määrittely on yleensä suuritöisin. Ei olekaan tavatonta, että vaatimusmäärittelyyn käytetty työmäärä on jopa yhtä suuri kuin varsinaisen toteutuksen. Vaatimusmäärittelyn tavoitteena on muodostaa kaikkien kehittämisen osapuolten yhteinen näkemys hankittavan tietojärjestelmän toiminnallisuuksista sekä sen reunaehdoista. Näitä osapuolia ovat esimerkiksi tulevat käyttäjät, päätöksentekijät, muut sidosryhmät sekä järjestelmän kehittäjät. (Forselius 2013, 26–29.)

Aina vaatimusmäärittely ei kuitenkaan ole osa kokonaisen tietojärjestelmän hankintaa. Jos kyseessä on esimerkiksi uuden toiminnallisuuden lisääminen olemassa olevaan ohjelmistoon tai uuden ohjelmiston lisääminen laajempaan tietojärjestelmään, tehdään vaatimusmäärittelyä vain tietojärjestelmän osalle, kuten ohjelmistolle. Tällöin tehtävän vaatimusmäärittelyn tulee erityisesti huomioida myös laajemman kokonaisuuden asettamat vaatimukset, kuten esimerkiksi rajapinnat. (IEEE 830-1998, 4.)

Vaatimusmäärittelyn ja vaatimustenhallinnan suhteesta toisiinsa on useita erilaisia näkemyksiä. Jotkut kuvaavat vaatimusmäärittelyn yläkäsitteeksi, jonka yhtenä osa-alueena on vaatimustenhallinta. Toiset taas kuvaavat vaatimustenhallintaa tukitoimintona, jonka yhtenä osana on vaatimusten määrittely. Kuvassa 4 on esitetty Wiegertsin ja Beattyn näkemys vaatimusmäärittelyn osa-alueista. Kuvattiin vaatimusmäärittelyn osa-alueita miten hyvänsä, tulee alla kuvatut osa-alueet toteuttaa tavalla tai toisella. (Wiegerts & Beatty 2013.)

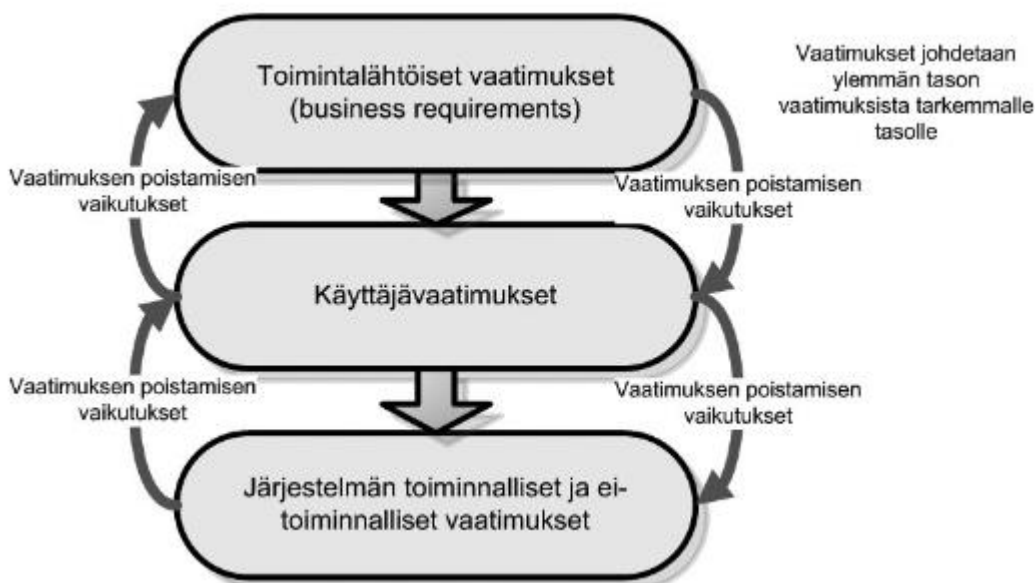


Kuva 4. Vaatimusten käsittelyn osa-alueet (Wiegerts & Beatty 2013.)

## 5.1 Vaatimus

Vaatimuksen määritelmä sekä käytetyt termit vaihtelevat kontekstista riippuen. Eri kohderyhmille yksittäinen vaatimus voi olla esimerkiksi käyttäjävaatimus, ohjelmistovaatimus, liiketoimintavaatimus, toiminnallinen vaatimus, järjestelmävaatimus, ja niin edelleen. Tietojärjestelmää hankkivan asiakkaan käyttämä termi vaatimukselle saattaa kuulostaa kehittäjän mielestä korkean tason tavoitteelta, ja kehittäjän käyttämä termi taas asiakkaan mielestä tarkan tason käyttöliittymäkuvaukselta. Tietojärjestelmän vaatimuksen määritelmäksi ei myöskään sovi sanakirjan määritelmä vaatimukselle. (Wiegiers & Beatty 2013.)

Haikala ja Mikkonen (2011, 61) määrittelevät vaatimuksen kuvaavan toiminnallisuutta, joka tuotteella tulee voida tehdä tai ominaisuudeksi, joka sillä tulee olla. Kotonyan ja Sommervillen (1998, 4) määritelmän mukaan vaatimukset määrittelevät palvelut, jotka järjestelmän pitää tuottaa sekä reunaehdot, joiden rajoissa järjestelmän tulee toimia. ISO/IEC/IEEE:n standardi (2011, 5) määrittelee vaatimuksen lausunnoksi, joka tulkitsee ja tarkentaa tarpeet ja niiden ehdot sekä rajoitukset.



Kuva 5. Vaatimusten ryhmittely (JUHTA 2009b, 10.)

JUHTA (2009b, 10) jakaa vaatimusmäärittelyä koskevassa suosituksessaan vaatimukset kolmeen ryhmään, jotka on esitetty kuvassa 5. Wiegiersin ja Beattyn mukaan toimintälähtöiset vaatimukset kuvaavat, miksi organisaatio on hankkimassa tietojärjestelmää, ja mitä sillä pyritään saavuttamaan. Toimintälähtöiset vaatimukset esittää tyypillisesti hankkeen rahoittaja, asiakas tai tulevan loppukäyttäjäorganisaation johto. Toimintälähtöiset vaatimukset kuvataan tavallisesti jo hankkeen asettamiskirjassa, business casessa tai vastaavassa dokumentissa.

Käyttäjävaatimukset kuvaavat toimia, joita järjestelmän käyttäjän tulee voida tuotteen avulla toteuttaa. Käyttäjävaatimuksista käytetään myös laajempaa nimitystä sidosryhmävaatimukset, joka kuvaa paremmin tyypillisen tietojärjestelmän useampien käyttäjäryhmien asettamia vaatimuksia. Kaikkien käyttäjävaatimusten tulee olla linjassa toimintalähtöisten vaatimusten kanssa. (Wiegers & Beatty 2013.)

Järjestelmävaatimukset jaetaan tavallisesti toiminnallisiin ja ei-toiminnallisiin vaatimuksiin. Toiminnalliset vaatimukset kuvaavat järjestelmältä vaadittua toiminnallisuutta, joka tuottaa toiminnalle sen tavoittelemat hyödyt. Ei-toiminnalliset vaatimukset taas kuvaavat esimerkiksi teknisiä ja laadullisia vaatimuksia ja rajoitteita järjestelmälle tai sen toteuttamiselle. Rajanveto toiminnallisten ja ei-toiminnallisten vaatimusten välillä ei aina ole helppoa. Vaikka vaatimus saattaa äkkiseltään vaikuttaa ei-toiminnalliselta vaatimukselta, saattaa se tarkemmin tutkittaessa paljastua myös toiminnalliseksi vaatimukseksi ja toisinpäin. (Pohjonen 2002, 28; Kotonya & Sommerville 1998, 188.)

Vaatimusmäärittelyä koskevilla standardeilla, kuten IEEE 830-1998 ja ISO/IEC/IEEE 29148:2011, vaatimukset luokitellaan eri tavoin. Kummastakin standardista löytyvät niin ikään toiminnalliset vaatimukset, mutta sen sijaan ei-toiminnalliset vaatimukset on avattu laajemmin omien luokkiensa alle. IEEE 830-1998 jakaa vaatimukset kuuteen eri luokkaan. ISO/IEC/IEEE 29148:2011 taas jakaa vaatimukset 13 eri luokkaan. Kummassakin standardissa yksi luokka kuvaa toiminnallisia vaatimuksia ja jäljelle jäävät luokat ei-toiminnallisia. Kuten yleisemmin käytössä olevassa kaksijakoisessa luokittelussa, myös näissä tarkemmissa luokitteluissa ongelmana on vaatimusten tarkka jaottelu eri luokkiin. (Granholm 2013, 116–119.)

### 5.1.1 Toiminnalliset vaatimukset

Toiminnalliset vaatimukset määrittelevät sen toiminnallisuuden, mitä järjestelmässä tulee olla. Ne kuvaavat myös esimerkiksi sen, miten järjestelmä kommunikoi muiden järjestelmien kanssa. Toiminnalliset vaatimukset voivat olla geneerisiä vaatimuksia järjestelmän toiminnasta tai tarkan tason vaatimuksia esimerkiksi järjestelmän kommunikoinnista vanhojen järjestelmien kanssa. (Sommerville 2011, 85–87.)

Toiminnalliset vaatimukset tulisi olla johdettavissa esitutkimusvaiheessa kerätyistä toiminnan sekä sidosryhmien tarpeista. Toiminnalliset vaatimukset eivät ota kantaa vaatimuksen täyttämistapaan eli esimerkiksi järjestelmän toteutustapaan, vaan kuvaavat vain, mitä toiminnallisuutta järjestelmältä vaaditaan.

### 5.1.2 Ei-toiminnalliset vaatimukset

Siinä missä toiminnalliset vaatimukset kuvaavat järjestelmältä vaadittavia toiminnallisuuksia, ei-toiminnalliset vaatimukset tavallisesti asettavat järjestelmälle ja sen ympäristölle rajoitteita ja reunaehtoja. Ei-toiminnalliset vaatimukset voivat kuvata esimerkiksi vastausaikoja, käytettävyyttä, järjestelmän ylläpidettävyyttä ja käytettäviä standardeja. Myös lainsäädäntö sekä organisaation toimiala ja omat prosessit asettavat järjestelmälle ja sen kehittämislle vaatimuksia. Koska ei-toiminnalliset vaatimukset ovat reunaehtoja tai rajoitteita, voivat ne tärkeytensä vuoksi jopa edellyttää joidenkin toiminnallisten vaatimusten hylkäämistä. (Pohjonen 2002, 29; Kotonya & Sommerville 1998, 187–188.)

Ei-toiminnalliset vaatimukset koskevat usein koko järjestelmää, kun toiminnalliset vaatimukset kohdistuvat johonkin yksittäiseen toiminnallisuuteen. Jotkin ei-toiminnalliset vaatimukset, kuten käytettävä ohjelmointikieli tai käyttöliittymästandardi, kohdistuvat myös järjestelmän toteutustapaan. Koska ei-toiminnalliset vaatimukset usein koskevat koko järjestelmää, ne saattavat edellyttää muutoksia myös moniin toiminnallisiin vaatimuksiin. (Sommerville 2011, 87–90.)

### 5.1.3 Rajoitteet

Rajoitteet on eräänlainen erikoistapaus vaatimuksista. Ne voivat olla osana jostain toista vaatimusta ja rajoittaa esimerkiksi vaaditun toiminnallisuuden suoritusaikaa tai tulosjoukkoa. Ne voivat olla myös omana vaatimuksenaan, joka on sidoksissa toisiin vaatimuksiin tai täysin itsenäisenä vaatimuksena. Kuvattaessa rajoite toisen vaatimuksen sisään, ei rajoitetta kutsuta vaatimukseksi, eikä sitä tällöin tarvitse kuvata vaatimusten kuvaamisen edellyttämällä tavalla. Erillisinä vaatimuksina kuvatut rajoitteet luokitellaan tyypillisesti oman luokkansa alle tai osaksi ei-toiminnallisia vaatimuksia riippuen käytetystä luokittelutavasta. (Haikala & Mikkonen 2011, 61; Granholm 2013, 122–123.)

Esimerkiksi lainsäädäntö asettaa monesti rajoitteita tietojärjestelmälle tai sen tukemalle toiminnalle. Teknisempi esimerkki rajoitteesta voisi olla kohdelaitteen käytettävissä olevan fyysisen muistin määrä. Rajoitteet ovat tyypillisesti lopullisia, eikä niiden osalta käydä neuvottelua tai tehdä priorisointia, kuten vaatimusten osalta.

### 5.2 Kartoitus

Vaatimusten kartoitusvaiheen tarkoituksena on tunnistaa erilaisin keinoin asiakkaan ja sidosryhmien vaatimukset järjestelmälle. Vaatimusten kartoitus ei ole ainoastaan asiakkaiden kertomien vaatimusten keräämistä yhteen dokumenttiin, vaan kyseessä on analyyttinen ja kollaboratiivinen prosessi, jossa yhdessä sidosryhmien kanssa kehitetään vaatimuksia. Vaatimusten kartoittamiseen pitäisi osallistua kaikki tulevan järjestelmän sidosryhmät, kuten järjestelmän omistaja, käyttäjät, kumppanit, toimittajat sekä kaikki sellaiset tahot, joilla voi olla vaatimuksia järjestelmälle. (Kosola 2013, 24–28; Wiegers & Beatty 2013.)

Vaatimusten kartoittaminen useiden erilaisten sidosryhmien kanssa on haastava prosessi. Sidoryhmillä ei välttämättä ole selkeää käsitystä siitä, minkälaisia vaatimuksia järjestelmälle tulisi asettaa tai he eivät osaa kuvata mitä he järjestelmältä haluavat. Toisaalta asiakasorganisaation ulkopuolisilla sidoryhmillä taas voi olla hyvinkin erilaiset tarpeet tulevalle järjestelmälle kuin järjestelmän loppukäyttäjillä. Myös taloudelliset, liiketoiminnalliset sekä poliittiset tekijät voivat vaikuttaa vaatimuksiin ja erityisesti niiden priorisointiin. Tästä syystä vaatimusten kartoittamiseen käytetään tyypillisesti useita erilaisia tekniikoita. (Sommerville 2011, 102–104.)

Paakki on jakanut vaatimusten kartoitustekniikat suoriin ja epäsuoriin tekniikoihin. Suoria kartoitustekniikoita ovat esimerkiksi haastattelut, havainnointi sekä työpajat, ja epäsuoria tekniikoita ovat esimerkiksi taustatutkimus, kyselyt, kertomukset sekä prototyypit.

Haastattelut ovat tärkein vaatimusten kartoitustekniikka. Erilaisia haastattelutyyppejä ovat esimerkiksi avoimet haastattelut ja teemahaastattelut. Avoimet haastattelut ovat keskustelunomaisia, vapaamuotoisia tilaisuuksia, joiden perusteella pyritään kartoittamaan tarpeita alkuvaiheessa. Teemahaastattelussa haastattelun teemat on etukäteen päätetty, ja haastattelija voi teemahaastattelulla pyrkiä tarkentamaan esimerkiksi aiemmin hankittua tietoa.

Työpajat voivat olla joko rakenteellisia tai rakenteettomia, kuten haastattelutkin. Vaatimusten kartoitus työpajoissa on samantapainen prosessi kuin haastatteluisa, mutta suurempi osallistujamäärä mahdollistaa ideoiden jalostamisen, ja sitä kautta uusien ideoiden syntyminen haastatteluja paremmin.

Havainnoinnissa havainnoija voi esimerkiksi osallistua havainnoitavaan toimintaan tai tarkkailla havainnoitavaa toimintaa tai henkilöä. Havainnoinnilla voidaan havaita myös sellaisia asioita, joita haastatteluisa jätetään mainitsematta, joko tietoisesti tai epähuomiossa. Haastateltava saattaa esimerkiksi pitää jotain asiaa itsestäänselvyytenä. Havainnoinnilla voidaan tyypillisesti tunnistaa vain nykyjärjestelmän toimintaa ja heikkouksia, mutta ei tulevan järjestelmän tarpeita.

