



## Vaatimusten käsittely ohjelmiston kehittämisessä

Susanna Viikki

Opinnäytetyö

Tietojenkäsittelyn koulutusohjelma

2013



<b>Tekijä tai tekijät</b> Susanna Viikki	<b>Ryhmä tai aloitusvuosi</b> HETI10SIM3
<b>Opinnäytetyön nimi</b> Vaatimusten käsittely ohjelmiston kehittämisessä	<b>Sivu- ja liitesivumäärä</b> 34 + 22
<b>Ohjaaja tai ohjaajat</b> Kai Kivimäki	
<p>Tämän opinnäytetyön tarkoituksena oli kartoittaa toimeksiantajayrityksen projekti- ja portfoliohallinnan järjestelmän nykytila sen hyödyntämisen ja järjestelmään tehtyjen muutosten näkökulmasta. Tavoitteena oli tuottaa kaikille järjestelmän kanssa toimiville tahoille yhteinen ymmärrys järjestelmän nykytilasta. Työn tuotos toimii lähtökohtana tulevan kehittämissuunnitelman luonnissa.</p> <p>Työ rajattiin koskemaan vain järjestelmään jo toteutettujen vaatimusten dokumentointia, uusien vaatimusten keräys ja kartoitus rajattiin työn ulkopuolelle. Työssä ei myöskään kehitetty toimeksiantajayrityksessä käytössä olevaa vaatimusedokumentointi mallipohjaa tai tutkittu mallipohjan toimivuutta sovellusalueelle.</p> <p>Tietoperustassa käsitellään vaatimusten käsittelyn osa-alueita, vaatimusmäärittelyä ja vaatimustenhallintaa. Lisäksi tutustutaan vaatimusmäärittelyn aktiviteetteihin. Vaatimusten käsittely luo perustan onnistuneelle ohjelmistokehittämiselle ja -ylläpidolle. Hyvin toteutetun vaatimusmäärittelyn avulla voidaan varmistaa, että ohjelmisto vastaa tilaajan odotuksia ja toteutus vastaa tilaajan tarvetta. Tietoperustassa perehdytään myös ohjelmiston takaisinmallinnukseen sekä sen hyödyntämiseen järjestelmän uudelleendokumentoinnin työtapana.</p> <p>Aiheen valinta ja suunnittelu sekä työn tuotoksen toteutustavan suunnittelu sijoittui marras- joulukuulle 2012. Työn tietoperusta kirjoitettiin tammikuussa 2013. Tämän jälkeen suunniteltiin ja toteutettiin työn tuotokset.</p> <p>Työn tuloksena syntyi projekti- ja portfoliohallinnan järjestelmän vaatimusedokumentointi, johon dokumentoitiin järjestelmän nykytilaan vaikuttaneet liiketoiminnan ja toiminnalliset vaatimukset. Tämän lisäksi kartoitettiin järjestelmän tukemat kehittämisprosessit. Vaatimukset ryhmiteltiin niiden tukemien prosessien ja toteutustavan mukaisesti kuvaamaan järjestelmän nykytilaa. Työn tuloksina toimeksiantaja sai järjestelmän kehittämissuunnitelmaan tarvittavan taustamateriaalin.</p>	
<b>Asiasanat</b> Vaatimus, vaatimusten käsittely, ohjelmiston kehittäminen, takaisinmallinnus	

Degree Programme in Information Technology

<p><b>Author or authors</b> Susanna Viikki</p>	<p><b>Group or year of entry</b> HETI10SIM3</p>
<p><b>The title of thesis</b> Requirements engineering in Software development</p>	<p><b>Number of pages and appendices</b> 34 + 22</p>
<p><b>Supervisor or supervisors</b> Kai Kivimäki</p>	
<p>This Bachelor's thesis examines the current state of project and portfolio management software used by the case company. The goal was to provide information on how and why the software was configured and customized as it was.</p> <p>The thesis was limited to documenting only the requirements that were already implemented in the software. The collecting and analyzing of new requirements were left outside the scope of the study. The requirements document template was not developed, neither was the suitability of the template analyzed in this study.</p> <p>The thesis consists of a theory section and a case section. The theory section discusses the characteristics of requirements engineering, and states the activities used in the requirements definition and requirements management. Requirements engineering creates a base for successful software engineering and maintenance. Through well performed requirements engineering it can be ensured that the software fulfills customers' expectations and the implementing meets the needs of the customers. The theory part also focuses on reverse engineering and how to make use of reverse engineering as a method of software redocumentation.</p> <p>The case of the study was planned during November and December 2012. The theory section was written in January 2013. After that the case of the study was executed.</p> <p>The case part presents how the result of the study was accomplished. The result of the thesis was a requirement documentation of project and portfolio management software. The documentation included business requirements and functional specifications that have affected the configuration or customization to the software. Development processes supported by the software were also documented. The current state of the project and portfolio management software was presented through requirements grouped by supported processes and the way they were implemented. The case company can use the current state document as base material for the roadmap.</p>	
<p><b>Key words</b> Requirement, requirements engineering, software engineering, reverse engineering</p>	

# Sisällys

1	Johdanto .....	1
1.1	Opinnäytetyön tavoitteet.....	3
1.2	Opinnäytetyön rajaus.....	3
2	Vaatimusten käsittely .....	5
2.1	Vaatus	6
2.2	Asiakasvaatimuksista ohjelmistovaatimuksiin .....	8
2.3	Vaatimusten käsittelyn osa-alueet.....	9
2.4	Vaatimusten jäljitettävyys .....	14
2.5	Vaatimusten jatkotyöstäminen .....	15
2.6	Ohjelmien takaisinmallinnus.....	15
2.7	Vaatimusmäärittelyn merkitys.....	18
3	Vaatimusten dokumentointi .....	20
3.1	Vaatimusten dokumentoinnin suunnittelu ja toteutuksen aikataulu .....	21
3.2	Valmisohjelmisto ja sen hyödyntäminen.....	22
3.3	Vaatimusten dokumentoinnin toteutus.....	24
3.4	Takaisinmallinnus vaatimusten dokumentoinnissa.....	28
4	Yhteenveto .....	30
4.1	Johtopäätökset .....	32
4.2	Jatkokehitysehdotukset.....	33
	Lähteet.....	35
	Liitteet.....	38
	Liite 1. Keskeiset käsitteet.....	38
	Liite 2. Projektisuunnitelma .....	41
	Liite 3. Ohjelmistotoimittajan näkemys ohjelmiston konfiguraatioista ja kustomoinneista.....	45
	Liite 4. Prosessit kehittämisessä .....	46
	Liite 5. Vaatimusdokumentaatio .....	47
	Liite 6. Dokumentoitujen vaatimusten yhteenveto .....	59

# 1 Johdanto

Vaatimusten käsittely on oleellinen osa kaikkea ohjelmistokehittämistä ja -ylläpitoa. Vaatimusmäärittely antaa suuntaviivat ja ohjenuorat ohjelmistokehittämiseen. Vaatimusmäärittelyn eri aktiviteettien kautta saadaan tarkka kuvaus kehitettävästä ohjelmistosta. Hyvin toteutetun vaatimusmäärittelyn avulla voidaan varmistua, että ohjelmiston kehityksen tuotos vastaa tilaajan odotuksia ja toteutus tilaajan tarvetta. Tämä opinnäytetyö on selvitys ohjelmiston nykytilasta. Nykytilan määrittely toteutettiin työssä dokumentoimalla ylläpito vaiheessa olevan ohjelmiston vaatimukset. Lisäksi vaatimukset ryhmiteltiin niiden tukemien prosessien ja vaatimuksen toteutustavan mukaisesti.

Työn toimeksiantaja on finanssisektorilla toimiva yritys. Työssä tuotettu nykytilan selvitys ja vaatimusten dokumentointi tehtiin toimeksiantajalla käytössä olevaan projekti- ja portfoliohallinnan valmisohjelmistoon. Valmisohjelmisto on toteutettu niin, että se voidaan helposti muokata vastaamaan hyödyntävän organisaation tarpeita.

Opinnäytetyön raportissa on kuvattu tietoperustana käytetty vaatimusten käsittely, sen osa-alueiden vaatimusmäärittelyn ja vaatimustenhallinnan prosessit sekä niiden aktiviteetit. Lisäksi työssä kuvataan ohjelmien takaisinmallinnusta yleisesti sekä pohditaan takaisinmallinnuksen hyödyntämistä ohjelmiston uudelleendokumentoinnin työtapana ja ohjelmiston jatkokehittämisen pohjana. Raportissa on kuvattu työn toteutus, työtavat ja tuotokset. Työn teko ajoittui toukokuun 2012 ja toukokuun 2013 väliseen aikaan. Työn alkuperäinen aihe ja sen myötä myös aikataulu muuttuivat toimeksiantajan puolelta loppusyksyllä 2012. Aiheen tarkennuksen jälkeen tarkennettiin toteutuksen aihe ja suunniteltiin sen näkökulma, rajaus sekä tuotoksen muoto. Tietoperustan kirjoittamisen jälkeen tehtiin tuotoksen tarkempi suunnittelu ja toteutus. Työn aikataulu on kuvattu projektisuunnitelmassa.

Työn toteutuksen lähtötilanteessa toimeksiantajalla oli käytössään yhtenäiset ja toimivat dokumenttien mallipohjat ja ohjeistukset tietojärjestelmien vaatimusmäärittelylle. Työssä ei ollut tarvetta kehittää vaatimusmäärittelyn mallipohjia tai ohjeistuksia. Mallipohjista huolimatta ohjelmiston dokumentaatiosta puuttui vaatimusdokumentaatio lähes kokonaan. Ohjelmistoa ylläpitävä ja kehittävä muutaman hengen työryhmä oli pysynyt

muuttumattomana vuosia. Varsinaisia integraatioita muihin järjestelmiin ei juuri ollut. Työryhmällä oli hyvä ymmärrys toimeksiantajan organisaation tarpeista sovelluksen hyödyntämisessä sekä sovelluksen tukemista prosesseista. Sovellus tuki ennen kaikkea organisaation sisäistä työtä. Työryhmä tunsu ohjelmiston toiminnallisuudet sekä teknologian erinomaisesti. Edellä mainitut seikat vaikuttivat siihen, että ohjelmiston vaatimusmäärittelyn dokumentaatio oli jäänyt olemattomaksi. Työryhmä pystyi toteuttamaan ohjelmiston kehittämistyötä ohjaavat vaatimukset, vaikka niitä ei dokumentoitu.

Sovelluksen hyödyntämistä oli laajennettu toimeksiantajan organisaatiossa vuosien saatossa työajanseurannasta, henkilötyönlaskutuksesta ja projektinhallinnasta kapasiteettihallintaan sekä viime vuosina hankehallintaan ja kehittämisen suunnitteluun. Ohjelmiston edeltäjät ja nykyisen ohjelmiston eri versiot ovat olleet käytössä kymmeniä vuosia. Nyt oli nähty tarpeelliseksi tarkastella sovellusta ja sen tarjoamia mahdollisuuksia suhteessa toimeksiantajan hyödyntämisen tarpeisiin. Sovelluksen hyödyntäminen nykyisellään on aiheuttanut merkittäviä muutoksia, konfigurointeja ja kustomointeja järjestelmään. Muutosten määrä ja niiden merkitys hyödyntämisessä viittaa enemmän räätälöityyn, itse tehtyyn, ohjelmistoon kuin valmisohjelmistoon. Ohjelmiston tulevaisuuden kehittämissuunnitelman tarve on havaittu. Tässä työssä toteutettu nykytilan kuvaus tulee toimimaan sovelluksen kehittämissuunnitelman pohjana ja esiselvityksenä.

Opinnäytetyö prosessin alussa työn aiheena oli kehittämissuunnitelman laatiminen. Työn edetessä havaittiin, että ennen kuin kehittämissuunnitelma voidaan toteuttaa, tarvitaan parempi kuvaus sovelluksen nykytilasta ja hyödyntämisestä. Kuvaustavaksi valittiin vaatimusten dokumentointi. Työssä dokumentointiin vaatimukset, joiden toteutus on aiheuttanut joko konfigurointia tai kustomointia ohjelmistoon. Valmisohjelmiston perustoiminnallisiin liittyviä vaatimuksia ei dokumentoitu työssä. Konfiguroinnin ja kustomoinnin käsitteitä käytetään työssä ohjelmistotoimittajan määrityksen mukaisesti. Kaikki työn keskeiset käsitteet on kuvattu liitteessä 1. Työn aihealuetta käsitellään toimeksiantajan näkökulmasta.

## 1.1 Opinnäytetyön tavoitteet

Työn tietoperustan tavoitteena on luoda käsitys vaatimusten käsittelystä ohjelmistokehittämisessä ja ylläpidossa koko ohjelmiston elinkaaren ajan. Tietoperustan tavoitteena on tutustua vaatimusmäärittelyyn ja vaatimusten hallinnan prosesseihin sekä niiden aktiviteetteihin. Takaisinmallinnuksen osalta tavoitteena on luoda käsitys takaisinmallinnuksen hyödyntämisestä jälkikäteen tehtävän ohjelmiston dokumentoinnin työtapana ja ohjelmiston jatkokehittämisen pohjana. Lisäksi tietoperustan tavoitteena on ymmärtää vaatimusten käsittelyn merkitys ohjelmistokehittämisessä.

Tämän opinnäytetyön tavoitteena oli kartoittaa järjestelmän nykytila sen hyödyntämisen ja muutosten näkökulmasta. Tuotoksen tavoitteena on luoda kaikille järjestelmän kanssa toimiville tahoille yhteinen lähtökohta järjestelmän nykytilan arviointiin. Työn lähtötilanteessa järjestelmän kehittämisomistajalla, kehittämiseen osallistuvilla prosessien omistajilla tai ylläpidosta vastuullisella johdolla ei organisaatiossa tapahtuneista muutoksista johtuen ollut selkeää käsitystä järjestelmän nykytilasta. Tuotoksen avulla pyrittiin luomaan ymmärrys miksi järjestelmää oli jouduttu konfiguroimaan ja kustomoimaan, sen sijaan että olisi hyödynnetty valmisohjelmiston omia toiminnallisuuksia.

Työn tarkoitus on dokumentoida järjestelmän vaatimukset sekä koota vaatimuksista yhteenveto. Työssä selvitettiin ja dokumentoitiin järjestelmän nykytilaan vaikuttaneet liiketoiminnan ja toiminnalliset vaatimukset. Vaatimukset ryhmiteltiin niiden tukemien prosessien ja vaatimuksen toteutustavan mukaisesti kuvaamaan järjestelmän nykytilaa. Työn lopputuloksena syntyneen yhteenvedon informaation perusteella organisaatiossa voidaan aloittaa järjestelmän tulevaisuuden kehittämissuunnitelman luominen.

## 1.2 Opinnäytetyön rajaus

Tietopohjasta rajattiin ulkopuolelle vaatimusmäärittelyn mallipohjia koskeva tarkempi esittely tai erilaisten mallipohjien hyötyjen ja haittojen vertailu. Työssä käytettävän vaatimusdokumentaation mallipohjan sopivuutta sovellusalueelle ei myöskään arvioitu. Tietopohjan takaisinmallinnuksen osuudesta on rajattu ulkopuolelle takaisinmallinnuksen hyödyntäminen tuote- tai palveluvakoiluun tai –kopiointiin.

Työssä kuvattiin järjestelmän nykytilannetta. Kuvauksen ulkopuolelle rajattiin uusien, mahdollisesti hyödynnettävien tai vaatimukset täyttävien järjestelmän toiminnallisuuksien tutkiminen. Ohjelmiston toiminnallisuuksien yleinen esittely rajattiin työn ulkopuolelle. Toteutuksen ulkopuolelle rajattiin loppukäyttäjien mielipiteiden kartoitus järjestelmästä, sen tuesta tai toiminnallisuuksista. Työssä ei kehitetä ohjelmiston tukemia prosesseja, eikä oteta kantaa voidaanko liiketoimintavaatimuksien tai prosessien muutosten avulla hyödyntää ohjelmistoa omia toiminnallisuuksia konfiguraatioiden tai kustomointien sijaan.

Työn toteutuksesta rajattiin ulkopuolelle vaatimusmäärittelyn mallipohjien tai ohjeistuksien kehittäminen. Vaatimusmäärittelyn dokumenttien käyttöönotto ohjelmiston kehittämisessä ja ylläpidossa on rajattu työn ulkopuolelle. Työssä ei dokumentoitu vaatimuksia, jotka kohdistuvat projekti- ja portfoliohallinnan ohjelmiston perustoiminnallisuuksiin tai -ominaisuuksiin. Myöskään prosessiin tai raportointiin liittyviä vaatimuksia, jotka eivät aiheuttaneet ohjelmistoon konfiguraatioita tai kustomointeja, ei dokumentoitu. Dokumentoituja toiminnallisia vaatimuksia ei jatkotyöstetty eli spesifikaatioita ei tässä työssä luotu.



## 2 Vaatimusten käsittely

Tietojärjestelmää kehittäessään organisaatio kehittää samalla omaa toimintaansa. Kehittämisen tarkoituksena on saavuttaa toimintatavan muutos, jonka tavoitteena on Pohjosen mukaan auttaa toimijaa suuntautumaan tavoitteisiinsa paremmin sekä mahdollistaa uusien ja vaativampien tavoitteiden asettamisen. Lisäksi tavoitteena on mahdollistaa uusia toimintoja tai tehostaa olemassa olevia toimintatapoja. Toimijan toiminta perustuu teknologian hyödyntämiseen. Kehittämisen tulee kohdistua joko ihmisiin, teknologiaan tai toimintoihin. (Pohjonen 2002, 14.)

Ohjelmiston uudistamisena tehtävä evoluutio on ohjelman taloudellisen ja strategisen arvon nostamista. Evoluutiomuutokset koskevat aina ohjelmistokokonaisuutta. Ylläpidettävät ja uudistettavat ohjelmistot ovat yritykselle niin tärkeitä, että niiden korvaaminen uusilla olisi riskialtista ja hankalaa. Kaikkiin ohjelmistoihin tehdään muutoksia esimerkiksi käyttäjien tarpeiden muuttuessa tai ympäröivän maailman muutoksen vuoksi. (Harsu 2003, 65–67.)

Tietojärjestelmien ja ohjelmistojen kehittämistä voidaan tarkastella kahdesta eri näkökulmasta; räätälöity tai mittatilaustyönä tehty kehittäminen (bespoke development) ja valmisohjelmiston kehittäminen (package development). Räätälöimällä kehitetty ohjelmisto saadaan vastaamaan täydellisesti yrityksen vaatimuksia ja ohjelmisto sekä sen arkkitehtuuri saadaan sopimaan täydellisesti organisaation prosesseihin. Valmisohjelmistoja on mahdollista konfiguroida ja kustomoida organisaation tarpeiden mukaan. Valmisohjelmiston kehittämisessä järjestelmä konfiguroidaan niin, että se vastaa mahdollisimman hyvin organisaation vaatimuksia. Valmisohjelmiston kehittämisessä yleinen yhteys järjestelmän kehittämisen ja organisaation prosessien välillä on käänteinen verrattuna räätälöidyn järjestelmän kehittämiseen. Perinteisesti räätälöityjen järjestelmien kehittämisessä IT-systeemien ja järjestelmien on kohdattava vaatimukset ja organisaation prosessit. Valmisohjelmiston kehittämisessä organisaation prosessien on mukauduttava ohjelmiston vaatimuksiin ja rajoitteisiin. (Beynon-Davis 2009, 349 - 350.)

Räätälöidyn ohjelmiston kehittäminen on kallista ja resursseja vaativaa, joten pienemmät organisaatiot hyödyntävät enemmän paketteja tai palkkaavat ulkoisen hankkijan tuottamaan ohjelmiston. Organisaation on aina luotava business case valmisohjelmiston hyödyntämisestä tai sen räätälöidyn ohjelmiston rakentamisesta. Kehittämisen ja kustannusten näkökulmasta Beynon-Davis näkee kriittiseksi, mikäli organisaatio käyttää valmisohjelmistoa kuten räätälöityä ohjelmistoa. (Beynon-Davis 2009, 350.)

Vaatumusten käsittelyn prosessien tuotoksena saadaan ohjenuorat kaikkeen ohjelmiston kehittämiseen. Vaatimusmäärittelystä saadaan ensin suuntaviivat ja myöhemmin tarkat kuvaukset millainen ohjelmisto kehitystyön tuloksena olisi synnyttävä. Vaatimusmäärittely yhdistää kaikki ohjelmistotuotantoon liittyvät aktiviteetit ja tehtävät. Se on osa jokaista onnistunutta ohjelmistotuotantoprojektia. (Pohl & Rupp 2011, 3.)

Robertson ja Robertson näkevät, että ohjelmiston käyttöönoton jälkeen seuraa väistämättä muutospyyntöjä ja vaatimustenhallintaa. Muutospyynnöt syntyvät, sillä ympäristössä ja ohjelmistoon kohdistuvissa tarpeissa tapahtuu muutoksia. Ohjelmisto korvataan uudella, kun sitä ei enää voida muuttaa siihen kohdistuvien muutostarpeiden tahtiin. Myös Harsu näkee ohjelmistojen muutostarpeen väistämättömänä. Pohlin ja Ruppin mukaan vaatimukset ohjaavat kehittämistä ja ylläpitoa koko ohjelmiston elinkaaren ajan. (Harsu 2003, 65; Pohl & Rupp 2011, 131; Robertson & Robertson 2011, 357.)

## 2.1 Vaatimus

Vaatimus koostuu ohjelmiston avulla ratkaistavien ongelmien kuvauksesta, ohjelmiston toiminnallisuuksien ja ominaisuuksien selvityksistä, sekä suunnittelun ja toteutuksen rajoitteista. Kotonyan ja Sommervillen mukaan vaatimus on

- käyttäjän kokema toiminnallisuus
- yleinen järjestelmän ominaisuus
- järjestelmän reunaehto
- ohjeistus laskutoimitusten suorittamiseen
- järjestelmän kehittämisen reunaehto.