Taustatutkimuksen tarkoitus on tutustua olemassa olevaan dokumentaatioon ja tunnistaa siitä vaatimuksia. Dokumentaatio voi olla esimerkiksi organisaatioon, ongelmakenttään ja nykyjärjestelmiin liittyvää dokumentaatiota. Taustatutkimuksessa haasteena on tyypillisesti dokumentaation suuri määrä.

Kertomuksilla voidaan kuvata miten järjestelmän pitäisi toimia. Tyypillisin kertomustyyppi on skenaario. Niitä voidaan pitää eräänlaisena käsikirjoitukseksi järjestelmälle. Tarkoitus on kuvata, mitä järjestelmällä saa ja ei saa tehdä (positiiviset ja negatiiviset skenaariot). Skenaariot ovat eri sidosryhmille helposti ymmärrettäviä kuvauksia järjestelmän toiminnasta, mutta tyypillisesti ne eivät kata koko järjestelmää, vaan toimivat vain esimerkkeinä.

Prototyypit ovat yksinkertaistettuja malleja tulevasta järjestelmästä tai sen osasta. Ne toimivat kirjoitettua tekstiä parempana esimerkkinä eri sidosryhmille, ja niiden perusteella on helpompi tarkentaa toiminnallisia vaatimuksia. (Paakki 2011, 54–77.)

### 5.3 Analysointi

Analysointivaiheessa käydään huolellisesti läpi jokainen kartoitusvaiheessa tunnistettu vaatimus. Vaatimusten kartoitusvaiheessa tunnistetut vaatimukset ovat alustavia, ja analysointivaiheessa ne muuttuvat ja täydentyvät. Vaatimuksia saatetaan yhdistää tai niitä voidaan jakaa useammaksi vaatimukseksi. Vaatimuksista etsitään esimerkiksi mahdollisia ongelmia ja ristiriitaisuuksia, varmistetaan, ettei yhdessä vaatimuksessa todellisuudessa ole useampaa vaatimusta sekä varmistetaan vaatimuksen tarpeellisuus ja toteuttamiskelpoisuus. (Kotonya & Sommerville 1998, 77–79.)

Analysoidut vaatimukset ryhmitellään tarkoitukseen sopivalla tavalla ja ne tulee yksilöidä. Ryhmittelyssä käytetään yleisimmin arkkitehtuurisuunnittelumallia, jossa järjestelmä jaetaan osakokonaisuuksiin, ja ryhmitellään vaatimukset näiden kokonaisuuksien mukaan. Vaatimusten yksilöintiin käytetään tyypillisesti juoksevaa numerointia tai ryhmittelyn mukaan juoksevaa numerointia. Vaatimusten yksilöinnillä mahdollistetaan vaatimukseen viittaaminen muualta. Ryhmittelyn ja yksilöinnin lisäksi vaatimuksille kirjataan perustelut, vaatimuksen esittäjä sekä liittyminen muihin vaatimuksiin. Nämä tiedot ovat tarpeen esimerkiksi vaatimukseen kohdistuvien muutostarpeiden selvittämisessä järjestelmän elinkaaren myöhemmissä vaiheissa. (Haikala & Märijärvi 2000, 82–85; Sommerville 2011, 100–102.)

Analysoinnin yhteydessä vaatimukset tulee myös priorisoida. Priorisoinnilla pyritään varmistamaan tärkeimpien vaatimusten toteuttaminen lopulliseen järjestelmään. Esimerkiksi keskenään ristiriitaisten vaatimusten esiintyessä voidaan toinen vaatimuksista joutua poistamaan. Myös ohjelmiston toteuttamisvaiheessa, aikataulu- tai taloudellisista syistä, joudutaan usein jättämään osa

vaatimuksista täyttämättä tai siirtää niitä myöhemmin toteutettaviksi. (JUHTA 2009b, 15.)

#### 5.4 Määrittely

Määrittelyvaiheen tarkoitus on tarkentaa vaatimukset formaaliin muotoon ja dokumentoida ne. Vaatimukset tulisi mahdollisuuksien mukaan kirjoittaa sellaiseen muotoon, että ne kuvaavat, mitä järjestelmän tulee tehdä, eikä sitä, miten se tulee tehdä. Näin vaatimuksia kirjoitettaessa ei vielä rajoiteta mahdollista toteutusta. Joissakin tilanteissa voi kuitenkin olla järkevää myös asettaa rajoituksia vaatimusten täyttämiseksi. Esimerkiksi hankittavan järjestelmän yhteensopivuus muiden organisaation käytössä olevien järjestelmien kanssa saattaa asettaa rajoituksia toteutukselle. (Kotonya & Sommerville 1998, 7.)

Taulukossa 1 on listattu standardissa ISO/IEC/IEEE 29148:2011 kuvatut ominaisuudet, jotka vaatimuksen tulisi täyttää. Taulukossa esitettyjen ominaisuuksien lisäksi jokaisesta vaatimuksesta tulisi kuvata myös vähintään vaatimuksen esittäjä, omistaja, kriittisyys sen omistajalle sekä tietenkin itse sisältö.

Taulukko 1. Vaatimuksen ominaisuudet (ISO/IEC/IEEE 29148:2011, 11.)

<i>OMINAISUUS</i>	<i>SELITYS</i>
<i>TARPEELLISUUS</i>	Vaatimus on välttämätön järjestelmän halutun toiminnallisuuden saavuttamiseksi. Jos vaatimus poistettaisiin, järjestelmä jäisi vajaaksi, eivätkä muut vaatimukset täyttäisi tarvetta. Vaatimukset ovat edelleen tarpeellisia, ja sellaiset vaatimukset, joilla on suunniteltu voimassaoloaika, on selkeästi tunnistettu.
<i>TOTEUTUS-RIIPPUMATTOMUUS</i>	Vaatimuksella kuvataan vain se, mikä on tarpeen järjestelmän toiminnan kannalta. Vaatimuksessa ei rajoiteta tapaa, jolla se tulee täyttää.
<i>YKSIKÄSITTEISYYS</i>	Vaatimus on kuvattu yksinkertaisesti ja se on helposti ymmärrettävissä. Vaatimusta ei voi tulkita väärin.
<i>RISTIRIIDATTOMUUS</i>	Vaatimus ei ole ristiriidassa minkään toisen vaatimuksen kanssa.
<i>TÄYDELLISYYS</i>	Vaatimus yksin tai muiden asetettujen vaatimusten kanssa, täyttää kaikki sidosryhmien järjestelmälle asettamat tarpeet.
<i>YKSILÖLLISYYS</i>	Vaatimuksessa on kuvattu vain yksi vaatimus, eikä vaatimus ole päällekkäinen yhdenkään toisen vaatimuksen kanssa.
<i>TOTEUTETTAVUUS</i>	Vaatimus on teknisesti toteutettavissa, huomioiden järjestelmälle asetetut rajoitteet.

**JÄLJITETTÄVYYS**

Vaatus on jäljitettävissä sekä eteen- että taakse-päin. Vaatus tulee voida jäljittää ylemmän tason vaatimuksiin ja aina sidosryhmän esittämään tarpeeseen asti sekä toisaalta lähdekoodiin asti.

**TODENNETTAVUUS**

Vaatus on kuvattu niin, että sen täytyminen järjestelmässä on todennettavissa.

Standardi ISO/IEC/IEEE 29148:2011 kuvaa myös mallin vaatimusdokumentille (Software Requirements Specification, SRS), jota kutsutaan toisinaan myös toiminnalliseksi määrittelyksi. Tekstimuotoisten vaatimusten rinnalla käytetään usein visuaalisia malleja, kuten käsitemalleja, tilakaavioita ja tietovirtakaavioita. Vaatimuksia ei tarvitse välttämättä dokumentoida yhteen standardin määrittelemään dokumenttiin, vaan dokumentointitapa voi olla määritetty esimerkiksi organisaation standardeissa. Nykyisin vaatimukset dokumentoidaan usein vaatimustenhallintaan tarkoitettuihin tietojärjestelmiin, joissa niiden ylläpito on yksinkertaisempaa. (Wiegiers & Beatty 2013.)

## 5.5 Validointi

Validointi on vaatimusmäärittelyn viimeinen vaihe. Kotonya ja Sommerville summaavat vaiheen tarkoitukseksi katselmoida vaatimusmäärittelyn tuotos, vaatimusdokumentti, ja hyväksyä se tulevan järjestelmän suunnittelun pohjaksi. Kaikkien vaatimusmäärittelyyn osallistuneiden tahojen tulisi osallistua validointivaiheeseen.

Validointivaiheen tehtävät ovat osittain päällekkäisiä analysointivaiheen kanssa, mutta vaiheiden päämäärä on erilainen. Validoinnin pohjana tulisi tarkastella sidosryhmien alkuperäisiä tarpeita, ja varmistaa että kaikki tarpeet on tunnistettu ja ne täyttävät vaatimukset on kuvattu oikein. Kotonya ja Sommerville kuvaavat analyysivaiheen tarkoitukseksi selvittää, onko oikeat vaatimukset tunnistettu (*Have we got the right requirements?*), ja validointivaiheen tarkoitukseksi selvittää, onko vaatimukset kuvattu oikein (*Have we got the requirements right?*). (Kotonya & Sommerville 1998, 87–88.)

Vaatimusten validointiin käytetään erilaisia keinoja. Tyypillisin tapa on järjestää validointitilaisuuksia, joissa vaatimusdokumentti luetaan ja siitä etsitään vikoja tai puutteita. Ilman konkretiaa on kuitenkin hankala varmistua siitä, että vaatimus todellisuudessa täyttää asiakkaan tarpeet. Prototyyppejä käytämällä voidaan konkretisoida vaatimuksia sidosryhmille, mikä helpottaa tarpeen täyttymisen arviointia. Validointivaiheessa voidaan myös suunnitella jo testitapaukset vaatimusten todentamiseksi. Mikäli vaatimukselle ei pystytä määrittelemään selkää testitapausta, voi vaatimus olla huonosti kuvattu. (Sommerville 2011, 110–111.)



### 5.6 Vaatimusten hallinta

Laajojen tietojärjestelmien vaatimukset muuttuvat jatkuvasti. Valtaosa järjestelmän vaatimuksista tunnistetaan ja määritellään esitutkimus- ja määrittelyvaiheissa, mutta tyypillisesti niitä tunnistetaan lisää tai muutetaan jo projektin edetessä. Projektin alkuvaiheessa kaikkia asiakasvaatimuksia ei välttämättä ole tunnistettu tai ymmärretty oikein, toimintaympäristössä tapahtuu muutoksia, jotka heijastuvat järjestelmän vaatimuksiin tai alkuvaiheessa tehdyt teknologiavalinnat osoittautuvat myöhemmin kelvottomiksi. Syitä on monia ja muutoksia tulee väistämättä, joten muutoksiin tulee varautua. (Haikala & Märijärvi 2000, 86–87.)

Vaatimustenhallinnan tarkoitus on hallita hyväksytyihin vaatimuksiin tapahtuvia muutoksia ja ylläpitää vaatimusten välisiä suhteita sekä riippuvuuksia. Esitettyjen muutosten vaikutukset järjestelmään ja muihin vaatimuksiin tulee tunnistaa ja analysoida ennen niiden hyväksymistä. Uuden vaatimuksen hyväksyminen tai aiemmin hyväksytyjen vaatimusten muuttaminen aiheuttaa muutoksia myös järjestelmän suunnitteluun ja toteutukseen. Mikäli muutoksia ei hallita, saattaa prioriteetiltaan tärkeämmät vaatimukset jäädä toteuttamatta tai lykkääntyä, mikä voi aiheuttaa projektin aikataulun ja budjetin ylittymisen. (Kotonya & Sommerville 1998, 113–114.)

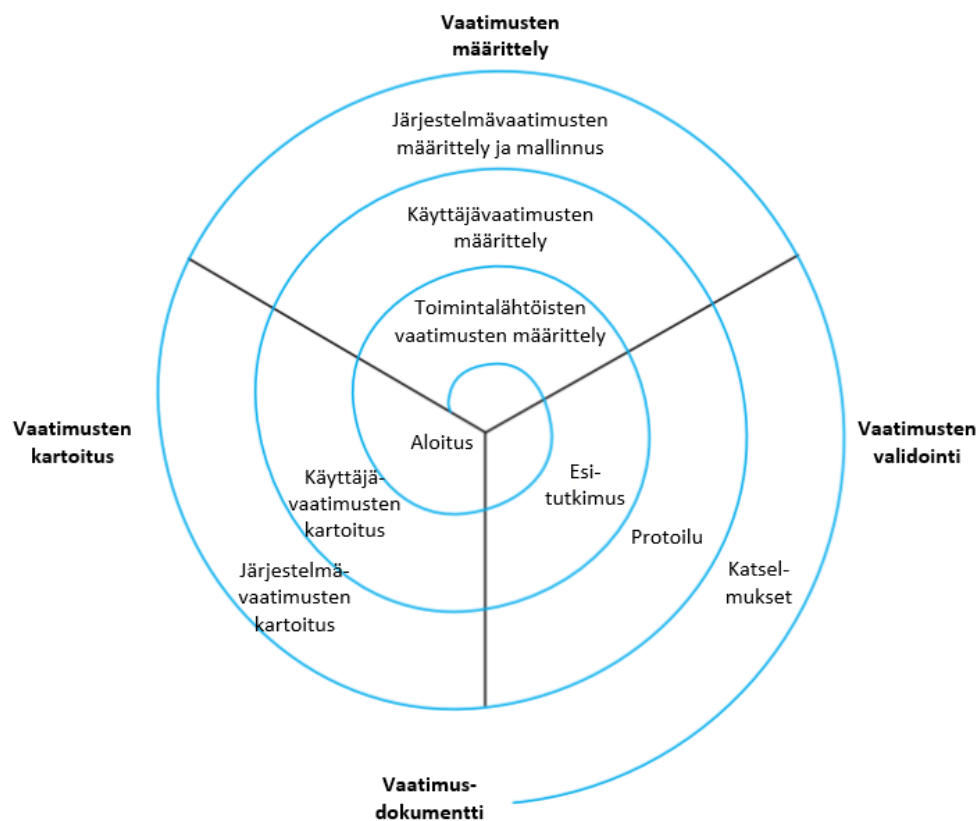
Vaatimusten väliset riippuvuudet sekä jäljitettävyyden eteen- ja taaksepäin ovat tärkeässä asemassa vaatimustenhallinnassa. Vaatimusmuutosta arvioitaessa voidaan esimerkiksi jäljittää muutoksen vaikutus ohjelmakoodiin asti sekä toisaalta tarkistaa mikä asiakasvaatimus mahdollisesti jää täyttämättä, jos vaatimus poistetaan. Riippuvuuksien ylläpitämisen hyödyt ilmenevät vasta projektin myöhemmissä vaiheissa tai mahdollisesti vasta ohjelmiston ylläpitovaiheissa, mistä johtuen sen tarpeellisuutta saatetaan vähätellä ja jopa jättää kokonaan tekemättä. Tämä taas tavallisesti johtaa ohjelmiston elinkaaren aikana korkeampiin kustannuksiin. (Haikala & Märijärvi 2000, 87; Kotonya & Sommerville 1998, 115.)

### 5.7 Vaatimusmäärittelyn prosessi

Eri organisaatiot määrittelevät ja hallitsevat vaatimuksia eri tavoin. Esimerkiksi hävittäjälentokoneita valmistavan organisaation vaatimusmäärittelyprosessi on todennäköisesti erilainen, kuin verkkokauppasovellusta kehittävän, kolmen hengen IT-yrityksen. Yleensä erot prosesseissa löytyvät kuitenkin niiden tarkkuustasosta, ja abstraktimmalla tasolla suurin osa vaatimusmäärittelyprosesseista seuraa samaa mallia (Kotonya & Sommerville 1998, 32).

Vaatimusmäärittely on iteratiivinen prosessi. Vaikka monesti (esimerkiksi Wiegers & Beatty 2013 ja Kotonya & Sommerville 1998, 32) vaatimusmäärittelyn prosessia kuvataankin peräkkäin toteutettavina vaiheina, todellisuudessa vaiheet limittyvät keskenään. Esimerkiksi vaatimusten kartoituksen yhteydessä tavallisesti tehdään jo analysointia ja priorisointia. Sommerville

(2011) on kuvannut vaatimusmäärittelyprosessia kuvassa 6 esitetyllä spiraalin muotoisella mallilla, joka etenee toimintälähtöisten vaatimusten määrittelystä valmiiseen vaatimusdokumenttiin.