Vaatimukset kuvaavat järjestelmän toimintaa ja käyttäytymistä, sovelluksen tietoja, sovelluksen toimintaan liittyviä rajoitteita tai määrittäviä sovelluksen osioista ja attribuuteista. (Kotonya & Sommerville 2002, 7-8.)

IEEE Std 610.12-1990 standardi (Pohl & Rupp 2011, 3) määrittää vaatimuksen olevan:

- Edellytys tai kyvykkyys, jonka avulla käyttäjä voi ratkaista ongelman tai saavuttaa tavoitteen.
- Edellytys tai kyvykkyys, jonka järjestelmän tai -komponentin on omattava tai täytettävä, jota sopimus, standardi, määritelmä tai muu formaalin dokumentin ehto täyttyy.
- Dokumentoitu esitys edellisissä kohdissa kuvatuista edellytyksistä tai kyvykkyksistä.

Vaatimus on järjestelmän implementoinnin ohjeistus ja kehittämisen rajoite. Vaatimukset kuvaavat miten järjestelmän tulee toimia tai millainen rakenne tai tietojen arvojoukko sovelluksesta tulee löytyä. Sekä Pohjonen että Robertson ja Robertson korostavat, että vaatimuksissa ei oteta kantaa tekniseen toteutukseen, jolla vaatimukset tulisi täyttää. Suunnittelua ja toteutusta koskevat rajoitteet voivat aiheuttaa ongelmia toteutukseen, mikäli ne ovat ristiriidassa muiden vaatimusten kanssa. (Kotonya & Sommerville 2002, 7; Pohjonen 2002, 28; Robertson & Robertson 2011, 2.)

Pohjonen sekä Haikala ja Mikkonen jakavat vaatimukset toiminnallisiin ja ei-toiminnallisiin vaatimuksiin. Pohl ja Rupp käyttävät ei-toiminnallisista vaatimuksista termiä laadullinen vaatimus. Kaikissa kolmessa teoksessa nostetaan kolmanneksi vaatimusten luokaksi reunaehdot tai rajoitteet. Pohjonen näkee reunaehdot ei-toiminnallisten vaatimusten erikoistapauksina. (Pohjonen 2002, 28; Haikala & Mikkonen 2011; 61, Pohl & Rupp 2011, 7-8.)

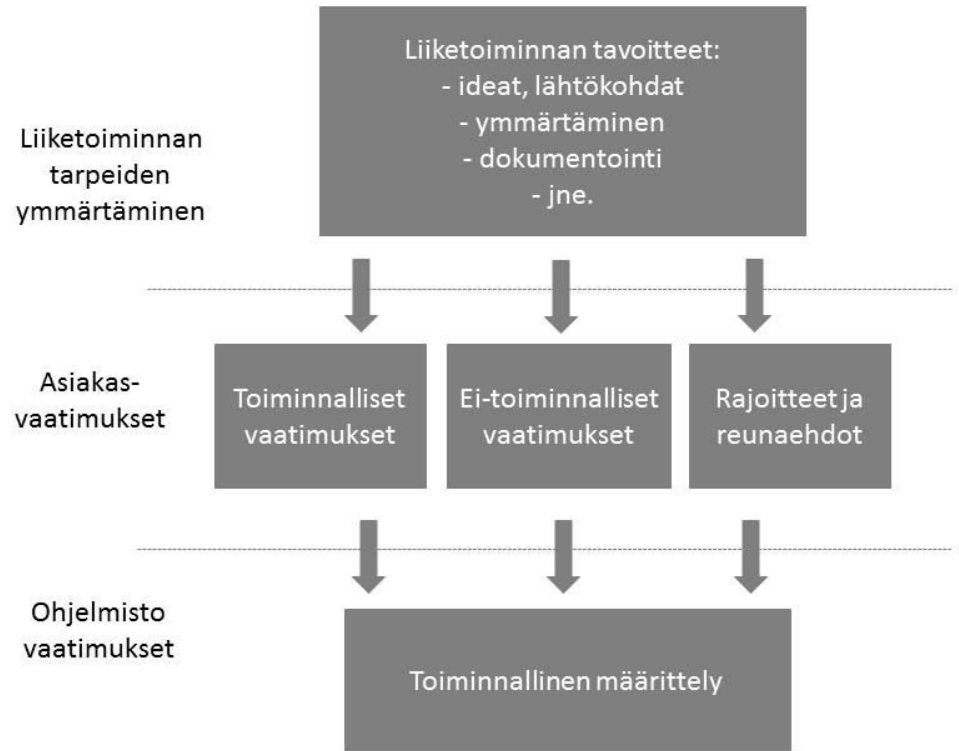
Toiminnalliset vaatimukset kuvaavat toiminnallisuuksia, jotka järjestelmän on täytettävä. Pohl ja Rupp sekä Robertson ja Robertson jakavat toiminnalliset vaatimukset toiminnallisuuksien vaatimuksiin, käyttäytymisen vaatimuksiin sekä tieto-vaatimuksiin. Ei-toiminnalliset eli laadulliset vaatimukset määrittävät järjestelmältä haluttuja ominaisuuksia kuten suorituskykyä, saavutettavuutta, luotettavuutta tai skaalautuvuutta ja näin täy-

dentävät toiminnallisia vaatimuksia. Ei-toiminnalliset vaatimukset määritetään järjestelmäarkkitehtuurin pohjalta. Pohjonen näkee ei-toiminnalliset vaatimukset järjestelmän ominaisuuksiksi tai reunaehdoiksi, joiden vallitessa järjestelmän tulee täyttää toiminnalliset vaatimukset. Rajoitteet voivat koskea järjestelmää tai kehittämistyön prosessia. Rajoitteet nimensä mukaisesti ohjaavat järjestelmän kehittämistä reunaehtojen tapaan. (Pohjonen 2002, 28; Pohl & Rupp 2011 7-8; Robertson & Robertson 2011, 9-11.)

## **2.2 Asiakasvaatimuksista ohjelmistovaatimuksiin**

Asiakasvaatimukset pohjautuvat liiketoiminnallisiin tavoitteisiin ja niiden takaa tulee aina löytyä liiketoimintaan pohjautuva syy ja perustelu. Asiakasvaatimukseen löytyy aina perustelu kysymyksellä miksi? Mikäli vaatimukselle ei ole perustelua, se on ohjelmistovaatimus, ei asiakasvaatimus. Kuviossa 1 on kuvattu miten liiketoiminnallisista tavoitteista muodostetaan ohjelmiston toiminnallinen määrittely. Vaatimusten kuvaus aloitetaan ylätasolla olevien liiketoiminnallisten tavoitteiden ymmärtämisellä. Liiketoimintatarpeiden analysointi tuottaa asiakasvaatimuksia, jotka voidaan jakaa toiminnallisiin ja ei-toiminnallisiin vaatimuksiin sekä reunaehtoihin ja rajoitteisiin. Asiakasvaatimukset kuvaavat asiakkaan konkreettisia tarpeita, eivätkä ne yleensä käsitä sovellusalueen sanastoa. (Haikala & Mikkonen 2011, 63.)

Asiakasvaatimuksien sijaan voidaan puhua ominaisuuksista. Ominaisuus käsittää joukon ohjelman toiminnallisuuksia, joilla asiakkaan tarvetta ratkaistaan. Asiakasvaatimuksen ominaisuuksia ovat virheettömyys, selkeys, tarkkuus ja ymmärrettävyys. Näillä ominaisuuksilla varmistetaan, että vaatimuksen täytyminen on mitattavissa ja vaatimus on oltava testattavissa. Testaamalla mitataan täyttykö vaatimus. Vaatimuksen ominaisuuksiin kuuluu myös taaksepäin jäljitettävyys eli vaatimuksen alkuperä on oltava tiedossa sekä eteenpäin jäljitettävyys eli vaatimuksen tekninen toteutus ja täyttymistä testaavat testitapaukset on oltava tiedossa. (Haikala & Mikkonen 2011, 63.)



Kuvio 1 Asiakas- ja ohjelmistovaatimukset (Haikala & Mikkonen 2011, 62).

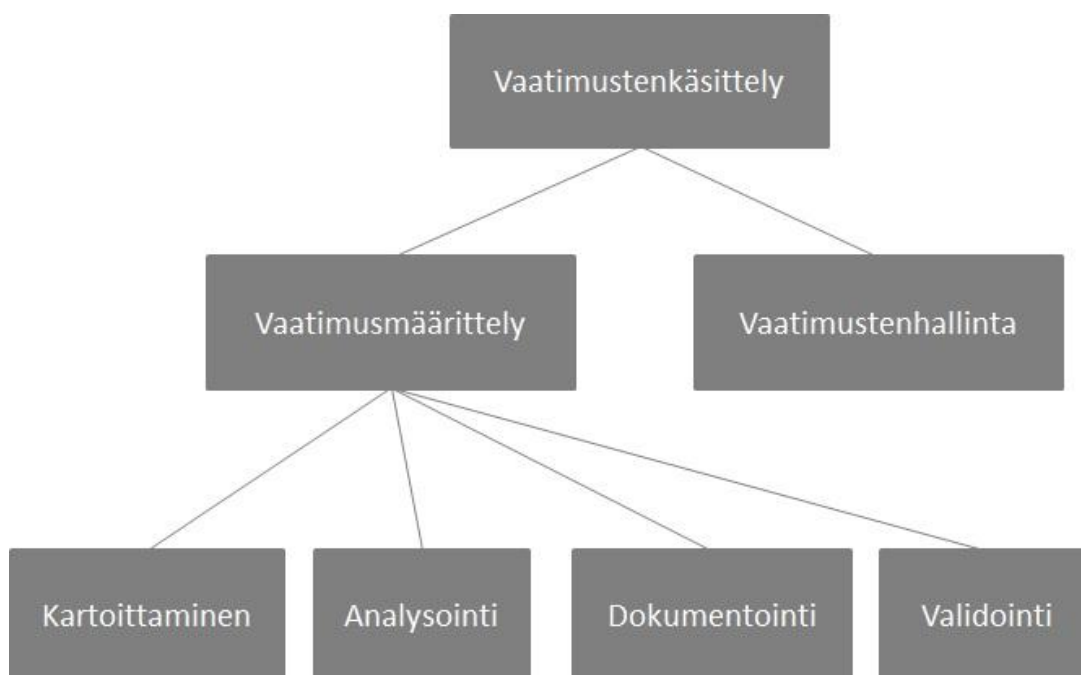
Asiakasvaatimukset tai ominaisuudet toteutetaan ohjelmistovaatimuksien avulla (kuvio 1). Haikalan ja Mikkosen ohjeistavat kirjamaan ohjelmistovaatimukset toiminnalliseksi määrittelyksi. Ohjelmistovaatimukset kuvaavat ohjelman toimintoja, joiden avulla asiakasvaatimukset tuotetaan tai tarjoillaan loppukäyttäjälle. Ohjelmiston toteuttamista varten ohjelmistovaatimukset kuvataan joukkona teknisiä vaatimuksia. Paakki taas jakaa toimintaympäristön vaatimukset järjestelmävaatimuksiksi ja rajapinnan vaatimukset ohjelmistovaatimuksiksi. (Haikala & Mikkonen 2011, 61–63; Paakki 2011.)

### 2.3 Vaatimusten käsittelyn osa-alueet

Wiegertsin mukaan vaatimusten käsittely on suurimmaksi osaksi kommunikaatiota. Kottonya ja Sommerville korostavat, että vaatimusten käsittelyn on sisällettävä erilaisia aktiviteetteja, joiden tavoitteena on ongelmien ja erimielisyyksien ratkominen. Enders ja Rombach näkevät vaatimusten käsittelyn oppimis- ja neuvotteluprosessiksi. Vaatimusten avulla opitaan liiketoiminnan prosesseista ja miten niitä on tarvetta tukea. Neuvot-

telutaitoja tarvitaan tilanteissa, joissa eri käyttäjäryhmien vaatimukset ovat jopa toistensa vastakohtia tai toiset poissulkevia. (Enders & Rombach 2003, 17; Kotonya ja Sommerville 2002, 7; Wiegers 2000.)

Haikala & Mikkonen kuvaavat vaatimusten käsittelyn prosessin kuvion 2 mukaisesti. Vaatimusten käsittely sisältää vaatimusmäärittelyn aktiviteetteineen sekä vaatimushallinnan (Haikala & Mikkonen 2011, 65.)



Kuvio 2 Vaatimusten käsittelyn osa-alueet (Haikala & Mikkonen 2011, 65).

**Vaatimusmäärittely** on osa jokaista ohjelmistotuotantoprojektia ja se yhdistää kaikki ohjelmistotuotannon aktiviteetit ja tehtävät. Vuoren määrittelyn mukaan vaatimusmäärittelyllä on useita olemuksia; vaatimukset kuvaava dokumentti, vaatimusten syntyyn johtava prosessi, yhteinen ymmärrys kehitettävästä kohteesta ja yhteinen ymmärrys vaatimuksista. Vaatimusmäärittelyyn tulisi aina lähteä selkeän konseptin kanssa. Mikäli järjestelmää kehitetään ilman yhteistä näkemystä lähtö- tai tavoitetilasta, on yhtenäiseen ja toimivaan lopputulokseen lähes mahdotonta päästä. (Vuori 2009, 20.)

Paakin mukaan vaatimusmäärittelyssä vastataan kysymyksiin; Mitä halutaan, Miksi halutaan ja Kuka ottaa vastuun? Vaatimusmäärittelyssä selvitetään ehdot, joilla ohjelmiston tulee toimia sen toimintaympäristössä. Lisäksi selvitetään ohjelmiston ja toimintaympä-

ristön rajapinta eli miten kommunikaatio ohjelmiston ja sen toimintaympäristön välillä toteutetaan. Vuori muistuttaa, että hyvän vaatimusmäärittelyn tulee jättää ratkaisun suunnittelulle tilaa ja lyödä lukkoon vain olemassa olevat reunaehdot, joita ei voi muuttaa. (Paakki 2011; Vuori 2009, 21.)

Vaatimusmäärittely aloitetaan vaatimusten kartoittamisella. Menetelminä voidaan käyttää käyttäjien ja sidosryhmien haastatteluita, aivoriihiä ja työpajoja. Vaatimusmäärittelyn on pohjauduttava liiketoiminnan, prosessien sekä käyttäjien tarpeiden ymmärtämiseen. Vuori kehottaa aloittamaan vaatimusmäärittelyn riskianalysillä, jonka avulla tunnistetaan järjestelmän kriittiset piirteet. Näin suunnittelussa ja vaatimuksissa osataan panostaa tärkeisiin osiin. (Haikala & Mikkonen 2011, 66–67; Vuori, 2009, 21.)

Vaatimusten analysoinnissa tarkennetaan vaatimuksia sekä selvitetään niiden keskinäiset suhteet ja prioriteetti. Hamlet ja Maybee näkevät vaatimusten analyysin eli selvityksen mitä ohjelmiston on tehtävä, kehittämisprosessin tärkeimpänä vaiheena. Vuori korostaa, että asiakkaan vaatimuksia täytyy arvioida, täydentää tai laajentaa tarvittaessa sekä ottaa mukaan uusia näkökulmia. Vasta tämän jälkeen voidaan puhua vaatimusmäärittelystä. Järjestelmäkehittämisessä on selvitettävä käyttäjien tarpeet, jotta kehitettävä järjestelmä tukee käyttäjien tarpeita sekä soveltuu saumattomasti käyttäjien arkeen. Järjestelmäkehittämisessä vaatimusmäärittelyn ongelmana on usein vaatimusten jäsentäminen vain yhdestä näkökulmasta. (Haikala & Mikkonen 2011, 66–67; Hamlet & Maybee 2002, 119; Vuori, 2009, 20 - 21.)

Vaatimusmäärittelyn tuotoksena syntyy vaatimusmäärittely-dokumentti tai dokumentaatio, johon vaatimukset on koottu. Vaatimukset dokumentoidaan vaatimusten kartoittamisesta lähtien sovitulla tavalla dokumenttiin, esimerkiksi Excel-taulukkoon. Haikala ja Mikkonen suosittelevat sisällyttämään dokumentaatioon sekä asiakas-, että ohjelmistovaatimukset. Paraskaan mallipohja ei auta vaatimusten dokumentaatioon tai sisällön luomiseen. Vaatimusten luomisen ja kirjoittamisen taidon oppii parhaiten yrityksen ja erehdyksen kautta. Hamlet ja Maybee ohjeistavat kuvaamaan dokumentaatioissa mitä on tehtävä, ei sitä, miten asiat tulee tehdä. Robertson ja Robertson näkevät, että vaatimus kannattaa kuvata mahdollisimman ylätasolla, sillä silloin se on mahdollisimman kaukana ratkaisusta ja jättää ratkaisutavan avoimeksi. Vaatimusmäärittelyn yhtey-

dessä vaatimusten dokumentoinnin lisäksi voidaan luoda jo alustava käyttöohje sekä testaussuunnitelma. (Haikala & Mikkonen 2011, 66–68; Hamlet & Maybee 2002, 154; Pohjonen 2002, 28; Robertson & Robertson 2011, 274.)

Haikala ja Mikkonen muistuttavat, että dokumentoinnissa voidaan hyödyntää vaatimustenhallintaan luotuja työkaluja tai järjestelmiä, joissa voidaan määritellä vaatimuksen dokumentaation pohja sekä toteutuksen työprosessi. Vuoren mukaan vaatimustenhallintatyökalut taas eivät anna parasta mahdollista tukea vaatimusten luomisvaiheessa. Työkalu voi jopa haitata ongelman kokonaiskuvan havaitsemista, mikäli työkalun hierarkiarakenteet piilottavat konseptin ja vision. Robertson ja Robertson korostavat, että työkalujen on hyvä muistaa olevan tuki omalle työlle, ei oman työn korvaaja. (Haikala & Mikkonen 2011, 66–68; Robertson & Robertson 2011, 347; Vuori, 2009, 21.)

Vaatimuksista dokumentoidaan useita eri tietoja, joilla on eri käyttö ja hyödynnystarkoituksia. Vaatimuksista dokumentoitavia perustietoja ovat vaatimuksen luontipäivä, tekijä eli vaatimuksen kirjaaja ja asiakas eli mistä tai keneltä vaatimus on saatu. Vaatimukseen on hyvä lisätä tyyppitieto eli onko vaatimus lisäys, muutos tai korjaus. Vaatimuksen kuvauksessa voidaan käyttää apuna käyttötapauksia, käyttäjätarinoita, aktiviteettikaavioita tai tapahtumasekvenssikaavioita. Vaatimuksen suhde muihin vaatimuksiin eli miten se vaikuttaa tai miten se liittyy niihin sekä vaatimuksen tarpeellisuus on hyvä dokumentoida. Näiden tietojen avulla pystytään helpommin tekemään päätöksiä vaatimusten toteuttamisesta. Vaatimuksen pysyvyys tai muutosherkkyys ja testattavuus eli vaatimuksen täyttymisen testaussuunnitelma on hyvä kirjata ylös. Tärkeä vaatimuksen lisätieto on aika-arvio eli vaatimuksen toteuttamisen alustava työmääräarvio. Dokumentoitavien tietojen määrä riippuu sovitusta käytännöistä sekä sovelluksen luonteesta. (Haikala & Mikkonen 2011, 64–65.)

Sovitulla tavalla dokumentoidut vaatimukset validoidaan. Validointi eli kelpoistaminen on perinteisesti toteutettu katselmoimalla vaatimusdokumentaatio asiakkaan kanssa. Robertson ja Robertson korostavat, että validoinnin onnistumiseen vaikuttaa kieli, jolla vaatimukset on kirjoitettu. Asiakas voi hyväksyä vain vaatimukset, jotka on kirjoitettu asiakkaan ymmärtämällä tavalla. (Haikala & Mikkonen 2011, 67; Robertson & Robertson 2011, 26.)



**Vaatimustenhallinta** on vaatimusmäärittelyn ohella toinen vaatimuskäsittelyn osa-alue. Ohjelmiston kehitysprosessille on tyypillistä, että järjestelmälle alussa asetetut vaatimukset muuttuvat ja lisääntyvät työn edetessä. Kehitystyö etenee harvoin suoraviivaisesti stabiileista asiakasvaatimuksista vaatimukset täyttävään järjestelmään. Pohjosen kokemuksen mukaisesti käytännössä kehitystyön esitutkimusvaiheessa tehty vaatimusmäärittely laajenee koko sovelluksen elinkaaren aikaiseksi vaatimustenhallinnaksi. Pohl ja Ruppin näkevät järjestelmän elinkaaren aikaiset uudet vaatimukset ja vaatimusten muutokset elinkaareen kuuluviksi ja korostavat, että ne eivät ole negatiivinen asia. Käytettävää järjestelmää kehitetään ja ennen kaikkea tullaan muuttamaan. Mikäli muutoksen aikana ei kiinnitetä aktiivisesti huomiota kompleksisuuden vähentämiseen, kasvaa järjestelmän kompleksisuus huomauttavat Endres ja Rombach. (Endres & Rombach 2003, 163 – 167; Pohl & Rupp 2011, 131; Pohjonen 2001, 81.)

Haikalan ja Mikkosen mukaan vaatimustenhallinnan keskeinen prosessi on vaatimusmuutosten hallinta, johon kuuluu muutospyyntöjen tekeminen, niiden analysointi, testaus sekä hyväksyminen. Muutosprosessissa vaatimukset käsitellään ja hyväksytään vaatimusmäärittelyn tapaan. Jokaisessa kehittämistyössä on oltava sovittu sekä yhteisesti käytettävä tapa muutosten hallintaan. Lisäksi on kuvattava prosessi, jonka mukaisesti toiveet hyväksytään toteutettaviksi muutoksiksi. Robertsonin ja Robertsonin mukaan muutostenhallinnassa on olennaista hallita miten ja milloin muutokset toteutetaan. Oikea ajoitus ja muutoksen kannattavuus on ensiarvoisen tärkeää ohjelmiston arvon kohottamisessa. Onnistuneen muutostenhallinnan takaa toimiva palautekanava, jota kautta loppukäyttäjät voivat antaa palautetta. Toimiva palautekanava auttaa näkemään muutoksen tarpeen ja näin ohjelma saadaan pidettyä ajan tasalla. (Haikala & Mikkonen 2011, 67; Robertson & Robertson 2011, 358 – 359.)