Kuva 6. Vaatimusmäärittelyprosessi (Sommerville 2011, 99.)

### 5.8 Vaatimusmäärittely ketterässä ohjelmistokehityksessä

Kuten luvuissa 3 ja 4 kuvataan, vaatimusmäärittely on perinteisesti kuulunut hankinnan ja ohjelmistotuotannon valmisteluvaiheen tehtäviin. Tyypillisesti tämä tarkoittaa luvussa 5 kuvattujen vaatimusmäärittelyn osa-alueiden toteuttamista.

Ketterässä ohjelmistokehityksessä ohjelmiston toteuttaminen pyritään aloittamaan mahdollisimman varhaisessa vaiheessa. Tällöin vaatimuksia ei kuvatakaan ennen toteutusvaihetta tarkasti vaatimuksen määritelmän mukaisesti, vaan niitä kuvataan epäformaalein käyttäjätarinoin koko toteutusvaiheen ajan. Käyttäjätarinoilla kuvataan lyhyitä skenaarioita käyttäjän näkökulmasta ja ne ovat ketterässä ohjelmistokehityksessä vaatimusten pääasiallinen esitysmuoto. Kehitystyön alkuvaiheessa määritellään asiakkaan kannalta kaikkein tärkeimmät vaatimukset tai ominaisuudet, minkä jälkeen toteutustyö päästään aloittamaan. (Sommerville 2011, 57–58.)

Koska kaikkia ohjelmiston vaatimuksia ei ole määritelty valmisteluvaiheessa, on asiakas ketterässä ohjelmistokehityksessä tiiviisti mukana koko toteutusvaiheen ajan. Asiakkaan tehtävänä on erityisesti tarkentaa ja priorisoida vaatimuksia. Ketterissä malleissa vaatimusmäärittely onkin jätetty asiakkaan, tai esimerkiksi Scrumissa tuoteomistajan, tehtäväksi. Näin ollen yksi ohjelmistotuotannon haastavimmista osa-alueista jää menetelmän ulkopuolelle, ja on tuoteomistajan osaamisesta kiinni, kuinka hyvin kaikkien sidosryhmien vaatimukset tulee huomioitua. (Haikala & Mikkonen 2011, 65; Leffingwell 2011, 51.)

Leffingwell toteaa myös vaatimusten priorisoinnin eroavan perinteisesti kuvasta vaatimuksen tarpeellisuutta kuvaavasta priorisoinnista. Ketterässä mallissa tarinoille tulee saada keskenään verrattavissa oleva prioriteetti, jonka perusteella korkeimman prioriteetin omaavat tarinat otetaan ensimmäisenä toteutukseen. Priorisointia tehdään myös jatkuvasti projektin edetessä. Alkuvaiheessa priorisoidaan suurempia kokonaisuuksia, ja iteraatioittain tulee priorisoida vähintään seuraavaan toteutusiteraatioon mukaan tulevat ominaisuudet.

Ei-toiminnallisten vaatimusten tunnistamiseen ketterät menetelmät eivät juuri tuo lisäarvoa. Ketterien menetelmien ajatus keveistä käyttäjätarinoista ei monesti sovi ei-toiminnallisten vaatimusten tunnistamiseen. Vaikka ne tavallisesti tunnistetaankin perinteisin keinoin, Leffingwell ehdottaa, että myös ei-toiminnalliset vaatimukset voidaan kuvata käyttäjätarinoiden muotoon. Toisena hyvänä vaihtoehtona ei-toiminnallisten vaatimusten esittämiseen Leffingwell mainitsee erillisen vaatimusdokumentin. (Leffingwell 2011, 209–210, 342, 347–348.)

Koska projektin edetessä vaatimukset muuttuvat ja niitä tunnistetaan lisää, perinteisessä vaatimusmäärittelyssä hyväksytyä vaatimusmäärittelyä ylläpidetään vaatimustenhallinnan avulla. Ketterässä mallissa erillistä vaatimustenhallintaa ei kehitystyön aikana tarvita, koska muutoksia ja uusia vaatimuksia arvioidaan jatkuvasti uusien iteraatioiden yhteydessä. Tällöin vaatimustenhallintaa otetaan tavallisesti käyttöön vasta tuotteen käyttöönoton yhteydessä. (Haikala & Mikkonen 2011, 67.)

## 6 CASE OIKEUSREKISTERIKESKUS

Työn toimeksiantajana toimii Oikeusrekisterikeskus, joka on oikeusministeriön hallinnonalalla toimiva virasto. Oikeusrekisterikeskuksen tehtävistä määrätään lailla 625/2012, joka tuli voimaan 1.4.2013 oikeusministeriön hallinnonalalla toteutetun tietohallinnon palvelutuotannon uudelleenorganisoinnin yhteydessä.

Laki Oikeusrekisterikeskuksesta määrittelee Oikeusrekisterikeskuksen tehtävät seuraavasti:

Oikeusrekisterikeskus on oikeusministeriön hallinnonalaan kuuluva virasto, jonka tehtävänä on:

- 1) toimia oikeusministeriön hallinnonalan tietojärjestelmien ja rekisterien rekisterinpitäjänä sekä välittää hallinnonalan viranomaisten ilmoittamia tietoja muille viranomaisille;
- 2) huolehtia sakkoihin, menettämisseuraamuksiin, maksuihin ja saataviin liittyvistä täytäntöönpanotehtävistä sekä käyttää valtion puhevaltaa näissä tehtävissä;
- 3) huolehtia oikeusministeriön hallinnonalaan kuuluvan yksikön toimeksiannosta tutkimustarkoituksiin käytettävien rekisterien ylläpidosta ja niihin liittyvistä tehtävistä;
- 4) huolehtia oikeusministeriön hallinnonalan tietojärjestelmien ylläpidosta ja kehittämisestä yhteistyössä hallinnonalan virastojen kanssa siten kuin palvelusopimuksissa on sovittu.

(Laki Oikeusrekisterikeskuksesta 1§.)

Uudelleenorganisoinnin myötä aiemmin Oikeushallinnon tietotekniikkakeskuksen tehtäviin kuuluneet tietojärjestelmien kehittäminen ja ylläpito siirtyivät Oikeusrekisterikeskuksen tehtäviksi. Oikeusrekisterikeskus huolehtii sekä asiakkaidensa tuotantokäytössä olevien tietojärjestelmien kunnossapidosta ja kehittämisestä että uusien tietojärjestelmien kehittämishankkeista. Oikeusrekisterikeskuksen asiakkaita ovat muun muassa tuomioistuimet, ulosottovirastot, syyttäjävirastot sekä rikosseuraamuslaitos.

Oikeusrekisterikeskuksessa työskentelee tällä hetkellä noin 120 henkilöä ja toiminta on organisoitu neljälle toimialalle, joiden lisäksi johtajan alaisuudessa toimii hankintoihin ja sopimushallintaan erikoistunut ryhmä. IT-palvelutuotanto jakautuu kolmelle eri toimialalle. Viranomaistoiminta -toimialan yhtenä tehtävänä on huolehtia vaalitietojärjestelmän ylläpito- ja tukitoiminnasta. Asiakkuuksien, talouden ja palveluiden hallinta -toimialan teh-

tävänä on vastata IT-palvelutuotannon asiakkuudenhallinnasta ja sen toimintamallien kehittämisestä sekä tuotantokäytössä olevien tietojärjestelmien tuki-toiminnasta. Järjestelmäkehitys- ja projektipalvelut -toimialan tehtävänä on huolehtia tietojärjestelmien kehittämistehtävistä sekä projektitoiminnasta.

### 6.1 Tarve toimintaa tukevalle järjestelmälle

Melko pian uudelleenorganisoinnin jälkeen havaittiin tarve uutta toimintamallia tukevalle järjestelmälle. Yleisesti puhuttiin Toiminnanohjausjärjestelmästä, joka esiintyy käsitteenä myös opinnäytetyön liitteenä olevassa vaatimusmäärittelyssä. Eri tahoilla oli erilaisia tarpeita järjestelmälle omien tehtäviensä tueksi.

Järjestelmäkehitys- ja projektipalveluita tuotetaan Oikeusrekisterikeskuksen muille toimialoille tilaaja-tuottaja-mallia mukaillen, joten tilauksia varten tarvittiin pikaisesti kanava. Kanavaa pitkin palvelupäälliköt sekä projektipäälliköt voivat tehdä työtilauksia ja esittää resurssitarpeita. Ensi hätään toteutettiin tilauslomake käytössä olevalle intranet-alustalle, josta työtilaukset ohjautuivat sähköpostitse niiden vastaanottajille.

Tietojärjestelmien kehitys- ja kunnossapitotehtävissä käytetään monesti sovellustoimittajan tarjoamaa alustaa muun muassa työtilausten esittämiseen ja järjestelmädokumentaation säilytykseen. Oikeusrekisterikeskuksen asiantuntijat tarvitsevat tapauskohtaisesti useamman toimittajan järjestelmiä töidensä tekemiseen. Keskitetyn sovellustuen työn mahdollistamiseksi osa dokumentaatiosta on jouduttu tallentamaan uudelleen toiseen tietokantaan keskitetyn näkymän saamiseksi.

Keskustelua järjestelmästä ja tarpeista käytiin eri foorumeissa, mutta koska organisoitua ja keskitettyä tarpeiden tunnistamista ei vielä ollut aloitettu, päätettiin järjestelmälle toteuttaa vaatimusmäärittely.

## 6.2 Tarpeiden tunnistaminen ja priorisointi

Ennen opinnäytetyön aloittamista Toiminnanohjausjärjestelmästä on keskusteltu satunnaisesti ja alustavia tarpeita on kartoitettu, mutta varsinaista esiselvitystä järjestelmälle ei kuitenkaan ole tehty. Näin ollen ennen vaatimusten määrittelyn aloittamista tuli toiminnan tarpeet järjestelmälle tunnistaa. Toimintalähtöiset tavoitteet kerättiin haastattelemalla Oikeusrekisterikeskuksen IT-palvelutuotantoon liittyvien toimialojen johtajia sekä muita keskeisiä henkilöitä.

Tarpeiden tunnistuksen yhteydessä kirjattiin yhteensä 64 eritasoista toimintoa, raporttia tai näkymää, jotka järjestelmän tulisi sisältää. Eri käyttäjäryhmiä tai rooleja tunnistettiin 10. Tunnistetut tarpeet jaettiin käyttötarkoituksen ja käyttäjäryhmien mukaan yhdeksään toimintokokonaisuuteen, jotka on esitetty kuvassa 7.



Kuva 7. Tunnistetut toimintokokonaisuudet

Koska järjestelmälle tunnistettiin odotettua enemmän toiminnallisuuksia ja erilaisia kokonaisuuksia, päätettiin tässä vaiheessa tehtävä vaatimusmäärittely rajata vain osaan toiminnallisuuksista; näin kiireellisin tarve saataisiin täytettyä nopeammin. Koska suureen osaan tarvittavaa toiminnallisuutta oli nykyisin olemassa olevat järjestelmät ja menetelmät, ei niiden korvaamista katsottu kiireellisimmäksi. Vaikka nykyisten järjestelmien käyttö hankaloittaakin toimintaa jossain määrin, oletettiin sen kuitenkin hoituvan, kuten aiemminkin. Tehtävien- ja resurssienhallinnan osa-alueen hoitamiseen sen sijaan ei ollut riittäviä työvälineitä, ja sen katsottiin olevan tärkeässä roolissa myös tuke-  
massa koko palvelutuotantomallia ja organisaatiota. Tästä syystä vaatimusmäärittely rajattiin koskemaan kyseistä osa-aluetta ja sen tarvitsemin osin myös muita osa-alueita.

Järjestelmän tärkeimmät toimintälähtöiset tavoitteet ovat toiminnan läpinäkyvyyden ja hallittavuuden parantaminen sekä Oikeusrekisterikeskuksen tavoitteet kokonaisuutena huomioivan tehtävien priorisoinnin tukeminen. Toimintälähtöiset tavoitteet voidaan tiivistää kolmeen kohtaan. Ensiksi sopimukset ja niihin liittyvät tehtävät tulee olla kaikkien Oikeusrekisterikeskuksen työntekijöiden nähtävillä. Työntekijöiden tulee nähdä suorittamansa tehtävän liittymisen asiakkaan sopimukseen. Toiseksi sopimusvelvoitteiden täyttymistä pitää voida seurata reaaliaikaisesti ja vaivattomasti. Kolmanneksi kaikkiin sopimuksiin liittyvät tehtävät ja velvoitteet tulee saada prioriteetteineen näkyviin yhtenä kokonaisuutena. Näin resurssimanageritoiminnolla on mahdollisuus priorisoida Oikeusrekisterikeskuksen tavoitteiden kannalta kulloinkin tärkeimpiä tehtäviä eri asiakkuuksienkin välillä.

### 6.3 Vaatimusten kartoitus

Syksyn 2013 aikana muutamissa ryhmätöissä sekä työpajoissa oli jo kerätty järjestelmälle tarpeita ja vaatimuksia. Koska tarpeita oli jo aiemmin kartoitettu, oli luonnollista aloittaa järjestelmän vaatimusten kartoittaminen niistä. Aiheesta kirjoitetussa dokumentaatioissa esiintyvät tarpeet liittyivät pääosin työtilausten ja -tehtävien hallintaan sekä dokumentaation hallintaan, mutta mukana oli myös pitkä lista ei-toiminnallisia vaatimuksia. Pelkän dokumentaation perusteella sai muodostettua melko selkeän kuvan järjestelmään kaivatusta toiminnallisuudesta. Aiemmin pidetyt ryhmätyöt ja työpajat olivat kuitenkin Järjestelmäkehitys- ja projektipalvelut -toimialan sisäisiä, joten muiden toimialojen näkökulma tuotetussa dokumentaatioissa oli korkeintaan välillinen.

Aiemmin luodun dokumentaation analysoinnin lisäksi vaatimuksia kartoitettiin haastatteluin sekä työpajoin. Haastatteluja käytiin Oikeusrekisterikeskuksen IT-palvelutuotannossa mukana olevien toimialojen edustajien kanssa. Haastattelut toteutettiin avoimina eli tarkkaa strukturoitua agendaa ei ollut, vaan haastattelutilanne oli enemmän keskustelun luonteinen. Haastatteluissa ei myöskään haluttu ohjata haastateltavaa esimerkiksi jollakin temalla, vaan annettiin haastateltavan kuvailla omin sanoin edustamansa tahon tarpeita ja vaatimuksia järjestelmälle. Samalla pyrittiin tunnistamaan kaikki järjestelmän oleelliset sidosryhmät sekä käyttäjäryhmät. Myös työpajat pidettiin pääosin avoimina, eikä tarkkaan määriteltyä agendaa ollut, vaikka työpajan aikana keskityttiinkin tyypillisesti vain pieneen osaan tarpeita.

Kartoitusvaiheessa vaatimukset kerättiin Excel-taulukkoon, jota käytettiin myös myöhemmissä vaiheissa vaatimusten dokumentointitapana. Vaatimuksia kartoitettaessa niistä pyrittiin jo karsimaan päällekkäisyyksiä ja ristiriitaisuuksia. Koska vaatimukset kerättiin pääosin haastatteluin, ei karsintaa ja ristiriitaisuuksien etsintää kuitenkaan voitu täysin toteuttaa vielä tässä vaiheessa.

## 6.4 Vaatimusten analysointi ja määrittely

Vaatimusten kartoituksen jälkeen taulukkoon kerätyt vaatimukset käytiin tarkasti läpi. Kartoitusvaiheessa kirjatusta yksittäisistä vaatimuksista paljastui useita tapauksia, jotka jaettiin kahdeksi tai useammaksi vaatimukseksi. Kartoitusvaiheen vaatimuksista löytyi jopa sellaisia toiminnallisuuksia, jotka järjestelmässä ehdottomasti tarvitaan, mutta joita ei vielä aiemmin ollut tunnistettu. Kartoitusvaiheessa vaatimuksia listattiin 60 kappaletta, ja analysointivaiheen myötä vaatimusten määrä nousi 97 kappaleeseen. Näistä kolme hylättiin tarpeettomina.

Vaatimukset jaettiin perinteisen kaksijakoisen mallin mukaan toiminnallisiin ja ei-toiminnallisiin vaatimuksiin. Jaottelun lisäksi vaatimukset ryhmiteltiin taulukossa 2 listattuihin ryhmiin.

Taulukko 2. Vaatimusten ryhmittely

<i>TOIMINNALLISET</i>	<i>EI-TOIMINNALLISET</i>
Haku	Arkkitehtuuri
Hälytykset	Helppokäyttöisyys
Järjestelmä	Luotettavuus
Kalenteri	Rajapinnat
Raportointi	Saatavuus
Resurssit	Tietoturva
Sopimukset	Ylläpidettävyys
Tehtävät	
Uutissyöte	
Yleiset	
Ylläpito	

Kartoitusvaiheessa vaatimukset kirjattiin muistiin vapaamuotoisesti, ja siitä syystä eri lähteistä kerätyt vaatimukset erosivat kirjoitusasultaan. Jokainen vaatimus kirjoitettiin vaatimusmuotoon, niin että ne kuvaavat, mitä järjestelmän tulee tehdä tai mitä käyttäjän tulee voida järjestelmällä tehdä. Esimerkiksi kartoitusvaiheessa kirjatusta vaatimuksesta, *resurssien allokointi tehtäville*, muodostui lopulliseen vaatimusmäärittelyyn vaatimus: *Tietyillä käyttäjäreoleilla tulee voida varata henkilöresurssi tehtävälle. Tällöin järjestelmä merkitsee tehtävän automaattisesti henkilön työjonoon.*

Vaatimusten priorisointiin olisi tavallisesti hyvä osallistua kaikkien vaatimuksia esittäneiden tahojen. Tässä tapauksessa suuri osa vaatimuksista pystyttiin johtamaan syksyllä 2013 pidetyistä ryhmätöistä, mutta käytävissä olevasta dokumentaatiosta ei selvinnyt vaatimusten tai tarpeiden esittäjää. Tästä syystä vaatimusten priorisointiin osallistui vain osa vaatimuksia esittäneistä henkilöistä.



Priorisointi toteutettiin kaksiosaisena. Ensimmäisessä vaiheessa kartoitetut ja analysoidut vaatimukset lähetettiin kullekin henkilölle, henkilökohtaisesti priorisoitavaksi. Mikäli ensimmäisessä vaiheessa annetut prioriteetit olivat yksimielisiä, kirjattiin kyseinen prioriteetti vaatimuksen prioriteetiksi. Toisessa vaiheessa priorisointia tarkennettiin työpajassa. Tässä työpajassa käsiteltiin vain ne vaatimukset, joiden osalta ensimmäisessä vaiheessa oli ristiriitoja. Näin työpajaan tarvittavaa aikaa saatiin lyhennettyä.

Vaatimukset dokumentoitiin määrittelyn yhteydessä jo kartoitusvaiheessa käytettyyn Excel-taulukkoon. Taulukkoa muutettiin eri vaiheiden tarpeiden mukaisesti esimerkiksi lisäämällä priorisointisarakkeita ja ryhmittelyjä. Vaatimustaulukon lisäksi vaatimusmäärittelyn yhteydessä tuotettiin vaatimusdokumentti. Lopullisessa vaatimustaulukossa jokaisesta vaatimuksesta esitettiin taulukossa 3 kuvatut tiedot. Vaatimusdokumentti, vaatimustaulukko ja erillinen raportointia koskeva dokumentti ovat tämän opinnäytetyön liitteinä.

Taulukko 3. Vaatimustaulukon sisältämät tiedot

<i>TIETO</i>	<i>KUVAUS</i>
<i>ID</i>	Vaatimuksen yksilöivä tunniste; juokseva numero
<i>LUOKKA</i>	Toiminnallinen tai ei-toiminnallinen
<i>RYHMÄ</i>	Vaatimuksen ryhmittelytieto; ryhmät on kuvattu taulukossa 2
<i>VAATIMUS</i>	Vaatimuksen kuvaus
<i>LÄHDE</i>	Vaatimuksen esittänyt taho, kuten henkilö, ryhmä tai asiakirja
<i>PÄIVÄMÄÄRÄ</i>	Vaatimuksen kirjaamispäivä
<i>PRIORITEETTI</i>	Vaatimuksen prioriteetti: <ul style="list-style-type: none"> <li>- Kriittinen (Vaatimukset ovat pakollisia ja oleellisia järjestelmän toiminnan kannalta; ilman niitä järjestelmä ei ole käyttökelpoinen)</li> <li>- Korkea (Vaatimukset ovat tärkeitä, mutta eivät puuttuessaan estä järjestelmän perustoimintaa)</li> <li>- Matala (Vaatimukset tuovat lisäarvoa järjestelmälle, mutta eivät haittaa suuresti puuttuessaan; tehdään jos resursseja riittää.)</li> </ul>
<i>KOMMENTIT</i>	Vapaa kommenttikenttä vaatimukselle

## 7 KEHITTÄMISEHDOTUKSIA VAATIMUSTEN HALLINTAAN

Nykyisin toimeksiantajan eri ohjelmistoprojekteissa vaatimusmäärittely tehdään erilaisin menetelmin. Vaatimusmäärittelyn lopputuloksena valmistuu erilaisia dokumentteja, joissa vaatimukset esitetään eri tavoin.

Kuten luvussa 5 kuvataan, ei vaatimusmäärittelylle ole olemassa yhtä oikeaa mallia, vaan jokaisen organisaation tulisi määritellä oma prosessinsa vaatimusten määrittelyä ja hallintaa varten. Koska vaatimusmäärittely on osa laajempaa vaatimustenhallinnan käsitettä, tulisi prosessi määritellä huomioiden vaatimusten koko elinkaari, eikä keskittyä pelkkään määrittelyvaiheeseen.

Ei-toiminnallisten vaatimusten osalta moniin ohjelmistoprojekteihin pätee monet samat vaatimukset. Esimerkiksi tietoturvaan ja toimintaympäristöön liittyvistä vaatimuksista kannattaisikin luoda valmis malli vaatimusmäärittelyä varten, joka voidaan pienellä työllä liittää uusiin vastaaviin vaatimusmäärittelyihin. Näin voidaan pienentää riskiä jonkin tärkeän vaatimuksen unohtamiselle ja helpottaa vaatimusten kartoitustyötä.

Vaatimusmäärittelyä varten kannattaisi luoda myös valmiit dokumenttipohjat, joissa kuvataan vähintään minimitaso vaatimusmäärittelyn lopputuloksille. Näin voidaan helpottaa vaatimusmäärittelyä tuottavan henkilöstön ymmärrystä työn edistymisestä suhteessa odotettuun lopputulokseen sekä parantaa valmiin määrittelyn luettavuutta. Kun vaatimusmäärittelyn lopputuloksena saadaan aina tietty vähimmäissisältö, tuotetaan samalla myös vaatimustenhallintaa varten riittävän laadukasta sisältöä.

Vaatimusten priorisointia varten olisi hyvä myös määritellä selkeät kriteerit. Monesti yksittäisten vaatimusten merkitys priorisointivaiheessa korostuu ja lopulliseen vaatimusmäärittelyyn päätyy turhaan liian korkean prioriteetin vaatimuksia. Pakollisten vaatimusten korkealla määrällä saadaan tyypillisesti suotta kasvatettua hankkeen budjettia ja pahimmillaan huomio kiinnittyy myöhemmissä vaiheissa väärin asioihin.

Vaatimustenhallintaa varten tulisi ottaa käyttöön myös sitä tukeva tietojärjestelmä. Vaatimusten ylläpito erillisissä dokumenteissa ja taulukoissa on työlästä, ja siitä johtuen vaatimustenhallinnan taso tyypillisesti laskee tai unohtuu kokonaan ohjelmiston pitkän elinkaaren aikana. Vaatimustenhallinta kannattaa mahdollisuuksien mukaan toteuttaa samalla järjestelmällä, jolla testausta hallitaan. Näin vaatimusten täyttymistä voidaan helposti seurata testauksen edetessä, ja vaatimukset ja niitä todentavat testitapaukset saadaan helposti ja visuaalisesti linkitettyä toisiinsa.

## 8 YHTEENVETO

Opinnäytetyön tavoitteena oli toteuttaa vaatimusmäärittely Oikeusrekisterikeskuksen IT-palvelutuotannon hallintaa tukevalle tietojärjestelmälle. Koska ennen opinnäytetyön aloittamista järjestelmälle ei ollut tehty kattavaa tarpeiden tunnistusta, tuli ne tunnistaa vaatimusten kartoittamisen yhteydessä. Jo prosessin alkuvaiheessa huomattiin tarpeiden määrän olevan oletettua suurempi, ja työn jouduttamiseksi kokonaisuutta rajattiin kiireellisimpään tarpeeseen. Ennen järjestelmän hankinnan aloittamista kannattaakin vielä varmistaa, voidaanko hankinnassa huomioida myös muiden tarpeiden täyttäminen edes joltain osin.

Tarpeita ja vaatimuksia kartoitettiin pääosin haastatteluin ja niitä täydentävin työpajoin sekä olemassa olevaa dokumentaatiota tutkimalla. Tutkimusmenetelmänä käytetty tapaustutkimus osoittautui hyväksi menetelmäksi vaatimusmäärittelyä toteuttamiseksi. Suurin haaste vaatimusten kartoittamisessa oli sopivien aikataulujen löytäminen, koska monet prosessiin osallistuneet henkilöt ovat hyvin kiireisiä. Jatkossa vaatimusmäärittelyyn kannattaakin varata reilusti kalenteriaikaa nimenomaan kaikille sopivien aikojen löytämiseksi.

Työssä tutustuttiin myös tietojärjestelmien hankintaan ja ohjelmistotuotannon menetelmiin sekä vaatimusmäärittelyn teoriaan. Vaatimusmäärittelyä tarkasteltiin myös ketterän ohjelmistokehityksen näkökulmasta ja käytiin läpi sen merkittävimpiä eroja perinteiseen ohjelmistokehitykseen verrattuna.

Lisäksi työn teoreettisen taustan ja käytännön osuudessa tehtyjen havaintojen perusteella tunnistettiin joitakin kehittämissuhteita toimeksiantajan oman vaatimusten hallinnan kehittämiseksi.

## LÄHTEET

- Agile Alliance. 2001. Agile Manifesto. Viitattu 1.4.2014.  
<http://agilemanifesto.org/>
- Eriksson, P & Koistinen, K. 2005. Monenlainen tapaustutkimus. Kuluttajatutkimuskeskuksen julkaisusarja 4:2005. Helsinki: Kuluttajatutkimuskeskus. Viitattu 7.3.2014.  
[http://www.ncrc.fi/files/4957/2005\\_04\\_verkkojulkaisu\\_tapaustutkimus.pdf](http://www.ncrc.fi/files/4957/2005_04_verkkojulkaisu_tapaustutkimus.pdf)
- Forselius, P. 2013. Onnistunut tietojärjestelmän hankinta. Helsinki: Talentum Media Oy.
- Granholt, G. 2013. Katsaus kompleksisten järjestelmien elinkaaren suunnitteluun. Espoo: VTT Technology 121. Viitattu 13.2.2014.  
<http://www.vtt.fi/inf/pdf/technology/2013/T121.pdf>
- Haikala, I. & Mikkonen, T. 2011. Ohjelmistotuotannon käytännöt. Helsinki: Talentum Media Oy.
- Haikala, I & Märijärvi, J. 2000. Ohjelmistotuotanto. Helsinki: Satku - kaup-pakaari Oyj.
- IEEE 830-1998. 1998. IEEE Recommended Practice for Software Requirements Specifications. Institute of Electrical and Electronics Engineers, Inc. (IEEE). Viitattu 11.2.2014.  
<http://www.math.uaa.alaska.edu/~afkjm/cs401/IEEE830.pdf>
- Immonen, J. 2002. Johdatus ohjelmistotuotantoon. Luentomoniste. Joensuun yliopisto. Viitattu 20.2.2014.  
[http://cs.joensuu.fi/~jimmonen/jot\\_moniste/](http://cs.joensuu.fi/~jimmonen/jot_moniste/)
- ISO/IEC/IEEE 29148:2011. 2011. Systems and software engineering - Life cycle processes - Requirements engineering. Institute of Electrical and Electronics Engineers, Inc. (IEEE). Viitattu 11.2.2014.  
<http://www.computer.org/portal/documents/82129/160549/IEEE+29148-2011.pdf>
- Julkisen hallinnon tietohallinnon neuvottelukunta (JUHTA). 2002. JHS 152 Prosessien kuvaaminen. Viitattu 28.2.2014.  
<http://docs.jhs-suositukset.fi/jhs-suositukset/JHS152/JHS152.pdf>
- Julkisen hallinnon tietohallinnon neuvottelukunta (JUHTA). 2009a. JHS 172 ICT-palvelujen kehittäminen: Esiselvitys. Viitattu 24.2.2014.  
<http://docs.jhs-suositukset.fi/jhs-suositukset/JHS172/JHS172.pdf>

Julkisen hallinnon tietohallinnon neuvottelukunta (JUHTA). 2009b. JHS 173 ICT-palvelujen kehittäminen: Vaatimusmäärittely. Viitattu 4.2.2014.  
<http://docs.jhs-suositukset.fi/jhs-suositukset/JHS173/JHS173.pdf>

Julkisen hallinnon tietohallinnon neuvottelukunta (JUHTA). 2011a. JHS 179 ICT-palvelujen kehittäminen: Kokonaisarkkitehtuurin kehittäminen. Viitattu 7.2.2014.  
<http://docs.jhs-suositukset.fi/jhs-suositukset/JHS179/JHS179.pdf>

Julkisen hallinnon tietohallinnon neuvottelukunta (JUHTA). 2011b. JHS 179 ICT-palvelujen kehittäminen: Kokonaisarkkitehtuurin kehittäminen. Liite 1. Viitattu 19.2.2014.  
[http://docs.jhs-suositukset.fi/jhs-suositukset/JHS179\\_liite1/JHS179\\_liite1.pdf](http://docs.jhs-suositukset.fi/jhs-suositukset/JHS179_liite1/JHS179_liite1.pdf)

Järvinen, P & Järvinen, A. 1995. Tutkimustyön metodeista. Tampere: Opinpa- ja Oy.

Kosola, J. 2013. Vaatimustenhallinnan opas. Maanpuolustuskorkeakoulun julkaisusarja 5 No 12. Helsinki: Maanpuolustuskorkeakoulu.

Kotonya, G. & Sommerville, I. 1998. Requirements Engineering. Processes and Techniques. Chichester: John Wiley & Sons Ltd.

Laki Oikeusrekisterikeskuksesta 625/2012. 16.11.2012.

Leffingwell, D. 2011. Agile Software Requirements. Lean Requirements Practices for Teams, Programs, and the Enterprise. Pearson Education, Inc.

Oikeusrekisterikeskus. 2014. Viitattu 11.3.2014.  
<http://www.oikeusrekisterikeskus.fi>

Paakki, J. 2011. Ohjelmistojen vaatimusmäärittely. Luentokalvot. Helsingin yliopisto. Viitattu 22.3.2014.  
<http://www.cs.helsinki.fi/u/paakki/Vaatimus-11-Luentokalvot-1.pdf>

Pohjonen, R. 2002. Tietojärjestelmien kehittäminen. Jyväskylä: Docendo Finland Oy.

Salo, O. 2006. Enabling Software Process Improvement in Agile Software Development Teams and Organisations. Espoo: VTT Publications. Viitattu 28.3.2014.  
<http://www.vtt.fi/inf/pdf/publications/2006/P618.pdf>

Sommerville, I. 2011. Software Engineering: Ninth Edition. Boston: Addison-Wesley.

Standish Group. 1995. Chaos Report. Viitattu 18.2.2014.  
<http://www.projectsmart.co.uk/docs/chaos-report.pdf>

Standish Group. 2013. Chaos Manifesto. Viitattu 18.2.2014.  
<http://www.versionone.com/assets/img/files/ChaosManifesto2013.pdf>

TTL Tietotekniikan liitto ry. 2005. Tietojärjestelmän hankinta. Helsinki: Talentum Media Oy.