Haikala ja Merijärvi määrittävät vaatimustenhallinnan tavoitteeksi asiakasvaatimukset ja – sopimukset täyttävän ohjelmiston toteutuksen. Vuori ja Wieggers näkevät vaatimustenhallinnan työkaluista olevan tukea juuri muutoksenhallintaan ja vaatimustenhallintaan. Vaatimusten käsittely on toimintona hyvin lähellä ohjelmiston tuotteenhallintaa ja työnjako niiden välillä on määriteltävissä monin eri tavoin. Haikala ja Mikkonen määrittävät vaatimuksen tilan hallinnan, vaatimusten välisten riippuvuuksien seurannan tai jäljitettävyyden ylläpidon sekä raportoinnin vaatimustenhallinnan tehtäviksi. Haikalan ja

Merijärven mukaan vaatimustenhallintaan lasketaan kuuluvaksi kaikki toimenpiteet joilla varmistetaan, että lopputuote vastaa asiakkaan vaatimuksia. (Haikala & Mikkonen 2011, 67; Haikala ja Merijärvi 2002, 93; Vuori 2009, 21; Wiegers 2000.)

Vaatimusmäärittelyn dokumentaatiota on ylläpidettävä vaatimustenhallinnan aikana. Hamlet ja Maybee näkevät vaatimusmäärittelyn dokumentaation epätäydelliseksi aina, kun siitä huomataan puuttuvan jotain ohjelmiston elinkaaren aikana. Robertson ja Robertson korostavat vaatimusdokumentaation olevan aina luontihetkensä kuvaus. (Hamlet & Maybee 2002, 154; Robertson & Robertson 2011, 357.)

Vaatimusten käsittelyn prosessia, kuten kaikkia prosesseja, on tärkeää pyrkiä parantamaan. Robertson ja Robertson suosittelevat prosessin työkulun parantamisen työkaluksi vaatimusten retrospektiiviä, jonka he näkevät yhdeksi toimivimmista menetelmistä hyvien ja huonojen käytäntöjen ja tapojen poimimiseen prosessista. Retrospektiivissä prosessiin osallistuneilta henkilöiltä kysytään kolme kysymystä; mikä onnistui, mikä epäonnistui ja mitä tekisin toisella tavalla. Näiden kysymysten rehellisten vastausten ja niiden analyysin perusteella prosessia on mahdollista parantaa. (Robertson & Robertson 2011, 31.)

## **2.4 Vaatimusten jäljitettävyys**

Vaatimusten jäljitettävyydellä ymmärretään asiakasvaatimusten seuranta koko niiden elinkaaren ajan. Jäljitettävyys voidaan tehdä eteenpäin (eteenpäin jäljitettävyys) asiakasvaatimuksen määrittelystä toteutukseen. Taaksepäin jäljitettävyys on taas vaatimuksen seuranta takaperin toteutuksesta asiakasvaatimukseen. Jäljitettävyys vaatii asiakasvaatimusten dokumentoinnin, asiakasvaatimusten ja järjestelmän riippuvuuksien dokumentaatiota ja ylläpitoa sekä asiakasvaatimusten välisten yhteyksien ja riippuvuuksien ylläpitoa. (Haikala & Mikkonen 2011, 63; Robertson & Robertson 2011, 353 – 356.)

Jäljitettävyys mahdollistaa muutostenhallinnassa ohjelmistoon kohdistuvien muutostarpeiden suunnittelun, niiden riippuvuuksien sekä toteuttamisen tai toteuttamatta jättämisen säteilyvaikutusten seurannan. Äärimmilleen vietyinä jäljitettävyyden avulla voidaan tunnistaa yksittäisestä vaatimuksesta polku dokumentaation kautta ohjelmakoodiin asti

tai yksittäisestä koodin osasta voidaan luoda linkitys siihen liittyviin vaatimuksiin. Tosin tähän päästä vain, mikäli dokumentaatiossa on ylläpidettynä paljon työtä vaativa jäljitettävyyssmatriisi. (Haikala & Mikkonen 2011, 63.)

Vaatimusten jäljitettävyys saavutetaan Robertsonin ja Robertsonin mukaan, kun jokainen järjestelmän osa voidaan yhdistää vaatimukseen. Toisaalta järjestelmässä ei pidä olla yhtään toiminnallisuutta, johon ei ole vaatimusta. Ohjelmistokehittämisen toteuttamisen vaiheessa ei hyvän vaatimustenhallinnan periaatteen mukaan tehdä vaatimuksiin kuuluvia päätöksiä toteutuksesta. (Hamlet & Maybee 2002, 119; Robertson & Robertson 2011, 353.)

## **2.5 Vaatimusten jatkotyöstäminen**

Vaatimusten jatkotyöstäminen ohjelmistokehittämisessä tarkoittaa spesifikaation laatimista. Teknisesti katsottuna ohjelmistojen kehittäminen on dokumentaation tuottamista ja tuotantoprosessi on kuvaus vaatimusten toteutuksesta. Tuotantoprosessissa edetään tasolta toiselle tuottamalla syötteeksi spesifikaatiodokumentti. Ylimmällä tasolla spesifikaatiodokumentti on laadittu luonnollisella kielellä ja alimmalla tasolla ohjelmistokielellä. (Haikala & Merijärvi 2003, 61.)

Aiemmin todettiin, että määrittely vastaa kysymykseen ”mitä” ja suunnittelu kysymykseen ”miten”. Toteutukseen liittyvät päätökset kannattaa tehdä vasta suunnittelun vaiheessa. Mitä aikaisemmin ja vähäisemmin tiedoin toteutuksen päätökset tehdään, sen todennäköisemmin niitä joudutaan muuttamaan. (Haikala & Merijärvi 2003, 62.)

## **2.6 Ohjelmien takaisinmallinnus**

Ohjelmien takaisinmallinnuksessa on tavoitteena ymmärtää mitä olemassa olevassa ohjelmassa tehdään. Takaisinmallinnuksessa keskeistä on tunnistaa ohjelman osat, selvittää ohjelman välisten osien suhteet ja luoda korkean tason kuvaus ohjelmasta. Takaisinmallinnus on tutkimusta, ei muuttamista. Harsu listaa takaisinmallinnuksen tehtäviksi dokumentaation lukeminen, lähdekoodin tutkiminen, järjestelmän ajaminen, käyttäjien ja kehittäjien haastattelu, testitapausten koodaaminen ja ajaminen, ajojälkien tuottaminen sekä versiohistorian analysointi. Endersin ja Rombachin mukaan

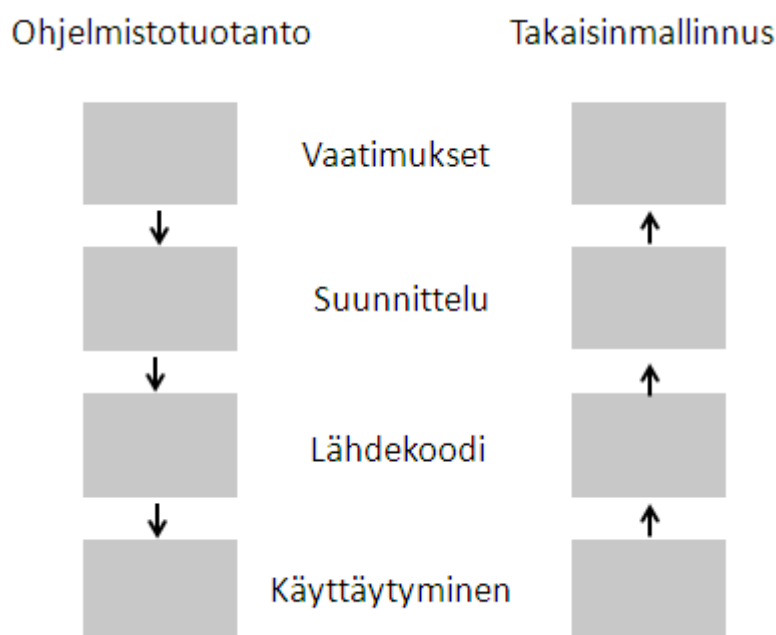
takaisinmallinnus sisältää kaikki toimenpiteet, joiden avulla pyritään parantamaan kehittämiseen liittyvän laadun kriteereitä kuten testattavuutta, ylläpidettävyyttä ja uudelleenkäytettävyyttä. Enders ja Rombach sekä Schwartz näkevät takaisinmallinnuksen ohjelmiston, systeemin tai teknisten komponenttien paranteluna ja päivittämisenä uudelle tasolle tai esineen purkamisena, jotta sitä voidaan parannella. (Enders & Rombach 2003, 176; Harsu 2012, 4; Schwartz 2001.)

Takaisinmallinnuksella pyritään hallitsemaan ohjelmiston monimutkaisuutta, tarjoamaan jatkossa vaihtoehtoisia näkemyksiä, mahdollisesti jäljittämään kadonneita tietoja, paljastamaan sivuvaikutuksia, visualisoimaan ja siirtymään lähemmäksi sovellusalueen käsitteitä. Lisäksi takaisinmallinnuksella mahdollistetaan ohjelmiston osien uudelleenkäyttö. Harsu määrittää takaisinmallinnuksen osa-alueiksi ohjelmiston suunnitteluratkaisun jäljittämisen, ohjelmiston uudelleendokumentoinnin ja rakenteellisen uudelleendokumentoinnin. Uudelleendokumentoinnissa tuotettava dokumentti voi olla tekstiä, hypertekstiä, ristiviittauksia tai kutsukaavioita, graafisia esityksiä, muistiinpanoja tai muuta kuvausta ohjelman käyttäytymisestä. (Harsu 2012, 5 – 12; Harsu 2003, 117.)

Takaisinmallinnuksen tavoitteena on tuottaa käsitys järjestelmästä sen ylläpitoa ja jatkokehittämistä varten. Takaisinmallinnuksen osa-tavoitteita on järjestelmän monimutkaisuuden hallinta uusia työkaluja ja menetelmiä kehittämällä sekä vaihtoehtoisten näkymien tarjoaminen ja järjestelmän kuvaaminen uudella ymmärrettävämällä tavalla, kuten tietokaavioina tai rakennekarttana. Lisäksi takaisinmallinnuksessa pyritään jäljittämään järjestelmästä kadonneita tietoja puuttuvien dokumenttien uudelleen kirjoittamisen muodossa. Tavoitteena voi olla myös ohjelmiston kehittämisen myötä syntyneiden sivuvaikutusten paljastaminen tai virheiden ennakointi. Harsu näkee takaisinmallinnuksen ohjelmien ylläpidon, uudistamisen ja parantamisen edeltävänä, ohjelmaa analysoivana vaiheena. (Harsu 2003, 119–120.)

Kuviossa 3 on verrattu takaisinmallinnusta ja ohjelmistotuotannon prosessia. Takaisinmallintamisessa lähdetään tutkimaan ohjelmiston käyttäytymistä analysoiden ja pyritään ymmärtämään ohjelmaa. Tavoitteena on ohjelman osien sekä niiden välisten suhteiden ymmärtäminen. Apuna käytetään lähdekoodia. Näillä toimenpiteillä pyritään selvittämään ohjelman suunnitteluratkaisut ja samalla selvitetään ohjelmiston toiminnal-

lisuus. Suunnittelun kautta päästään kiinni alkuperäisiin vaatimuksiin. Takaisinmallinnuksessa edetään ohjelmistokehityksen prosessiin nähden päinvastaisessa suunnassa ja sitä kutsutaan myös käännteistekniikaksi. (Harsu 2012, 2.)



Kuvio 3 Ohjelmistotuotanto vs. takaisinmallinnus (Harsu 2012, 1).

Takaisinmallinnuksessa on tärkeää saavuttaa sopiva tarkkuustaso. Harsu kuvaa takaisinmallinnusmallin hyviksi ominaisuuksiksi kuvauksen selkeyttä sekä prosessin kuvaamista. Kuvauksessa voi olla analyysi ratkaisun hyödyistä, kustannuksista tai seurauksista. Lisäksi voidaan arvioida miten ratkaisu sopii olemassa olevaan järjestelmään. Hyviä ominaisuuksia on myös prosessin kuvaaminen eli miten tilanteesta toiseen päästään. Oleellista takaisinmallinnuksen kannalta ei ole arvioida kuinka elegantly toteutettu ratkaisu on. (Harsu 2012, 40.)

Takaisinmallinnuksessa syntynyt kuvaus mahdollistaa ohjelman jatkoanalyysin. Takaisinmallinnuksen tavoitteena ei ole muuttaa ohjelman toimintaa, varsinkaan jos ohjelma toimii halutulla tavalla. Hyvällä ohjelmasuunnittelulla ei voida poistaa takaisinmallinnuksen tarvetta, sillä uudet vaatimukset asettavat uusia tarpeita ohjelmille. Schwartzin mukaan takaisinmallinnusta käytetään oppimisen apuvälineenä, uuden ehkä halvemmän version kehittämisessä markkinoilla olevasta tuotteesta, ohjelmistojen kehittämisessä ja parantamisessa, tiedon siirtämisessä eri käyttöjärjestelmien tai tietokan-

tojen välillä sekä dokumentoinnin paikkailussa tai tekemisessä jälkikäteen. (Harsu 2003, 117; Schwartz 2001.)

## 2.7 Vaatimusmäärittelyn merkitys

Robertson ja Robertson näkevät, että vaatimustenmäärittäminen on ainut keino luoda yhteinen ymmärrys ohjelmistoa koskevista tavoitteista ja tarpeista. Vaatimusmäärittelyn avulla eri sidosryhmät ymmärtävät ja voivat hyväksyä ohjelmaa koskevat vaatimukset. Vaatimukset ovat suunnittelun ja toteutuksen ohjenuora sekä testauksen perusta. Valmiin ohjelman testaus onnistuu ja on helpompaa, mikäli vaatimusmäärittely on tehty huolella. (Robertson & Robertson 2011, 26).

Viime vuosikymmenen aikana organisaatiot ovat havainneet erilaisia ongelmia, kuten käyttämättömiä sovelluksia, toteuttamattomia kehitysideoita ja tyytymättömiä käyttäjiä, jotka tuntevat sovellusten ennemmin hankaloittavan kuin helpottavan työntekoa. Yhteistä ongelmille on, että ne kumpuavat huonoista vaatimusmäärittelyistä ja vaikuttavat organisaation sovelluksen avulla saavuttamaan lisäarvoon. Sovelluksen elinkaaren vaiheista vaatimusmäärittelyissä luodaan sovelluksen arvo. Vaatimusmäärittelyiden perusteella määritetään sovelluksen arvo organisaatiolle, miksi sovellus kannattaa rakentaa ja sen kehittämiseen panostaa. Mikäli arvoa ei määritetä vaatimusmäärittelyn aikana, on sovelluksen suunnittelussa ja toteutuksessa suurempi epäonnistumisen riski. Arvo-ketju vaatii yhteisen ymmärryksen miksi sovellusta kehitetään, kenelle sitä kehitetään sekä mitä hyötyjä sen avulla saavutetaan. (Grant, 2012.)

Haikalan ja Mikkosen mukaan ohjelmistokehittämisessä projektien epäonnistuminen johtuu yli 60-prosenttisesti huonosta vaatimusmäärittelystä. Yksi onnistuneen ohjelmistoprojektin perusedellytyksistä on hyvin hoidettu vaatimusten käsittely. Robertson ja Robertson tuovat esiin faktan, että ohjelmistokehittämisessä havaitut virheet ovat 50 – 60 % peräisin vaatimusten virheistä. Vain noin 5 % virheistä johtuu toteutuksesta. Tarkkuudessa vaatimusten kanssa saavutetaan merkittäviä hyötyjä ja kustannustehokkuutta korjattavien virheiden määrän laskiessa. Mitä aikaisemmassa vaiheessa virheet havaitaan, sitä halvemmaksi niiden korjaus tulee. (Haikala & Mikkonen 2011, 61; Robertson & Robertson 2011, 262.)

Uuden teknologian ja sovellusten tuomat muutokset ihmisten väliseen kommunikaatioon ovat vaikuttaneet myös vaatimustenhallintaan. Vaatimuksia tulee esiin uusista, innovatiivisista lähteistä ja tämä pakottaa vaatimusmäärittelyssä käytettäviä toimintatapoja ja tekniikkaa uudistumaan. Vaatimusten analysointia tai niistä tehtävää arvon määrittystä ei voida tehdä enää perinteisin tavoin keskustelujen avulla. Analysointi onnistuu vain, mikäli todella ymmärtää vaatimuksen takana olevat tarpeet. (Higgins 2011.)

Sähköiset raportointityökalut toteutetaan johdon ja hallinnon tarpeiden mukaisesti ja loppukäyttäjänäkökulma jää määrittelyissä usein huomioimatta. Tietojärjestelmät toimivat hyvin yrityksen prosessien mukaisesti, mutta käyttäjien näkökulmasta järjestelmien käytettävyyttä ei saa yhtä hyvää arvosanaa. Systemien inkonsistenssi eli eri käyttöjärjestelmien epäyhtenäisyys, heikentää tietojärjestelmistä saatavaa tehokkuutta. Tietojen epäyhtenäisyyttä voidaan havaita myös tietojärjestelmän sisällä, esimerkiksi päivämääräkirjaamismuodossa tai valintatavoissa. Kognitiiviseen ergonomiaan perustuvien laskelmien mukaan käyttäjän tekemien virheiden määrä kasvaa neljä prosenttia ja työaika kuluu viidennes enemmän epäyhtenäisten tietojen vuoksi. Kognitiivisessa ergonomiassa selvitetään ihmisen tiedonkäsittelyyn liittyviä lainalaisuuksia, jotka vaikuttavat järjestelmän käyttöön. Huonoa kognitiivista ergonomiaa on esimerkiksi liian iso tai liian pieni kirjainten koko. Onnistunutta kognitiivista ergonomiaa toteutetaan mikäli tietojärjestelmän eri käyttäjäryhmät, heidän tarpeet, käytön tavoitteet ja tehtävät huomioidaan suunnittelussa. Jo tietojärjestelmän hankinnan vaiheessa on selvitettävä nykyistä paremmin järjestelemän lopulliset käyttäjät, millaisiin tehtäviin järjestelmää käytetään ja miten järjestelmä muuttaa käyttäjien työtä. (Kupila 2013, 58 - 59.)

Tietojärjestelmän tarkoitus on tietojen käsittelyn avulla helpottaa ja tehostaa toimintaa tai tehdä se ylipäättään mahdolliseksi. Mikkonen haastaa IT-yksiköt panostamaan työn tuottavuuteen järjestelmien käytettävyyden ja asiakastytyväisyyden keinoin (Kupila 2013, 59).

### 3 Vaatimusten dokumentointi

Työn tavoitteena oli dokumentoida projekti- ja portfoliohallinnan ohjelmiston konfiguraatioita tai kustomointia aiheuttaneet toiminnalliset vaatimukset. Seuraavaksi kuvataan työn sisältöä sekä toteutusta ja esellään työn tuotokset. Myös valmisohjelmisto, johon kohdistuvat vaatimukset työssä dokumentoitiin, esitellään. Lisäksi analysoidaan työn taustamateriaalina ollutta ohjelmistotoimittajan toteuttamaa esiselvitystä ohjelmiston konfiguraatioista ja kustomoinneista.

Toiminnallisten vaatimusten dokumentoinnissa on pyritty korostamaan ohjelmiston eri käyttäjäryhmien ja ohjelmiston tukemien prosessien näkökulmia. Tavoitteeseen pyrittiin kuvaamalla vaatimukset prosesseittain ja käyttäjäryhmittäin. Usean eri prosessin tarpeen täyttämä vaatimus on kuvattu prosessin näkökulman mukaisesti erillisinä vaatimuksina, vaikka vaatimusten tekninen toteutustapa olisikin osittain yhteinen. Tavoitteena oli myös tuoda esiin ohjelmiston ylläpidon näkökulmasta erilaisten vaatimusten vaikutukset. Tästä syystä vaatimusten toteutustapa kuvattiin ohjelmiston konfiguraation ja kustomoinnin näkökulmasta. Edellä mainitut tavoitteet ovat tärkeitä, sillä tuotosten perusteella lukijalle pyritään luomaan kokonaisvaltainen ymmärrys ohjelmiston tukemista prosesseista sekä käsitys vaatimusten toteutuksen hinnasta ohjelmiston ylläpidossa.

Työn tavoitteena oli dokumentoida muutokset tavalla, joka toimisi kehityssuunnitelman pohjana ja jonka pohjalta voidaan lähteä analysoimaan ohjelmiston tulevaa kehittämistä. Työssä toteutettu yhteenveto voi toimia myös eroavaisuusanalyysin perustana, ja sen avulla voidaan lähteä selvittämään syitä kustomointeja edellyttäneiden toiminnallisten vaatimusten ja ohjelmiston ohjelmistotoimittajan julkaisemien perustoiminnallisuuksien välillä.



### 3.1 Vaatimusten dokumentoinnin suunnittelu ja toteutuksen aikataulu

Työn suunnittelu tapahtui kahdessa osassa, joiden välissä työn aihe tarkentui. Touku-kuussa 2012 toteutettavaksi sovittiin ohjelmiston kehittämissuunnitelma. Toimeksiantaja oli havainnut kehittämissuunnitelman tarpeen. Opinnäytetyön prosessi alkoi tuolloin ja kehittämissuunnitelma hyväksyttiin työn aiheeksi.

Syksyn 2012 aikana työn tekijä ja toimeksiantaja tarkensivat työn toimeksiantoa yhteisymmärryksessä. Alkaneen prosessin aikana oli todettu, että alkuperäisen toimeksiannon kehittämissuunnitelman toteutukseen tarvitaan parempi ymmärrys lähtötilanteesta. Ohjelmisto tuki useita eri liiketoimintaprosesseja ja tuki oli mahdollistettu osaltaan ohjelmiston konfiguraatioiden ja kustomointien avulla. Kehittämissuunnitelmassa haluttiin arvioida ja ottaa kantaa ohjelmiston rooliin eri prosessien tukena, sekä ohjelmiston hyödyntämisen tapaan ylläpidon näkökulmasta. Prosessien tuen vaatimat ohjelmiston konfiguraatiot ja kustomoinnit näkyivät ylläpidon työmäärissä ja kustannuksissa. Työn aiheen tarkennuksen tarve näkyi tekijän työssä syksyllä 2012 tavoitteiden epämääräisyytenä ja työn etenemisen hidastumisena.