TIEKE Tietoyhteiskunnan kehittämiskeskus ry. 2005. Tietotekniikkahankinnat – hankintaprosessi. Viitattu 19.2.2014.  
<http://www.tieke.fi/display/tiehan/Hankintaprosessi>

Tietojärjestelmien hankinta Suomessa. 2013. Celkee Oy, Tietotekniikan liitto ry ja Ohjelmistoyrittäjät ry. Viitattu 18.2.2014.  
<http://www.ttlry.fi/sites/ttl.ttlry.mearra.com/files/Tietoj%C3%A4rjestelmien%20hankinta%20Suomessa%202013.pdf>

Wieggers, K. & Beatty, J. 2013. Software Requirements 3. Redmond: Microsoft Press. Viitattu 28.2.2014. Saatavissa Safari Books Online-tietokannassa:  
<http://proquestcombo.safaribooksonline.com/book/software-engineering-and-development/software-requirements/9780735679658/software-requirements/id564448?uicode=hamep>

Virtuaaliammattikorkeakoulu. 2007. Ylemmän AMK-tutkinon metodifoorumi. Case-tutkimus. Viitattu 7.3.2014.  
<http://www2.amk.fi/digma.fi/www.amk.fi/opintojaksot/0709019/1193463890749/1193464144782/1194348546586/1194356433452.html>

## VAATIMUSDOKUMENTTI

## 1 JOHDANTO

## 1.1 Järjestelmän tausta ja tarkoitus

Oikeusrekisterikeskus on oikeusministeriön hallinnonalaan kuuluva virasto, jonka yhtenä tehtävänä on huolehtia oikeusministeriön hallinnonalan tietojärjestelmien ylläpidosta ja kehittämisestä yhteistyössä hallinnonalan virastojen kanssa siten kuin palvelusopimuksissa on sovittu.

Tietojärjestelmäpalvelutuotantoon liittyvät työtehtävät on organisoitu kolmelle eri toimialalle. Viranomaistoiminta -toimialalla toimii vaalitietojärjestelmän ylläpito ja tuki. Asiakkuuksien, Talouden ja palveluiden hallinta -toimialalla toimii asiakas- ja palvelupäälliköt sekä tuotannon tukiryhmä. Järjestelmäkehitys- ja projektipalvelut toimialalla tuotetaan tietojärjestelmien kehittämistehtävät sekä projektitoiminta.

Toiminnanohjausjärjestelmän tärkein tavoite on parantaa toiminnan läpinäkyvyyttä ja hallittavuutta sekä tukea tehtävien priorisointia koko asiakaskunnan kesken.

- Sopimukset ja niihin liittyvät tehtävät tulee olla kaikkien Oikeusrekisterikeskuksen työntekijöiden nähtävillä.
- Sopimusvelvoitteiden täyttymistä pitää voida seurata reaaliaikaisesti.
- Eri sopimuksiin liittyvät tehtävät tulee saada prioriteetteineen näkyviin yhtenä kokonaisuutena.

## 1.2 Käytetyt lyhenteet ja sanasto

Lyhenne	Kuvaus
CSV	Comma Separated Value. Tekstimuotoinen tiedosto, jossa esitetään pilkulla tai muulla välimerkillä eroteltuna taulukkomuotoista tietoa.
FTP	File Transfer Protocol. Tiedonsiirtomenetelmä, jonka avulla siirretään tiedostoja kahden tietokoneen (asiakas ja palvelin) välillä.
PDF	Portable Document Format. Tulostimen ja näytön tarkkuudesta riippumaton kansainvälisesti standardoitu tiedostomuoto, jota käytetään laajasti sähköiseen julkaisuun.
JHS	JHS-suositukset ovat valtion- ja kunnallishallinnon tietohallinnolle annettuja suosituksia, menettelytapoja tai ohjeita. JHS-suosituksia antaa Julkisen hallinnon tietohal-

	linnon neuvottelukunta JUHTA.
VAHTI	Valtionhallinnon tieto- ja kyberturvallisuuden johtoryhmä

### 1.3 Yleiskatsaus dokumenttiin

Tämän dokumentin tarkoitus on kuvata perustiedot Oikeusrekisterikeskuksen toiminnanohjausjärjestelmästä.

Dokumentin toisessa luvussa annetaan yleiskuvaus järjestelmästä, sen toimintaympäristöstä ja käyttäjistä. Kolmannessa luvussa kuvataan järjestelmän keskeisimpiä käsitteitä ja niiden suhteita toisiinsa. Neljännessä luvussa kuvataan järjestelmän tärkeimpiä toimintoja. Viidennessä luvussa kuvataan järjestelmän ulkoisia ohjelmistoliittymiä. Kuudennessa luvussa listataan joitakin muita ei-toiminnallisia vaatimuksia ja lopuksi seitsemännessä luvussa rajoitteita.

Dokumentin ei ole tarkoitus olla tyhjentävä tai täydellinen kuvaus järjestelmästä vaan tässä kuvattua sisältöä tullaan täsmentämään hankinnan edetessä ja valitun teknologian sekä toimittajan valinnan jälkeen.

## 2 YLEISKUVAUS

### 2.1 Järjestelmän toimintaympäristö

Toiminnanohjausjärjestelmää tullaan käyttämään oikeusministeriön tietoverkosta. Myöhemmin toiminnallisuuden lisääntyessä voi tulla tarve käyttää järjestelmää myös internetin yli. Tällöin pääsy järjestelmään tulee toteuttaa suojattua yhteyttä käyttäen erikseen sallituista IP-osoitteista tai verkoista.

Järjestelmän tulee voida lähettää sähköpostia ja mahdollisesti integroitua Oikeusrekisterikeskuksen käytössä olevaan Lotus Notes -kalenterijärjestelmään.

### 2.2 Järjestelmän käyttäjät

Toiminnanohjausjärjestelmän käyttäjiä ovat Oikeusrekisterikeskuksen työntekijät (virkamiehet sekä konsultit).

Käyttäjät jakautuvat vähintään seuraaviin rooleihin:

Rooli	Kuvaus
Asiakaspäällikkö	Jokaiselle asiakkaalle on nimetty asiakaspäällikkö, joka vastaa sopimuksesta asiakkaan suuntaan.
Esimies	Jonkin järjestelmään määritellyn organisaation esimies. Esimies voi olla esimerkiksi ryhmän esimies, toimialan

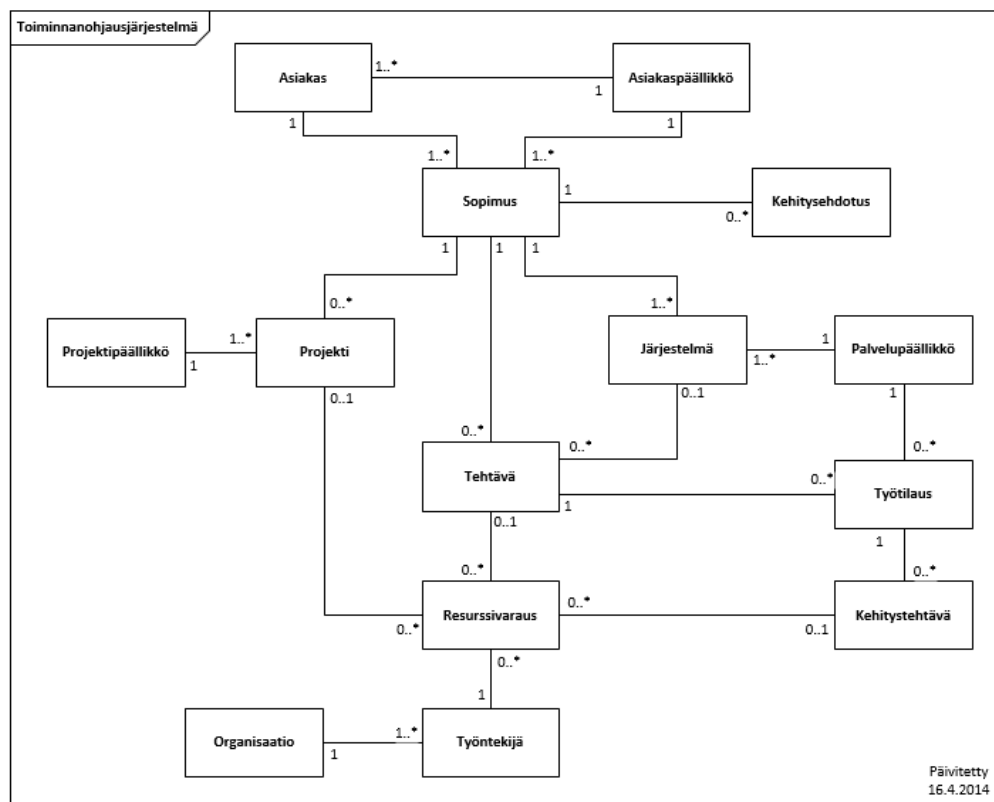


	johtaja tai Oikeusrekisterikeskuksen johtaja. Esimies on myös oman organisaationsa aliorganisaatioiden esimies.
Johto	Oikeusrekisterikeskuksen johtoryhmän jäsen
Palvelupäällikkö	Jokaiselle palvelusopimukselle on nimetty palvelupäällikkö, joka vastaa sopimukseen kuuluvista tietojärjestelmistä.
Projektipäällikkö	Jokaiselle projektille nimetään asettamispäätöksessä projektipäällikkö. Projektipäällikkö kuvaa projektin kokonaisvastuullista projektipäällikköä.
Pääkäyttäjä	Toiminnanohjausjärjestelmän pääkäyttäjä
Resurssimanageri	Henkilö, jonka tehtävänä on priorisoida Oikeusrekisterikeskuksen töitä ja jolla on oikeus siirtää henkilöresursseja kulloinkin tärkeimpänä katsomaansa tehtävään.
Työntekijä	Oikeusrekisterikeskuksen työntekijät, sekä virkamiehet että konsultit, joille voidaan osoittaa työtehtäviä järjestelmän kautta.

Käyttäjä voi kuulua yhteen tai useampaan rooliin.

## 3 KESKEISET KÄSITTEET

Tässä kuvataan toiminnanohjausjärjestelmän keskeiset käsitteet sekä niiden väliset yhteydet.



Käsite	Kuvaus
Asiakas	Asiakas, jonka kanssa palvelusopimus on sovittu tai tulosohtajaaja, jonka kanssa tulossopimus on sovittu. Asiakkaalle on aina nimetty asiakaspäällikkö ja asiakkaalla voi olla yksi tai useampi sopimus.
Asiakaspäällikkö	Jokaiselle asiakkaalle on nimetty asiakaspäällikkö, joka vastaa sopimuksesta asiakkaan suuntaan. Asiakaspäällikkö on Oikeusrekisterikeskuksen työntekijä. Asiakaspäällikköllä voi olla yksi tai useampi asiakas ja yksi tai useampi sopimus. Tulossopimuksen tapauksessa asiakkaana on tulosohtajaaja ja asiakaspäällikkönä tulossopimuksen vastuuhenkilö.
Järjestelmä	Suuri osa palvelusopimusten tehtävistä liittyy nykyjärjestelmien kehittämiseen ja kunnossapitoon. Järjestelmästä sovitaan aina palvelusopimuksella ja jokaiselle

	järjestelmälle on nimetty yksi palvelupäällikkö. Yhteen järjestelmään voi liittyä useampia tehtäviä.
Kalenteri	Kalenteria ei ole kuvattu edellä olevassa käsitelmässä. Kalenteri on järjestelmän tarjoama näkymä.
Kehitysehdotus	Palvelusopimuksen kehitysehdotuksiin voidaan kirjata esimerkiksi tarpeita tulevien vuosien suunnitteluun. Kehitysehdotus liittyy aina yhteen sopimukseen, vaikka ehdotus voi koskea esimerkiksi sopimukseen kuuluvaa järjestelmää.
Kehitystehtävä	Kehitystehtävä on tehtävän erikoistapaus. Kehitystehtävällä toteutetaan työtilauksella tilattu työ. Kehitystehtävä liittyy aina yhteen työtilaukseen. Kehitystehtävään sidotaan resurssit resurssivarauksin.
Organisaatio	Organisaatiot ovat hierarkkisia, ja niillä kuvataan vähintään Oikeusrekisterikeskuksen linjaorganisaatiota. Organisaatorakennetta tarvitaan esimiestiedon ja hierarkian kuvaamiseen esimerkiksi käyttöoikeuksia varten. Oikeusrekisterikeskuksen organisaatio on tällä hetkellä kolmitasoinen. Jokaisella organisaatiolla on yksi esimies. Organisaatioon kuuluvan työntekijän esimiehinä ovat kyseisen organisaation esimies sekä kaikkien yläorganisaatioiden esimiehet.
Palvelupäällikkö	Palvelupäällikkö vastaa yhden tai useamman järjestelmän toiminnasta. Palvelupäällikkö on Oikeusrekisterikeskuksen työntekijä. Palvelupäällikkö tilaa sisäisesti järjestelmään sovittuja kehitystehtäviä.
Projekti	Palvelusopimuksilla voidaan sopia sopimuskauden aikana tehtävistä projekteista. Projektin ei tarvitse alkaa tai loppua sopimuskauden aikana. Projekti-käsitettä voidaan käyttää myös hankkeista. Projektille on aina nimetty yksi vastuullinen projektipäällikkö. Projektin tehtäviä ei ylläpidetä järjestelmässä, mutta projektille nimettyjen työntekijöiden resurssivaraukset tulee kirjata järjestelmään.
Projektipäällikkö	Jokaiselle projektille nimetään asettamispäätöksessä yksi projektipäällikkö, joka vastaa projektista kokonaisuutena. Projektipäällikkö kuvaa siis projektin kokonaisvastuullista projektipäällikköä, eikä esimerkiksi hankkeissa jokaista projektipäällikköä. Projektipäällikkö-käsitettä voidaan käyttää myös esimerkiksi hankkeiden päälliköistä, eikä käsite ole sidoksissa henkilön virkänimikkeeseen tai rooliin linjaorganisaatiossa. Projektipäälliköllä voi olla vastuullaan yksi tai useampia projekteja.
Resurssivaraus	Kehitystehtävän toteuttamiseksi varataan yhden tai use-

	<p>amman työntekijän työaika. Resurssivarauksella varataan yhden työntekijän työaikaan yhteen kehitystehtävään määritellyksi ajaksi. Yksi kehitystehtävä voidaan toteuttaa yhdellä tai useammalla resurssivarauksella. Projekteihin tehdään myös resurssivarauksia. Projektien tehtävien hallinta ei tässä vaiheessa kuulu järjestelmän toiminnallisuuksiin, joten niiden tehtäviä ja työtilauksia ei järjestelmässä käsitellä. Työntekijöiden resurssivaraukset tulee kuitenkin kirjata järjestelmään käytettävissä olevan kapasiteetin seuraamiseksi. Myös tulossopimuksen tehtäviin voidaan varata resursseja suoraan ilman työtilauksia.</p>
Sopimus	<p>Toiminnanohjausjärjestelmän toiminnot keskittyvät Oikeusrekisterikeskuksen palvelusopimusten sekä tulossopimuksen hallintaan. Sopimuksella on aina yksi asiakaspäällikkö ja yksi asiakas. Sopimukseen voi liittyä useita kehitysehdotuksia. Sopimuksella sovitaan projekteista ja/tai tehtävistä ja/tai järjestelmistä.</p>
Tehtävä	<p>Sopimuksissa sovitaan tehtävistä, jotka Oikeusrekisterikeskus kyseisen sopimuskauden aikana toteuttaa. Tehtävät liittyvät tavallisesti johonkin järjestelmään, mutta voivat olla myös suoraan sopimukseen liittyviä tehtäviä. Palvelupäällikkö tekee palvelusopimuksessa sovittujen tehtävien täyttämiseksi työtilauksia, joilla tehtäviä täytetään. Muiden tehtävien osalta työtilauksia ei tehdä, vaan niille sovitaan erikseen resurssit ja tehdään resurssivaraukset. Esimerkkejä tällaisista tehtävistä on tulossopimuksen tehtävät sekä käyttäjän tai tämän esimiehen lisäämät tehtävät. Tehtävät voivat olla hierarkisia eli sopimuksella sovittuun tehtävään voi liittyä alitehtäviä.</p>
Työntekijä	<p>Oikeusrekisterikeskuksen työntekijä, jonka työaika varataan työtilauksen tai projektisopimuksen perusteella tehtäviin. Työaika varataan resurssivarauksin. Työntekijä kuuluu yhteen organisaatioon, jonka kautta hänellä on yksi tai useampi esimies. Myös Asiakaspäällikkö, Palvelupäällikkö ja Projektipäällikkö ovat Työntekijöitä.</p>
Työtilaus	<p>Palvelusopimuksessa sovitut tehtävät toteutetaan eri tavoilla, joilta palvelupäällikkö tilaa työt. Yksi palvelusopimuksen tehtävä jakaantuu tyypillisesti useampiin työtehtäviin, joten yhden tehtävän täyttämiseksi tehdään monesti useampi työtilaus. Työtilausten perusteella luodaan kehitystehtäviä, joille allokoitetaan niitä toteuttavat resurssit. Yksi työtilaus voidaan täyttää yhdellä tai useammalla kehitystehtävällä.</p>

## 4 TOIMINTOKUVAUS

Tässä luvussa kuvataan lyhyesti järjestelmän tärkeimpiä toimintoja. Kuvaus ei kata kaikkia järjestelmän toiminnallisuuksia, eikä kuvaus myöskään ole täydellinen. Kuvauksen on tarkoitus antaa yleiskuva järjestelmän tärkeimmistä toiminnoista, ja toiminnallisuuksia tullaan tarkentamaan teknologian ja toimitajan valinnan jälkeen.

		Asiakaspäällikkö	Esimies	Johito	Palvelupäällikkö	Projektipäällikkö	Resurssimanageri	Työntekijä	Paikayttäjä
Haku	Hakee tietoja	x	x	x	x	x	x	x	x
	Valitsee uutissyötteeseen	x	x	x	x	x	x	x	x
	Tallentaa hakutuloksen tiedostoon	x	x	x	x	x	x	x	x
Raportointi	Ajaa valmiin raportin	x	x	x	x	x	x	x	x
	Ajaa valitsemansa tiedot sisältävän raportin	x	x	x	x	x	x	x	x
	Tallentaa raportin tiedostoon	x	x	x	x	x	x	x	x
Sopimusten käsittely	Lisää sopimuksen	x		x	x				x
	Muokkaa sopimusta	x		x	x				x
	Poistaa sopimuksen			x					x
	Lisää kehitysehdotuksen	x	x	x	x	x	x	x	x
Tehtävien käsittely	Lisää tehtävän	x	x 3)	x	x		x	x 1)	x
	Muuttaa tehtävän prioriteettia		x	x	x		x		x
	Päivittää tehtävään liittyviä tietoja	x		x	x			x	x
	Lisää tehtävän työnooon		x	x			x	x	x
	Poistaa tehtävän	x 2)	x 2)	x	x 2)		x	x 2)	x
Työtilaukset	Tekee työtilauksen				x				x
	Muokkaa työtilausta				x		x		x
	Poistaa työtilauksen				x		x		x
	Muuttaa kehitystehtäväksi		x				x		x
Resurssien hallinta	Katselee resurssivarauksia	x	x	x	x	x	x	x	x
	Etsii vapaan resurssin	x	x	x	x	x	x	x	x
	Tekee resurssivarauksen		x	x			x		x
	Muokkaa resurssivarausta		x	x			x		x
	Poistaa resurssivarauksen		x	x			x		x
Ylläpito	Muokkaa organisaatorakennetta								x
	Siirtää työntekijän eri organisaatioon			x					x
	Muuttaa työntekijän roolia		x	x					x
	Lisää järjestelmään käyttäjän								x
	Muuttaa työntekijän tietoja		x	x				x	x
Muuttaa asiakkaan tietoja	x								x

1) Ainoastaan omaan työnooonsa  
2) Ainoastaan itse luomiaan tehtäviä  
3) Ainoastaan oman organisaationsa työntekijän työnooon

Edellä olevassa matriisissa on kuvattu järjestelmän tärkeimmät toiminnot sekä alustava näkemys käyttäjärooleista, joilla toimintoja tulee voida käyttää. Roo-leja ja käyttöoikeuksia tullaan tarkentamaan hankinnan edetessä, ja niitä tulee myös voida ylläpitää järjestelmän käyttöliittymältä.

#### 4.1 Haku

Käyttäjä voi hakea järjestelmän hakutoiminnolla kaikkea järjestelmään tallennettua tietoa. Haun pitää ulottua myös järjestelmään tallennettujen liitetiedostojen metatietoihin. Hakutuloksesta käyttäjä voi valita tiedon tai liitetiedoston omaan uutissyötteeseensä, jotta sen muutosten seuranta helpottuu. Hakutulos tulee voida myös viedä helposti esimerkiksi ulkoiseen tekstinkäsittely- tai taulukkolaskentasovellukseen sekä tallentaa käyttäjän työasemalle.

Hakutoiminto tuo käyttäjän nähtäville vain ne tiedot ja liitetiedostot, joihin käyttäjällä on käyttöoikeus.

#### 4.2 Raportointi

Järjestelmän tulee sisältää helppokäyttöinen raportointitoiminnallisuus, jolla käyttäjä voi ajaa valmiita järjestelmään tallennettuja raportteja sekä valita itse raportille kerättävät tiedot. Käyttäjän tulee voida valita ajanjakso, jolta raportti muodostetaan. Raportit tulee voida myös viedä helposti ulkoiseen tekstinkäsittely- tai taulukkolaskentasovellukseen sekä tallentaa käyttäjän työasemalle. Raporttien käyttöoikeudet tulee voida määrittellä järjestelmän käyttöliittymältä vähintään käyttäjän organisaation sekä käyttäjäroolien mukaan.

Järjestelmässä tulee olla vähintään seuraavat valmisraportit:

- Henkilöraportti
- Ryhmäraportti
- Johdon raportit
- Asiakasraportit
- Projekti- ja tehtäväraportti
- Järjestelmäraportti

Raportointia sekä edellä kuvattuja raportteja on kuvattu tarkemmin liitteessä 3 (Raportointi).

Raportointitoiminto tuo käyttäjän ajamalle raportille vain ne tiedot, joihin käyttäjällä on käyttöoikeus.

#### 4.3 Sopimusten käsittely

Järjestelmään tallennetaan oleellimmat tiedot Oikeusrekisterikeskuksen palvelusopimuksista sekä tulossopimuksista. Sopimukset ovat tietohallintopalvelutuotannon perusta ja kaikki toiminta liittyy sopimukseen. Jokaisen järjestelmän käyttäjän tulee voida liittää sopimukselle kehitysehdotuksia.

Lisää sopimuksen -toiminnolla käyttäjä voi lisätä järjestelmään uuden sopimuksen. Toiminto edellyttää pakollisten tietojen syöttämistä sopimuksen tallentamiseksi. Tietojen pakollisuus tulee olla muutettavissa pääkäyttäjän toimesta.

Muokkaa sopimusta -toiminnolla käyttäjä voi päivittää olemassa olevan sopimuksen tietoja. Toiminnolla pitää voida päivittää kaikkia sopimuksen tietoja.

Poistaa sopimuksen -toiminnolla käyttäjä poistaa koko sopimuksen. Sopimuksen poistaminen johtaa myös kaikkien siihen liittyvien muiden objektien poistamiseen. Toiminnon käyttöä tulee rajoittaa hyvin pienelle käyttäjäkunnalle.

Lisää kehitysehdotuksen -toiminnolla käyttäjä voi lisätä johonkin sopimukseen kehitysehdotuksen. Kehitysehdotus on vapaa tekstimuotoinen ehdotus, johon tulee voida liittää myös liitetiedostoja.

Sopimusten lisääminen ja ylläpito on rajoitettu käyttöoikeuksin tietyille käyttäjärooleille.

#### 4.4 Tehtävien käsittely

Tehtävä on toiminnanohjausjärjestelmässä monimuotoinen käsite. Tehtävä voi liittyä sopimukseen, järjestelmään tai työtilaukseen. Sopimuksen tehtävät voivat liittyä palvelusopimukseen tai tulossopimukseen. Järjestelmään liittyvät tehtävät voi olla suoraan palvelusopimuksella kirjattuja tehtäviä tai palvelupäällikön lisäämiä tehtäviä, joilla yhdessä täytetään palvelusopimuksen tehtäviä. Järjestelmien kehittämiseen liittyvistä tehtävistä palvelupäällikkö tekee sisäiset työtilaukset toiminnanohjausjärjestelmällä. Työtilaukset toteutetaan kehitystehtävillä, jotka luodaan tilausten perusteella. Tehtävät voivat olla myös hierarkisia eli tehtävä voi olla toisen tehtävän alitehtävä.

*Lisää tehtävän* -toiminnolla lisätään järjestelmään uusia tehtäviä. Toiminnon avulla eri käyttäjäroolissa toimiva käyttäjä voi lisätä erilaisia tehtäviä erilaisiin objekteihin. Esimerkiksi asiakaspäällikkö voi lisätä palvelusopimukselle uusia asiakkaan kanssa sovittuja tehtäviä. Jokainen käyttäjä voi lisätä myös itse järjestelmään tehtäviä. Tällöin tehtävä tulee kyseisen käyttäjän omaan työjonoon.

*Lisää tehtävän työjonoon* -toiminnolla jokainen käyttäjä voi itse lisätä tehtävän omaan työjonoonsa. Tiedyt käyttöoikeudet omaava käyttäjä voi myös määrittää tehtävän toisen työntekijän työjonoon.

*Poistaa tehtävän* -toiminnolla tehtävä poistetaan järjestelmästä. Mikäli tehtävään liittyy alitehtäviä tai muita tietoja, myös nämä tiedot poistuvat tehtävän poiston yhteydessä. Tehtävän poistaminen on rajoitettu käyttöoikeuksien tietyille käyttäjärooleille. Tämän lisäksi käyttäjä voi poistaa itse omaan työjonoonsa lisäämän työtehtävän käyttöoikeusrajoituksista huolimatta.

#### 4.5 Työtilaukset

Tällä toiminnolla palvelupäällikkö tilaa kehitystyötä Oikeusrekisterikeskuksen sisäisen tilaaja-tuottaja-mallin mukaisesti. Työtilaukset ohjautuvat kehityspäällikön työjonoon ja niiden perusteella luodaan kehitystiimin toteutettavaksi kehitystehtäviä. Työtilaukset tulee priorisoida ja niiden tallentamisen yhteydessä tulee syöttää riittävät tiedot kehitystehtävän lisäämiseksi ja toteuttamiseksi. Työtilaus tulee voida muuttaa suoraan kehitystehtäväksi tai palauttaa täydennettäväksi, mikäli sen tiedot ovat puutteelliset.

#### 4.6 Resurssien hallinta

Kaikkien järjestelmän käyttäjien tulee voida katsella resurssivarauksia. Varausista tulee nähdä ainakin vapaat ja varatut ajankohdat. Mikäli käyttäjällä ei ole käyttöoikeuksia resurssivaraukseen, ei hän saa nähdä sen sisältöä.

*Etsii vapaan resurssin* -toiminnolla käyttäjä voi etsiä tietylle ajanjaksolle vapaata resurssia. Toimintoa voidaan käyttää esimerkiksi resurssivarauksen tekemiseksi, mikäli käyttäjällä on siihen oikeudet.

Tietyillä käyttöoikeuksilla käyttäjälle tulee voida tehdä resurssivaraus. Varaus tehdään esimerkiksi projektisopimuksen mukaisten varausten lisäämiseksi, jotta vapaan resurssin etsintätoiminnolla saadaan tulokseksi todellinen tilanne. Työtilauksen perusteella tehtävään kehitystehtävään kiinnitetään resurssi(t) samalla toiminnolla. Resurssivarauksia tulee voida myös muokata jälkikäteen sekä poistaa.

#### 4.7 Ylläpito

Ylläpitotoiminnallisuudella tulee voida ylläpitää kaikkea järjestelmän sisäistä konfiguraatiota, kuten organisaatioita ja organisaatiorakenteita, asiakkaiden tietoja sekä järjestelmän käyttäjiä ja niiden oikeuksia.



## 5 JÄRJESTELMÄN ULKOISET LIITTYMÄT

### 5.1 Työasema

Käyttäjän on voitava siirtää ajamiensa raporttien sekä hakujen tulokset työasemallansa toimivaan tekstinkäsittely- tai taulukkolaskentajärjestelmään. Oikeusrekisterikeskuksen tällä hetkellä käytössä olevat toimisto-ohjelmistot ovat Microsoft Office 2010 sekä OpenOffice.org 3. Tietojen siirto tulee onnistua vähintään mainittuihin järjestelmiin.

Käyttäjän tulee myös voida tallentaa hakutulokset sekä raportit halutessaan käyttämälleen työasemalle. Raportit tulee voida tallentaa vähintään PDF-muotoisina tiedostoina ja hakutulokset vähintään CSV- tai PDF-muotoisina tiedostoina.

### 5.2 Suunta

Suunta on Oikeusrekisterikeskuksen talouden suunnittelujärjestelmä, jossa tehdään muun muassa toiminnan budjetointi sekä rahojen että henkilötyön osalta. Suunta tekee myös laskennat esimerkiksi Oikeusrekisterikeskuksen tekemän työn kustannusten osalta.

Koska palvelusopimusten työmääräsuunnitelmat kirjataan Suuntaan budjetoinnin yhteydessä, suunnitelmatiedot on järkevää noutaa integraation kautta toiminnanohjausjärjestelmään.

Suunnan integraatiot on toteutettu CSV-muotoisina FTP-tiedostosiirtoina Oikeusrekisterikeskuksen käyttämän integraatoratkaisun kautta.

### 5.3 Timme

Timme on Oikeusrekisterikeskuksen työajankirjausjärjestelmä. Timmen työaikakirjaukset siirtyvät automaattisesti talouden suunnittelujärjestelmä Suuntaan, jossa työajat kohdistuvat palvelusopimuksille.

Toiminnanohjausjärjestelmän tulee integroitua Timmeen työaikakirjausten sekä kirjauskohteiden siirtämiseksi. Kirjauskohteiden tulee olla järjestelmissä yhteneväiset, jotta toiminnanohjausjärjestelmässä nähdään tehtäviin käytetty työaika. Näin käytetyn työajan suhdetta suunniteltuun työmäärään voidaan myös tarkastella.

Timmen integraatiot on toteutettu CSV-muotoisina FTP-tiedostosiirtoina Oikeusrekisterikeskuksen käyttämän integraatoratkaisun kautta.

#### 5.4 Sähköposti

Järjestelmän on voitava lähettää sähköpostia. Järjestelmää ei tulla kuitenkaan liittämään Oikeusrekisterikeskuksen tavanomaisessa toimistokäytössä olevaan sähköpostijärjestelmään, vaan toiminnanohjausjärjestelmän tulee sisältää tarvittava toiminnallisuus.

### 6 JÄRJESTELMÄN EI-TOIMINNALLISET OMINAISUUDET

#### 6.1 Suorituskyky

Järjestelmän vasteajan tulee kaikkien käyttöliittymätoimintojen osalta olla korkeintaan 2 sekuntia. Vasteaika mitataan käyttäjän käyttöliittymällä antamasta käskystä järjestelmän kyselyyn tai käskyyn perustuvaan vastaukseen, kuten hakutulokseen. Raportoinnin osalta vasteaika saa kuitenkin olla korkeintaan 15 sekuntia käyttäjän käyttöliittymällä käynnistämästä raporttitoiminnallisuudesta valmiin raportin näyttämiseen.

Järjestelmää tulee alkuvaiheessa käyttämään noin 60 henkilöä, joista yhtäaikaista käyttäjiä arvioidaan olevan 20. Edellä kuvatut suorituskykyvaatimukset tulee täyttyä 20 yhtäaikaisen käyttäjän aiheuttamalla normaalikäyttöä vastaavalla kuormalla koko palveluajan.

#### 6.2 Saatavuus, tietoturvallisuus, varmuuskopiointi

Järjestelmän tulee täyttää tuotantokäytössä valtiohallinnon tietoturvallisuudesta annetun asetuksen 681/2010 mukaisen perustason tietoturvaluokituksen VAHTI-ohjeistuksen mukaisesti.