Työn aihe tarkentui loppusyksyllä 2012. Toimeksiantajan ja työn tekijä valitsivat lähtötilanteen kuvauksen työn aiheeksi ja kuvaustavaksi valittiin ohjelmiston vaatimusten dokumentointi sekä vaatimusten ryhmittely niiden tukemien prosessien ja toteutustavan mukaisesti. Nykyisen ohjelmiston eri versioita oli hyödynnetty toimeksiantajan organisaatiossa lähes kymmenen vuotta ja ohjelmiston tukemien prosessien määrä oli kasvanut vuosien saatossa. Ohjelmiston kehittämisessä ei ollut järjestelmällisesti dokumentoitu liiketoiminnan vaatimuksia, toiminnallisia vaatimuksia tai niiden toteutuksesta aiheutuneita ohjelmiston konfiguraatioita ja kustomointeja. Työssä nähtiin tavoitteen saavuttamisen kannalta parhaaksi nykytilan kuvaustavaksi dokumentoida vaatimukset ja linkittää ne ohjelmiston konfiguraatioihin ja kustomointeihin. Tavoitteena oli tuottaa ohjelmiston kanssa toimiville tahoille yhteinen ymmärrys ohjelmiston nykytilasta. Lisäksi työssä pyrittiin tuottamaan materiaalia, jonka pohjalta ohjelmiston kanssa toimivilla tahoilla on mahdollisuus arvioida ohjelmistossa tuettuja prosesseja. Materiaalin pohjalta tehtävässä arvioinnissa ei vaadita teknistä ymmärrystä itse ohjelmistosta tai proses-

sien tukemisen toteutustavan vaikutuksia. Näin mahdollistettiin työ kehityssuunnitelman toteutuksessa, joka onkin työ jatkokehitysehdotuksena.

Aiheen tarkennuksen jälkeen työn toimeksiannossa tuotokseksi sovittiin liiketoiminnan ja toiminnallisten vaatimusten dokumentointi, sekä vaatimusten yhteenveto. Työn tietopohjan työstö ajoittui kesän 2012 ja alkuvuoden 2013 välille. Ensin työssä keskityttiin tietojärjestelmän kehittämisen ja ylläpidon osuuteen, joka oli relevanttia sen hetkiseen kehittämissuunnitelman toimeksiantoon. Toimeksiannon muuttuessa tietopohjassa keskityttiin vaatimusten käsittelyyn. Vaatimusten käsittelyn tietoperusta kirjoitettiin tammikuussa 2013. Työn tuotoksen toteutus ajoittui helmi - maaliskuuhun 2013, eli tietopohjan kirjoittamisen jälkeiseen ajanjaksoon. Tietopohja viimeisteltiin tuotoksen toteutuksen aikana. Tuotos luovutettiin toimeksiantajalle huhtikuussa 2013. Toteutuksen tarkempi projektisuunnitelman tehtävälista ja aikataulu on työn liitteenä (liite 2).

### **3.2 Valmisohjelmisto ja sen hyödyntäminen**

Työn toteutuksen kohteena oli toimeksiantajan organisaation käytössä oleva projekti- ja portfoliohallinnan valmisohjelmisto. Ohjelmisto on erilaisista palveluista ja työkaluista koottu sovellusympäristö, joka on luotu tukemaan organisaation päätöksentekoa, parantamaan tuottavuutta ja tehokkuutta sekä vastaamaan muuttuviin liiketoiminnan vaatimuksiin. Ohjelmisto rakentuu toiminnallisista moduuleista. Moduulirakenne mahdollistaa ohjelmiston käytön ja/tai käyttöönoton osissa tai vaiheittain.

Toimeksiantajan organisaatiossa on käytössä vain osa ohjelmiston moduuleista. Ohjelmisto tukee useita eri liiketoimintaprosesseja; vuosisuunnittelua, hankehallintaa, projektinhallintaa, resurssienhallintaa, työajan seurantaa sekä henkilötyön laskutusta. Ohjelmiston avulla resurssit ja investoinnit kohdistetaan organisaation tavoitteiden mukaisesti sekä mahdollistetaan todelliseen tilanteeseen pohjautuva päätöksenteko. Ohjelmistosta saatavan tiedon avulla saadaan määritettyä kuinka toteutettavat projektit ja resurssit tukevat organisaation strategisia tavoitteita. Tiedon avulla nähdään miten hyvin työaika on hyödynnetty tehtävässä työssä ja näin saadaan tukea tulevaa suunnitteluun. Resurssienhallinnan avulla mahdollistetaan henkilökapasiteetin suunnittelu suhteessa tulevien investointien ja projektien kapasiteetti-vaatimuksiin. Projektienhallinnan eri toiminnalli-

suuksien avulla projektien hallinta saadaan seuraamaan parhaita käytäntöjä ja näin tuetaan projektipäälliköitä mahdollisimman hyvien projektisuunnitelmien luonnissa sekä projektien ohjauksessa. Lisäksi ohjelmistoa hyödynnetään eri toimintojen automatisoinnissa, optimoinnissa sekä liiketoimintaprosessien läpiviennin standardoinnissa. Projekteilla tehdyn työn kustannukset saadaan kohdistettua ohjelmiston avulla työn tilaajalle.

**Ohjelmiston konfiguraatiot ja kustomoinnit** olivat työssä tarkastelun kohteena osana vaatimusten dokumentaatiota. Työssä hyödynnettiin ohjelmistotoimittajan toteuttamaa esiselvitystä ohjelmiston konfiguraatioista ja kustomoinneista. Esiselvityksessä ohjelmistotoimittaja jaottelee sovellukseen tehdyt muutokset konfiguraatioihin tai kustomointeihin riippuen muutoksen luonteesta ja sen toteutustavasta. Ohjelmiston konfiguraatiot ja kustomoinnit ovat ohjelmiston ylläpidon kannalta merkittäviä käsitteitä, sillä niillä on erilaiset vaikutukset ohjelmiston versionvaihtoon ja tätä kautta ylläpidon kustannuksiin.

Konfiguraatio on ohjelmistotoimittajan tukema muutos. Ohjelmistotoimittaja määrittelee kustomoinneiksi sellaiset lisäykset ohjelmiston peruskoodiin, jota ei ole suunniteltu ohjelmiston perustoiminnallisuuksiin tai se ei ole ohjelmistotoimittajan julkaisema ohjelmiston osa. Konfiguraatioiksi luokitellut muutokset eivät aiheuta erillisiä toimenpiteitä versionvaihdossa ja ne säilyvät versionvaihdossa. Kustomoinnit eivät ole tuettuja versionvaihdossa ja ne voivat joko hankaloittaa versionvaihtoa tai kadota versionvaihdossa. Esiselvityksessä toimeksiantajan ohjelmistoon tekemät muutokset on jaettu viiteen eri ohjelmiston toiminnalliseen kokonaisuuteen, osaohjelmistoon. Osaohjelmistot ovat laskutus, hankkeiden budjetit, muut kuin henkilötyön kustannukset, resursointikonaisuudet sekä kustannusarviot. Osaohjelmistot on esitelty tarkemmin liitteessä 3. Jokainen osa-ohjelmisto on toteutettu muutostoimenpiteillä, jotka sisältävät sekä konfigurointia että kustomointia. Osaohjelmien avulla valmisohjelmisto on muokattu toimimaan toimeksiantajan prosessien mukaisesti.

### 3.3 Vaatimusten dokumentoinnin toteutus

Vaatimusdokumentaation toteutukseen kuuluvassa vaatimusten kuvauksessa selvitettiin ehdot, jotka ovat määrittäneet ohjelmiston toimimaan, kuten se toimii. Vaatimusmäärittelyssä vastattiin kysymyksiin mitä haluttiin ja miksi haluttiin, menneessä muodossa.

Ohjelmiston kehittämisessä vaatimusten käsittelyn prosessin vaatimusmäärittelyssä vastataan kysymyksiin tulevassa muodossa; Mitä halutaan, Miksi halutaan ja Kuka ottaa vastuun? Työn vaatimusmäärittelyssä ei otettu kantaa tekniseen ratkaisuun, vaan kuvattiin todellinen tarve, joka löytyi vaatimusten takaa. Näin ratkaisu pyrittiin jättämään vaatimusten ulkopuolelle, kuten 2.3 Vaatimusten käsittelyn osa-alueet aluvussa ohjeisteltiin hyvässä vaatimusmäärittelyssä tehtävän. Vaatimuksista työssä kuvattiin tarve ja reunaehdot, jotka ovat ohjanneet ja rajanneet teknistä toteutusta.

Työssä dokumentoitiin ohjelmiston toiminnalliset vaatimukset, jotka aiheuttivat konfiguraatioita tai kustomointia ohjelmistoon. Toiminnalliset vaatimukset linkitettiin liiketoimintavaatimuksiin. Työssä ei dokumentoitu ei-toiminnallisia vaatimuksia. Vaatimusten dokumentointi edellytti työn tekijältä vaatimusmäärittelyn aktiviteeteista vaatimusten kartoittamisen, dokumentoinnin sekä validoinnin toteuttamista.

Vaatimusmäärittely aloitettiin ohjelmiston tukemien prosessien ja sitä kautta niiden vaatimusten kartoittamisella aluvun 2.3 Vaatimusten käsittelyn osa-alueet ohjeistuksen mukaisesti. Kartoituksessa kuvattiin toimeksiantajan organisaation kehittämisen prosessit sekä niiden yhteydet projekti- ja portfoliohallinnan ohjelmistoon. Kartoituksen tuloksena kehittämisen prosessit jaoteltiin kolmeen eri luokkaan sen suhteen miten ne toimivat tai hyödyntävät ohjelmistoa ja sen tietoa. Jaottelun luokat olivat

- prosessi osaksi tai kokonaan ohjelmistossa,
- prosessit, joissa hyödynnetään ohjelmiston tietoa,
- prosessit, jotka ovat ohjelmiston ulkopuolella.

Kartoituksessa ei otettu kantaa miten tai mitä kautta prosessit, jotka hyödyntävät ohjelmiston tuottamaa tietoa, saavat tiedon tai missä ohjelmiston ulkopuolella toimivat prosessit toimivat.

Kartoituksen avulla saavutettiin kokonaiskuva projekti- ja portfoliohallinnan ohjelmistossa toimivista prosesseista ja sen pohjalta toimijoista, jotka liittyvät ohjelmistoon. Ohjelmiston kautta prosessit ja toimijat saatiin linkitettyä määritettäviin vaatimuksiin. Kartoitus toteutettiin vanhojen dokumentaatioiden pohjalta, haastatteluin sekä työn tekijän oman tietämyksen ja toimintojen tuntemuksen avulla. Vaatimuksiin liittyvät prosessit on lueteltu liitteen 4 kuvassa prosessit kehittämisessä.