Järjestelmän tulee olla asiakkaan käytettävissä (palveluaika) vähintään suosituksessa JHS 174 kuvatun palvelutason P2 mukaisesti, arkisin klo 7–19 välisen ajan. Järjestelmän tulee täyttää palveluaikana suosituksessa JHS174 kuvatun käytettävyyden K2 tason (99,0 % käytettävyys, maksimikatko palveluaikana 4 tuntia).

Järjestelmän huoltokatkot tulee ensisijaisesti ajoittaa palveluajan ulkopuolelle, mutta vähintään klo 8–16:15 välisen ajan ulkopuolelle. Mikäli huoltokatko ajoitetaan palveluajalle, tulee tästä sopia asiakkaan kanssa etukäteen vähintään 2 viikkoa ennen aiottua huoltokatkoa.

Kaikki järjestelmän sisältämä tieto tulee varmuuskopioida. Varmuuskopiosta pitää pystyä tarvittaessa palauttamaan koko järjestelmän toiminta toiselle laitteelle. Varmuuskopiointi tulee järjestää, niin että se ei aiheuta järjestelmälle käyttökatkoa.

### 6.3 Ylläpidettävyys

Järjestelmän tulee perustua yleisesti käytössä olevaan teknologiaan, jolle löytyy vähintään kolme Suomessa toimivaa toimittajaa. Järjestelmän ylläpitoon tarkoitettu tekninen dokumentaatio tulee tuottaa suomen- tai englanninkielisenä.

## 7 OHJELMISTORAJOITTEET

Järjestelmän käyttöliittymän tulee olla selainpohjainen. Järjestelmän tulee toimia vähintään Microsoft Internet Explorer 8:lla ja sitä uudemmilla versioilla, eikä sen käyttö saa vaatia työasemaan asennettavia lisäkomponentteja, kuten selaimen laajennuksia.

## VAATIMUSTAULUKKO

versio 1.0					
ID	Luokka	Ryhmä	Vaatus	Tila	Prioriteetti
50	Ei-toiminnalliset	Arkkitehtuuri	Käyttöliittymän tulee olla selainpohjainen	Hyväksytty	1 - kriittinen
51	Ei-toiminnalliset	Arkkitehtuuri	Järjestelmän tulee tukea vähintään seuraavia selainohjelmistoja: Internet Explorer 8 ja sitä uudemmat versiot	Hyväksytty	1 - kriittinen
52	Ei-toiminnalliset	Arkkitehtuuri	Järjestelmän käyttöliittymän on toimittava ilman työasemaan tai selaimen asennettavia lisäosia	Hyväksytty	1 - kriittinen
54	Ei-toiminnalliset	Arkkitehtuuri	Järjestelmän on tuettava yleisimpiä tekstinkäsittely- ja taulukkolaskentavälineitä ja formaatteja (Microsoft Office ja Open Office, PDF, CSV) sekä yleisimpiä kuvaformaatteja, kuten esimerkiksi TIFF, JPG, PNG ja GIF	Hyväksytty	1 - kriittinen
55	Ei-toiminnalliset	Arkkitehtuuri	Raportit ja hakutulokset on voitava siirtää yleisimpiin tekstinkäsittely- ja taulukkolaskentavälineisiin (Microsoft Office ja Open Office)	Hyväksytty	1 - kriittinen
59	Ei-toiminnalliset	Arkkitehtuuri	Järjestelmä on ominaisuuksiensa (esimerkiksi teknisten ja kaupallisten) puolesta voitava virtualisoida Vmware -virtuaalipalvelimille ilman merkittäviä virtualisoinnista aiheutuvia työ- ja lisenssikustannuksia	Hyväksytty	2 - korkea
61	Ei-toiminnalliset	Arkkitehtuuri	Järjestelmän on voitava lähettää sähköpostia SMTP-protokollaa käyttäen	Hyväksytty	2 - korkea
73	Ei-toiminnalliset	Arkkitehtuuri	Järjestelmään tulee voida tallentaa liitetiedostoja ilman formaattirajoituksia	Hyväksytty	2 - korkea
74	Ei-toiminnalliset	Arkkitehtuuri	Järjestelmän tulee mahdollistaa tietosisällön muokkaaminen samanaikaisesti useamman käyttäjän toimesta	Hyväksytty	3 - matala
77	Ei-toiminnalliset	Arkkitehtuuri	Järjestelmän tulee mahdollistaa integraatiot muihin järjestelmiin vähintään web services -rajapintaa sekä ftp-tiedonsiirtoa käyttäen	Hyväksytty	2 - korkea
79	Ei-toiminnalliset	Arkkitehtuuri	Kaiken järjestelmän sisällön tulee olla tulostettavissa helppoluokukseen muotoon	Hyväksytty	2 - korkea
47	Toiminnalliset	Haku	Käyttäjän tulee voida hakea käyttöoikeuksiensa puitteissa kaikkea järjestelmään tallennettua tietoa haku-toiminnallisuutta käyt-	Hyväksytty	1 - kriittinen

			täen.		
86	Toiminnalliset	Haku	Käyttäjän tulee voida valita hakutuloksesta tieto omaan uutissyöteeseensä	Hyväksytty	3 - matala
88	Toiminnalliset	Haku	Käyttäjän tulee voida tallentaa hakutulos työasemalleen vähintään CSV- tai PDF-muotoisena tiedostona.	Hyväksytty	2 - korkea
53	Ei-toiminnalliset	Helppokäyttöisyys	Järjestelmän tulee näyttää käyttäjälle selkeästi missä hän on ja mitä hän voi tehdä	Hyväksytty	2 - korkea
78	Ei-toiminnalliset	Helppokäyttöisyys	Käyttöliittymän tekstikokoa tulee voida muuttaa	Hyväksytty	2 - korkea
93	Ei-toiminnalliset	Helppokäyttöisyys	Järjestelmän linkkien tulee olla selkeästi nimetty, jotta käyttäjä tietää minne linkki johtaa	Hyväksytty	2 - korkea
94	Ei-toiminnalliset	Helppokäyttöisyys	Selaimen Eteen tai Taakse siirtymävalintojen käyttäminen ei saa vaarantaa sovelluksen toimintaa ja tiedon eheyttä	Hyväksytty	2 - korkea
95	Ei-toiminnalliset	Helppokäyttöisyys	Järjestelmän jokaisesta näytöstä on mahdollista palata vähintään järjestelmän pääsivulle sekä yleisimmistä toiminnoista myös edelliselle sivulle	Hyväksytty	2 - korkea
97	Ei-toiminnalliset	Helppokäyttöisyys	Käyttöliittymässä ei näytetä sellaisia valintoja, joihin käyttäjällä ei ole käyttöoikeuksia	Hyväksytty	2 - korkea
15	Toiminnalliset	Hälytykset	Järjestelmään tulee voida tallentaa automaattisia hälytyksiä. Hälytykset voivat liittyä esimerkiksi: - valmistumattomiin tehtäviin määritellyn aikarajan umpeutumiseen - tehtäviin, joita ei määriteltyn aikaan ole päivitetty - henkilölle allokoituun uuteen tehtävään	Hyväksytty	1 - kriittinen
31	Toiminnalliset	Hälytykset	Hälytyksille tulee voida määrittellä hälytyskohtaisesti yksi tai useampi sähköpostiosoite, johon järjestelmä hälytyksen tapahtuessa lähettää automaattisesti hälytysviestin	Hyväksytty	3 - matala
32	Toiminnalliset	Hälytykset	Mikäli hälytykseen liittyy yksi tai useampi käyttäjätieto, lähettää järjestelmä hälytysviestin myös tämän käyttäjän tiedoissa määriteltyn sähköpostiosoitteeseen.	Hyväksytty	2 - korkea
46	Toiminnalliset	Hälytykset	Käyttäjän tulee voida asettaa omia hälytyksiä	Hyväksytty	2 - korkea
45	Toiminnalliset	Järjestelmä	Kullakin järjestelmällä tulee olla määriteltynä vastuhenkilö	Hyväksytty	2 - korkea

16	Toiminnalliset	Kalenteri	Järjestelmässä tulee olla vähintään seuraavat eritasoiset kalenterinäkömät: - käyttäjäkohtainen - organisaatiokohtainen (ml. aliorganisaatiot) - asiakaskohtainen - sopimuskohtainen - projektiokohtainen - aikaperusteinen (esimerkiksi viikko-, kuukausi- ja vuosikalenteri)	Hyväksytty	1 - kriittinen
17	Toiminnalliset	Kalenteri	Kalenterinäkömästä tulee voida porautua yksittäiseen kalenterissa näkyvään tietoon	Hyväksytty	2 - korkea
18	Toiminnalliset	Kalenteri	Kalenterinäkömään tulee voida lisätä vähintään seuraavia tietoja: - käyttäjien lomajaksot sekä muut poissaolotiedot - tehtävät - koulutukset, tilaisuudet ja tapahtumat	Hyväksytty	2 - korkea
40	Toiminnalliset	Kalenteri	Järjestelmän tulee lisätä, päivittää ja poistaa henkilölle allokoitua tehtäviä myös käyttäjän Notes-kalenteriin. Päivitys- ja poistooperaatiot tulee rajoittua vain järjestelmän lisäämiin merkintöihin.	Hyväksytty	3 - matala
82	Ei-toiminnalliset	Luotettavuus	Järjestelmän käyttäjälle näytettävät tiedot tulee olla jatkuvasti ajantasaisia. Vaatimuksella tarkoitetaan järjestelmän sisäistä konsistenssia, järjestelmän ulkopuolisten rajoitteiden vaikutusta ei tässä tarvitse huomioida	Hyväksytty	1 - kriittinen
80	Ei-toiminnalliset	Rajapinnat	Järjestelmän tulee integroitua Oikeusrekiesterikeskuksen talouden suunnittelujärjestelmään (Suunta)	Hyväksytty	2 - korkea
81	Ei-toiminnalliset	Rajapinnat	Järjestelmän tulee integroitua Oikeusrekiesterikeskuksen käytössä olevaan Lotus Notes -kalenteriin	Hyväksytty	3 - matala
90	Ei-toiminnalliset	Rajapinnat	Järjestelmän tulee integroitua Oikeusrekiesterikeskuksen työajankirjausjärjestelmään (Timme)	Hyväksytty	2 - korkea
26	Toiminnalliset	Raportointi	Järjestelmän tulee sisältää yksinkertainen toiminnallisuus sisällön raportoimiseksi	Hyväksytty	1 - kriittinen

27	Toiminnalliset	Raportointi	Järjestelmän tulee tuottaa ainakin seuraavat raportit: 1) Henkilöraportti 2) Ryhmäraportti 3) Johdon raportit 4) Asiakasraportit 5) Projekti- ja tehtäväraportti 6) Järjestelmäraportti Raportit on kuvattu tarkemmin liitteessä (Raportointi.doc)	Hyväksytty	1 - kriittinen
28	Toiminnalliset	Raportointi	Käyttäjän tulee voida valita miltä ajanjaksolta raportti tuotetaan	Hyväksytty	1 - kriittinen
29	Toiminnalliset	Raportointi	Käyttäjän tulee voida rajata raportin sisältö vähintään henkilö-, organisaatio-, järjestelmä- ja/tai asiakaskohtaisesti sekä metatietojen kuten luokittelun mukaisesti: - esimerkki 1: ryhmän esimies haluaa raportoida oman ryhmänsä (organisaatio) tulevat ja tehdyt tehtävät - esimerkki 2: asiakaspäällikkö haluaa raportoida asiakkaalleen tälle edellisellä seurantajaksolla tehdyt tehtävät ja niiden toteutuneen työajan - esimerkki 3: palvelupäällikkö haluaa raportoida yksittäisen järjestelmän tulevat ja tehdyt tehtävät	Hyväksytty	2 - korkea
87	Toiminnalliset	Raportointi	Käyttäjän tulee voida tallentaa raportti työasemalleen vähintään PDF-muotoisena tiedostona.	Hyväksytty	1 - kriittinen
91	Toiminnalliset	Raportointi	Raportin sisältö tulee rajoittaa käyttäjän käyttöoikeuksien mukaan. Raportille ei saa tuoda tietoa, johon käyttäjällä ei ole oikeuksia.	Hyväksytty	1 - kriittinen
41	Toiminnalliset	Resurssit	Tietyillä käyttäjärooleilla tulee voida tehdä resurssivaraus ilman työtilausta tai tehtävää	Hyväksytty	1 - kriittinen
60	Ei-toiminnalliset	Saatavuus	Järjestelmän tulee olla asiakkaan käytettävissä vähintään suosituksessa JHS174 kuvatun palveluajan P2 mukaisesti (arkipäivisin klo 7-19).	Hyväksytty	1 - kriittinen
72	Ei-toiminnalliset	Saatavuus	Järjestelmässä on voitava varmuuskopioida kaikki järjestelmän palauttamiseksi tarvittava tieto.	Hyväksytty	1 - kriittinen

75	Ei-toiminnalliset	Saatavuus	Järjestelmän vasteaika kaikilla tapahtumilla tulee olla maksimissaan 2 sekuntia kuormituksesta riippumatta. Raportointitoiminnallisuudessa vasteaika saa kuitenkin olla maksimissaan 15 sekuntia	Hyväksytty	1 - kriittinen
76	Ei-toiminnalliset	Saatavuus	Järjestelmän on toimittava suorituskykyisesti vähintään 20 yhtäaikaisen käyttäjän tuottamalla kuormalla	Hyväksytty	1 - kriittinen
84	Ei-toiminnalliset	Saatavuus	Järjestelmän tulee täyttää palveluaikana suosituksessa JHS174 kuvatun käytettävyyden K2 tason (99,0 % käytettävyys, maksimikatko palveluaikana 4 tuntia).	Hyväksytty	1 - kriittinen
85	Ei-toiminnalliset	Saatavuus	Järjestelmän huoltokatkot tulee ensisijaisesti ajoittaa palveluajan ulkopuolelle, mutta vähintään klo 8-16:15 välisen ajan ulkopuolelle. Mikäli huoltakatko ajoitetaan palveluajalle, tulee tästä sopia asiakkaan kanssa etukäteen vähintään 2 viikkoa ennen aiottua huoltokatkoa.	Hyväksytty	1 - kriittinen
89	Ei-toiminnalliset	Saatavuus	Varmuuskopiointi ei saa aiheuttaa järjestelmälle käyttökatoa	Hyväksytty	1 - kriittinen
35	Toiminnalliset	Sopimukset	Käyttäjien tulee voida kirjata asiakkuuksiin ja sopimuksiin kohdittuvia kehitysehdotuksia, joita esimerkiksi asiakas- ja palvelupäälliköt voivat esittää asiakkaille.	Hyväksytty	2 - korkea
38	Toiminnalliset	Sopimukset	Järjestelmän tulee noutaa Suunta-järjestelmästä suunnitellut ja toteutuneet työmäärät sopimuksille	Hyväksytty	2 - korkea
39	Toiminnalliset	Sopimukset	Sopimukseen tulee voida liittää hypertekstilinkki sopimuskannassa olevaan sopimusasiakirjaan	Hyväksytty	2 - korkea
42	Toiminnalliset	Sopimukset	Järjestelmään tulee voida mallintaa sekä palvelusopimus että tulossopimus. Sopimuksille tulee voida kirjata vähintään seuraavat tiedot: - asiakas - asiakaspäällikkö - sopimuskausi - sovittu kokonaistyömäärä (HTP)	Hyväksytty	1 - kriittinen
6	Toiminnalliset	Tehtävät	Käyttäjän tulee voida merkitä tehtävä omaan työjonoonsa	Hyväksytty	1 - kriittinen
7	Toiminnalliset	Tehtävät	Tehtävä voi olla samanaikaisesti useamman henkilön työjonoissa	Hyväksytty	1 - kriittinen
8	Toiminnalliset	Tehtävät	Tehtävälle tulee voida määritellä vastuuhenkilö järjestelmän käyttäjien joukosta	Hyväksytty	1 - kriittinen



12	Toiminnalliset	Tehtävät	Tietyillä käyttäjärooleilla tulee voida varata henkilöresurssi tehtävälle. Tällöin järjestelmä merkitsee tehtävän automaattisesti henkilön työjonoon.	Hyväksytty	1 - kriittinen
19	Toiminnalliset	Tehtävät	Tehtäville tulee voida tallentaa vähintään seuraavat tiedot: - Kuvaus - Äiti-objekti (esimerkiksi ylemmän tason tehtävä tai sopimus) - Prioriteetti - Aikataulu (aloituspäivä sekä deadline) - Riippuvuudet muihin tehtäviin - Arvioitu työmäärä	Hyväksytty	1 - kriittinen
21	Toiminnalliset	Tehtävät	Käyttäjän tulee voida lisätä omaan työjonoonsa myös tehtäviä, jotka eivät liity mihinkään ylemmän tason tehtävään tai sopimukseen	Hyväksytty	1 - kriittinen
22	Toiminnalliset	Tehtävät	Tietyillä käyttäjärooleilla tulee voida lisätä toisen henkilön työjonoon tehtäviä, jotka eivät liity mihinkään ylemmän tason tehtävään tai sopimukseen	Hyväksytty	1 - kriittinen
24	Toiminnalliset	Tehtävät	Tietyillä käyttäjärooleilla tehtävän prioriteettia työntekijän omassa työjonoissa tulee voida määritellä huolimatta tehtävän sopimuksen mukaisesta prioriteetista. Tehtävän prioriteetti voi siis olla erilainen eri henkilöiden työjonoissa	Hyväksytty	2 - korkea
25	Toiminnalliset	Tehtävät	Tehtävällä tulee olla edistymisen seuranta varten asteikko (esimerkiksi valmiusaste), jonka avulla tehtävän edistymistä voidaan seurata	Hyväksytty	2 - korkea
30	Toiminnalliset	Tehtävät	Tehtävillä tulee olla pakollinen tilatieto	Hyväksytty	1 - kriittinen
34	Toiminnalliset	Tehtävät	Järjestelmän on pystyttävä näyttämään käyttäjien tehtäviin käytämä ja kirjaama työaika. Tämä voidaan toteuttaa esimerkiksi kirjaamalla käytetty työaika tehtäville tai integroitumalla nykyiseen työajanseurantajärjestelmään. Integroiduttaessa tehtäväkohtainen jaottelu pitäisi viedä myös nykyiseen järjestelmään oikean tarkkuuden saavuttamiseksi.	Hyväksytty	2 - korkea
43	Toiminnalliset	Tehtävät	Järjestelmän tulee noutaa Timme-järjestelmästä toteutuneet työmäärät tehtäville	Hyväksytty	2 - korkea
49	Ei-toiminnalliset	Tietoturva	Pääsy järjestelmään on rajoitettu tiettyihin verkko-osoitteisiin	Hyväksytty	1 - kriittinen

56	Ei-toiminnalliset	Tietoturva	Kaikki järjestelmän yhteydet tulee olla salattuja (esimerkiksi http- ja ftp-liikenne)	Hyväksytty	1 - kriittinen
58	Ei-toiminnalliset	Tietoturva	Järjestelmän tulee tukea roolipohjaista käyttövaltuuksien hallintaa	Hyväksytty	1 - kriittinen
63	Ei-toiminnalliset	Tietoturva	Sisäänkirjautumiset ja niiden yritykset kirjoitetaan lokiin, niin että yksittäisen käyttäjän kirjautumiset ja kirjautumisyritykset järjestelmään voidaan selvittää	Hyväksytty	1 - kriittinen
64	Ei-toiminnalliset	Tietoturva	Järjestelmän tuottamien lokien tiedon määrän on oltava parametroitavissa esimerkiksi siten, että "lokitustasoa" voidaan muuttaa	Hyväksytty	2 - korkea
65	Ei-toiminnalliset	Tietoturva	Järjestelmän käyttäjärooleihin on voitava liittää ulkoisesta keskitetystä käyttäjähakemistosta luettavia käyttäjiä ja käyttäjäryhmiä	Hyväksytty	2 - korkea
66	Ei-toiminnalliset	Tietoturva	Järjestelmän käyttöoikeuksien hallinta huolehtii siitä, että todennetut käyttäjät voivat suorittaa käyttöoikeuksiensa mukaisia toimintoja. Sallitut toimenpiteet määräytyvät käyttäjäryhmien ja -roolien mukaan	Hyväksytty	1 - kriittinen
67	Ei-toiminnalliset	Tietoturva	Käyttöoikeudet on voitava määritellä monipuolisesti ja yksityiskohtaisesti tarkastelukohteen mukaan. Mikäli käyttäjällä ei ole oikeutta tiettyyn toiminnallisuuteen tai tietoon, järjestelmä piilottaa tiedot tai estää toiminnon automaattisesti käyttäjän näkymästä	Hyväksytty	1 - kriittinen
68	Ei-toiminnalliset	Tietoturva	Samalle käyttäjälle on voitava liittää useita eri käyttäjäryhmiä ja -rooleja siten, että käyttäjällä voi olla eritasoisia oikeuksia eri toimintoihin	Hyväksytty	1 - kriittinen
69	Ei-toiminnalliset	Tietoturva	Järjestelmän tulee tukea kertakirjautumista (SSO) Kerberos-pohjaisella menettelyllä ja/tai IBM ISIM ESSO -tuotteella	Hyväksytty	2 - korkea
70	Ei-toiminnalliset	Tietoturva	Järjestelmään kirjaudutaan käyttäjätunnusta ja salasanaa käyttäen, jos kertakirjautuminen ei ole käytössä	Hyväksytty	1 - kriittinen
71	Ei-toiminnalliset	Tietoturva	Salasanoja ei tallenneta selkokielisenä järjestelmään	Hyväksytty	1 - kriittinen
83	Ei-toiminnalliset	Tietoturva	Järjestelmän on kirjoitettava lokia kaikista järjestelmässä tehdyistä muutoksista. Lokimerkinnästä on nähtävä kuka on muuttanut mitä tietoa.	Hyväksytty	2 - korkea

92	Ei-toiminnalliset	Tietoturva	Järjestelmän tulee täyttää valtiohallinnon tietoturvallisuudesta annetun asetuksen 681/2010 mukaisen perustason tietoturvaluokituksen VAHTI-ohjeistuksen mukaisesti.	Hyväksytty	1 - kriittinen
96	Ei-toiminnalliset	Tietoturva	Järjestelmä varmistaa käyttäjältä järjestelmään tallennettujen tietojen poiston	Hyväksytty	2 - korkea
11	Toiminnalliset	Uutissyöte	Käyttäjän tulee voida valita oman uutissyötteensä sisältö esimerkiksi metatietojen, asiakkuuden ja/tai sopimuksen mukaan	Hyväksytty	2 - korkea
13	Toiminnalliset	Uutissyöte	Tietyillä käyttäjärooleilla tulee voida valita sisältöä toisen henkilön uutissyötteeseen	Hyväksytty	3 - matala
14	Toiminnalliset	Uutissyöte	Tietyillä käyttäjärooleilla tulee voida valita sisältöä kaikkien henkilöiden yhteiseen uutissyötteeseen	Hyväksytty	3 - matala
44	Toiminnalliset	Uutissyöte	Järjestelmän tulee näyttää käyttäjälle uutissyöte viimeisimmistä järjestelmään tallennettujen tietojen muutoksista.	Hyväksytty	2 - korkea
1	Toiminnalliset	Yleiset	Järjestelmän tulee mahdollistaa automaattinen versiointi tiedoille ja liitetiedoille. Versiointi tulee olla kytkettävissä päälle/pois turhan tiedon keräämisen välttämiseksi	Hyväksytty	2 - korkea
2	Toiminnalliset	Yleiset	Järjestelmässä tulee olla toiminnallisuus dokumenttien ja tietojen katselmoinnille ja hyväksymiselle	Hyväksytty	2 - korkea
5	Toiminnalliset	Yleiset	Järjestelmän tietoalkiot tulee voida määritellä hierarkkisiksi. Esimerkiksi organisaatio -> aliorganisaatio -> työntekijä; sopimus -> sopimuksen tehtävä -> kehitystehtävä -> kehitystehtävän osatehtävä	Hyväksytty	1 - kriittinen
9	Toiminnalliset	Yleiset	Järjestelmän tulee sisältää tilatiedot kaikille keskeisille tiedoille (vähintään käsitellissä kuvatuille käsitteille) ja liitetiedoille	Hyväksytty	2 - korkea
10	Toiminnalliset	Yleiset	Järjestelmän keskeisille tiedoille (vähintään käsitellissä kuvatuille käsitteille) ja liitetiedoille tulee voida määritellä metatietoja (esimerkiksi luokittelu)	Hyväksytty	2 - korkea
20	Toiminnalliset	Yleiset	Järjestelmään tulee sisältyä henkilökohtainen näkymä, jossa käyttäjä voi vähintään tarkastella, hallinnoida ja päivittää omien tehtävien tietoja	Hyväksytty	1 - kriittinen

33	Toiminnalliset	Yleiset	Käyttäjän tulee voida tallentaa järjestelmään liitetiedostoja, jotka käyttäjän tulee voida linkittää yhteen tai useampaan järjestelmässä olevaan tietoon, kuten esimerkiksi tehtävään, sopimukseen tai järjestelmään	Hyväksytty	1 - kriittinen
36	Toiminnalliset	Yleiset	Järjestelmän tulee mahdollistaa porautuminen hierarkkisten tietojen kautta ylös sekä alaspäin, esimerkiksi tehtävästä työtilaukseen tai sopimukseen, ja edelleen sopimuksesta asiakkaaseen.	Hyväksytty	1 - kriittinen
57	Ei-toiminnalliset	Ylläpidettävyys	Järjestelmän tulee perustua yleisesti käytössä olevaan teknologiaan, jolle löytyy vähintään kolme Suomessa toimivaa toimittajaa	Hyväksytty	1 - kriittinen
62	Ei-toiminnalliset	Ylläpidettävyys	Järjestelmän ylläpitoon tarkoitettu tekninen dokumentaatio on suomen- tai englanninkielistä.	Hyväksytty	1 - kriittinen
4	Toiminnalliset	Ylläpito	Kaikkia järjestelmän määrittämiä ja tietosisältöä tulee voida ylläpitää (lisätä, poistaa ja muokata) käyttöliittymältä	Hyväksytty	1 - kriittinen
37	Toiminnalliset	Ylläpito	Pääkäyttäjän tulee voida määrittellä henkilötasolla sekä organisaatiotasolla käytettävissä olevan työajan.	Hyväksytty	2 - korkea
48	Toiminnalliset	Uutissyöte	Uutissyötteen sisältö tulee voida määrittellä sekä käyttäjäkohtaisesti, että kaikille yhteisesti.	Hylätty	
23	Toiminnalliset	Yleiset	Järjestelmän on oltava läpinäkyvä ja tuettava helppoa tiedonkulkua. Henkilöstön tulisi päästä tarkastelemaan kaikkia omaa virastoaan ja sen henkilöstöä koskevia tietoja	Hylätty	
3	Toiminnalliset	Ylläpito	Organisaatorakennetta, rooleja ja työntekijätietoja tulee voida ylläpitää (lisätä, poistaa ja muokata) käyttöliittymältä	Hylätty	

## RAPORTIT

Alla on listattu joitakin oleellisimpia raportteja, joita järjestelmän tulee tuottaa. Järjestelmässä tulee olla yksinkertainen käyttöliittymältä käytettävä raportointitoiminnallisuus, jolla jokainen käyttäjä voi ajaa tarvitsemiaan raportteja käyttöoikeuksiensa puitteissa. Raportointitoiminnallisuus ei siis saa rajoittua vain esimerkkinä kuvattuihin raportteihin.

Raporttien käyttöoikeudet tulee voida määritellä järjestelmän käyttäjärooleihin sekä organisaatietietoihin perustuen. Esimerkiksi ryhmäraportin ensisijainen käyttäjä on ryhmän esimies. Palvelupäällikön taas tulee nähdä helposti omien järjestelmiensä raportit.

### Henkilöraportti

Henkilöraportilla listataan yksittäisen henkilön työjonossa tai kalenterissa olevat tiedot. Käyttäjän tulee voida valita ajanjakso, jolta raportti luodaan. Käyttäjän tulee voida valita raportille vähintään seuraavia tietoja:

- tehdyt tehtävät
- suunnitellut tehtävät
- suunniteltujen ja tehtyjen tehtävien ero
- lomat ja poissaolot
- muut tapahtumat, koulutukset ynnä muut

### Ryhmäraportti

Ryhmäraportilla listataan jonkin järjestelmään määritellyn organisaation tietoja. Organisaatio voi olla esimerkiksi ryhmä, toimiala tai koko organisaatio. Käyttäjän tulee voida valita ajanjakso, jolta raportti luodaan. Käyttäjän tulee voida valita raportille kaikki henkilöraportilla mahdolliset tiedot sekä vähintään seuraavia tietoja:

- suunnitellut työmäärät
- toteutuneet työmäärät
- suunniteltujen ja toteutuneiden työmäärien ero
- ryhmän käytettävissä oleva työmääräkapasiteetti ja käyttöaste

### Johdon raportit

Johdon raporteilla voidaan raportoida sopimusten tavoitteiden täyttymistä.

- onko työn alla
- mitä puuttuu
- mitkä aloittamatta
- resurssien allokointi

### Asiakasraportit

Asiakasraporteilla raportoidaan yhteen asiakkuuteen liittyviä tietoja. Käyttäjän tulee voida raportoida vähintään seuraavia tietoja:

- miten asiakkuuden sopimukseen ja tehtäviin on kirjattu työaika
- mikä on suunnitellun ja toteutuneen työmäärän suhde / ero
- mitä tehtäviä asiakkuuteen liittyen on suunniteltu (vuosikello)
- kuinka asiakkuuden sopimuksissa sovittujen tavoitteiden täyttyminen etenee
- mahdollisesti käytetyt konsulttityötunnit

### Projekti- ja tehtäväraportti

Projekti- ja tehtäväraportilla voidaan raportoida yksittäiseen projektiin tai tehtävään liittyvien alitehtävien tilannetta.

- tehdyt tehtävät
- suunnitellut tehtävät
- suunniteltujen ja tehtyjen tehtävien ero
- suunnitellut työmäärät
- toteutuneet työmäärät
- suunniteltujen ja toteutuneiden työmäärien ero
- aikataulu

### Järjestelmäraportti

Järjestelmäraportilla raportointi keskittyy johonkin järjestelmään liittyviin tehtäviin, resursseihin sekä muihin tarpeellisiin tietoihin kuten suunniteltuihin versiovaihtoihin. Versiovaihdot voivat olla esimerkiksi yksittäisiä tehtäviä, joihin liittyy alitehtäviä, mutta niitä on korostettu tässä merkittävyytensä sekä muista raporteista poikkeavuuden vuoksi.

- tehdyt tehtävät
- suunnitellut tehtävät
- suunniteltujen ja tehtyjen tehtävien ero
- suunnitellut työmäärät
- toteutuneet työmäärät
- suunniteltujen ja toteutuneiden työmäärien ero
- tulevat versiovaihdot

Palvelupäälliköillä on roolinsa vuoksi suurin kiinnostus järjestelmäraportteihin. Palvelupäälliköllä on vastuullaan tyypillisesti useampia järjestelmiä, joten järjestelmäraportti tulee olla helposti ajettavissa esimerkiksi palvelupäällikön mukaan.

### Tehtäväkohtaiset raportit

Edellä kuvattujen raporttien lisäksi järjestelmän tulee mahdollistaa erinäisten yksittäisiin tehtäviin liittyvien raporttien tuottaminen erilaisin kriteerein. Esimerkiksi hankinnat ja koulutukset ovat yksiköitä, joiden raportointimahdollisuus on tärkeä. Hankinnat voivat järjestelmän kannalta olla tehtäviä tai projekteja, joihin liittyy alitehtäviä, mutta niihin liittyy tyypillisesti raportoinnin kannalta tärkeämpänä elementtinä kustannustieto, joka ei ole nyt kuvatus toiminnanohjausjärjestelmän keskeisiä tietoelementtejä. Koulutuksista taas halutaan raportoida esimerkiksi osallistujia, kestoja, kustannuksia ja mahdollisia sertifiointeja. Myös nämä tarpeet on kuvattavissa esimerkiksi tehtävin tai projektein, mutta edelleen niiden raportointi eroaa tavallisista tehtävistä tai projekteista.