Vaatimusmäärittelyyn kuuluva vaatimusten kartoitus valittiin tehtäväksi ohjelmistoa hyödyntävien prosessien selvityksellä. Näin varmistuttiin, että kuvattavat vaatimukset pohjautuvat liiketoiminnan, prosessien sekä käyttäjien todellisiin tarpeisiin. Ohjelmistoon tehdyt konfiguraatiot ja kustomoinnit olisi kyllä saatu dokumentoitua, mutta niiden linkitys vaatimuksiin nyt toteutetulla tavalla ei olisi onnistunut ilman prosessien selvitystä.

Työssä ei toteutettu vaatimusten analysointia sellaisena kuin se ala-luvussa 2.3 Vaatimusten käsittelyprosessin osa-alueena esiteltiin. Kaikki dokumentoidut vaatimukset oli jo toteutettu ja vaatimusten dokumentaation kannalta niiden prioriteetit olivat samat. Työssä ei ollut tarkoituksena analysoida tai parantaa vaatimuksia, eikä etsiä niihin uusia näkökulmia, vain dokumentoida vaatimusten nykytila. Tarkempi analyysi, vaatimusten arviointi, täydennys ja mahdollisesti laajennus sekä uusien näkökulmien huomiointi on jatkokehityksen aihe. Jatkokehittämisen aiheeksi jää myös vaatimusten analyysiin kuuluva vaatimusten käsittely ja jäsentäminen eri käyttäjäryhmien näkökulmasta.

Vaatimukset dokumentoitiin toimeksiantajan organisaatiossa käytettävään vaatimustaulukko-mallipohjaan. Mallipohja on luotu tukemaan koko vaatimusten käsittelyn prosessia. Mallipohjassa on eritelty liiketoiminnan vaatimukset, toiminnalliset ja ei-toiminnalliset vaatimukset, kaikki omiin välilehtiinsä. Mallipohjassa ohjataan kuvaamaan vaatimuksista mitä on tehtävä, ei sitä, miten ne tulee tehdä. Työn teoria-osuuden alaluvussa 3.1 Vaatimus todettiin tämän olevan hyvän vaatimuksen yksi ominaisuus. Mallipohja ohjaa aloittamaan vaatimusten määrittelyn liiketoiminta- tai asiakasvaatimuksista, joista johdetaan toiminnalliset vaatimukset. Jokainen toiminnallinen vaatimus, eli ohjelmistovaatimus, linkitetään liiketoimintavaatimukseen. Liiketoimintavaatimusten osalta jäljitettävyyden taataan, sillä mallipohjaan merkitään vaatimuksen lähde ja

päivämäärä. Vaatimusten jäljitettävyyden on näin turvattu, kuten alaluvussa 2.2 Asiakasvaatimuksista ohjelmistovaatimuksiin todettiin parhaaksi käytännöksi tehtävän.

Työssä vaatimusten dokumentointi aloitettiin toiminnallisten vaatimusten dokumentoinnista. Dokumentoinnissa käytettiin apuna ohjelmistotoimittajan esiselvitystä ohjelmiston tilasta. Esiselvityksessä oli lueteltu kaikki ohjelmistoon tehdyt konfiguraatiot ja kustomoinnit kuten prosessit, näkymät, taulut, indeksit, triggerit ja proseduurit. Lisäksi apuna käytettiin muuta ohjelmiston dokumentaatiota sekä itse ohjelmistoa. Myös ohjelmiston kehittämisestä vastaavan tiimin jäseniä haastateltiin vaatimusten dokumentaation ja varsinkin vaatimusten toteutustavan selvityksessä.

Dokumentoinnissa vaatimukset linkitettiin niitä hyödyntäviin prosesseihin ja käyttäjiin, jotka oli selvitetty kehittämisen prosessien kartoituksessa. Toiminnallisten vaatimusten dokumentoinnin jälkeen työssä johdettiin toiminnallisista vaatimuksista liiketoimintavaatimukset tai asiakasvaatimukset, kuten alaluvussa 2.2. Asiakasvaatimuksista ohjelmistovaatimuksiin niitä kutsuttiin. Liiketoimintavaatimuksissa pyrittiin kuvaamaan ohjelmiston käyttäjien tai prosessien konkreettisia tarpeita. Kuvauksessa pyrittiin välttämään sovellusalueen sanastoa sekä vastaamaan kysymykseen miksi. Liiketoimintavaatimus kuvattiin joukkona ohjelman toiminnallisuuksia, joilla tarve ratkaistaan, kuten alaluvussa 2.2. Asiakasvaatimuksista ohjelmistovaatimuksiin määritettiin. Liiketoimintavaatimusten dokumentointi edellytti tekijältä liiketoiminnallisten tavoitteiden ymmärtämistä ja miten nämä tavoitteet on pyritty toteuttamaan ohjelmiston toiminnallisuuden avulla.

Vaatimusten mallipohja ohjaa dokumentoimaan liiketoimintavaatimuksista tunnusteen, nimen, kuvauksen, käyttäjäryhmän, hyödyntävän prosessin tai toiminnan, lähteen ja päivämäärän, prioriteetin, tilan sekä mahdolliset kommentit. Toiminnallisista vaatimuksista dokumentoidaan edellisten lisäksi viittaus liiketoiminnalliseen vaatimukseen sekä palvelu tai käyttötapa missä vaatimus on huomioitu, toteutuserä, sovellus- / toteutusprojekti ja työn suuruusluokka. Mallipohjan avulla näistä perustiedoista saa raportointitietoja. Mallipohjasta vaatimukset voidaan viedä testauksen / vaatimustenhallinnan järjestelmään.

Toimeksiantajan organisaation vaatimusmäärittelyn mallipohja ei ole identtinen alaluovissa 2.2 Asiakasvaatimuksista ohjelmistovaatimuksiin esitellyiden kolmen vaatimusten luokan (toiminnalliset ja ei-toiminnalliset vaatimukset sekä reunaehdot ja rajoitteet) kanssa. Toimeksiantajan mallipohjassa arkkitehtuurin ja infrastruktuurin rajoitteet huomioidaan ei-toiminnallisissa vaatimuksissa. Ei-toiminnallisissa vaatimuksissa on kaikkia organisaation sovelluksia koskevia vakiovaatimuksia kuten suorituskyky. Vakiovaatimukset tarkennetaan mallipohjaan sovelluskohtaisesti. Toiminnallisten ja ei-toiminnallisten vaatimusten raja on näin selkeästi ohjattu. Toimeksiantajan vaatimusten luokittelutapa on rajoitteiden käsittelyn osalta vaatimukset ala luvussa 2.1 Vaatimus esitetyn Pohjosen näkemyksen mukainen. Ei-toiminnallisten vaatimuksien dokumentointi oli rajattu työn tuotoksen ulkopuolelle.

Työn tuotoksena luodut vaatimukset validoitiin ennen työn hyväksyntää. Validointi toteutettiin katselmoimalla vaatimukset sovellusta ylläpitävän työryhmän kanssa. Toimeksiantajan kanssa validointia tehtiin useaan otteeseen työn edetessä. Toimeksiantajan kanssa käytiin läpi tuotettavien dokumenttien muotoa ja tavoitetta sekä lopuksi valmiita vaatimuksia.

Liiketoimintavaatimuksia dokumentoitiin 15 kappaletta ja toiminnallisia vaatimuksia 85 kappaletta. Vaatimuksista koostuva vaatimusdokumentaatio on työn liitteenä (liite 6). Työn aikana ei löytynyt yhtään konfiguraatiota tai kustomointia, joka olisi rakennettu ilman, että taustalla olisi liiketoiminnan vaatimusta. Toki työn aikana huomattiin toiminnallisuuksia ja toteutuksia, joita ei enää hyödynnetä ja on näin otettu pois käytöstä. Ohjelmiston ylläpidossa ja muutoshallinnassa on huolehdittava vaatimustenhallinnasta, sillä ohjelmistoon ei pidä rakentaa yhtään toiminnallisuutta, johon ei ole vaatimusta taustalla. Vaatimustenhallinnan aktiviteettien sisällyttäminen ohjelmiston ylläpitoon on yksi työn jatkokehittämisen toimenpide-ehdotuksista.

Nykytilan kuvaukseen ryhmiteltiin yhteensä 14 toiminnallisen vaatimuksen ryhmää. Nykytilan kuvaus on työn liitteenä dokumentoitujen vaatimusten yhteenveto (liite 6). Nykytilan kuvauksessa vaatimuksista on dokumentoitu otsikko, kuvaus, toiminnallisen vaatimuksen tunnus sekä vaatimuksen toteutustapa. Vaatimukset ryhmiteltiin nykytilan kuvaukseen niiden tukemien prosessien sekä toteutustavan mukaan.

### 3.4 Takaisinmallinnus vaatimusten dokumentoinnissa

Vaatimukset kuvattiin ohjelmiston nykytilan mukaisesti ja ohjelmiston nykytilaa kuvaten. Vaatimusten dokumentoinnissa käytettiin apuna ohjelmien takaisinmallinnusta. Työn tietopohjan alaluvussa 2.6 Ohjelmien takaisinmallinnus esiteltiin ohjelmiston tutkimisena, jonka tavoitteena on tunnistaa ohjelman osat, ohjelmien osien suhteet sekä luoda kuvaus ohjelmasta. Takaisinmallinnusta hyödynnetään yleisesti tämän työn tapaan dokumentoinnin paikkailuissa tai tekemisessä jälkikäteen.

Työssä takaisinmallinnus toteutettiin ohjelmiston toiminnallisuuksia ja koodia tutkivalta ja analysoivalta. Takaisinmallinnuksessa hyödynnettiin myös dokumentaatiota sekä ohjelmiston kehittäjien haastatteluilla. Näiden aktiviteettien tuloksena saadut vaatimukset dokumentoitiin. Ohjelmistotuotannossa kehittämisessä edetään vaatimuksista suunnittelun kautta lähdekoodin luontiin ja ohjelmiston toteutukseen. Takaisinmallinnuksessa edetään ohjelmiston käyttäytymisestä ja lähdekoodin kautta suunnittelun ymmärtämiseen sekä vaatimusten selvittämiseen.

Alaluvun 2.6 Ohjelmien takaisinmallinnus mukaan takaisinmallinnus nähdään ohjelmien ylläpitoa, uudistamista ja parantamista edeltävä, ohjelmaa analysoivana vaiheena. Työn takaisinmallinnuksessa syntynyt kuvaus mahdollistaa ohjelman hyödyntämisen jatkoanalyysin. Takaisinmallinnuksessa dokumentaatioon valittiin työlle ja sen kohde-ryhmälle sopiva tarkkuustaso. Vaatimukset dokumentoitiin vaatimusmäärittelyn ohjeistuksen hyvien käytäntöjen mukaisesti. Tavoitteena oli tuottaa ymmärrettävää materiaalia ohjelmistosta ja sen tilasta jota voidaan hyödyntää jatkossa päätöksenteossa ja kehittämisessä. Takaisinmallinnukseen voi kuulua luvun 2.6 mukaan analyysia ratkaisun hyödyistä, kustannuksista sekä seurauksista tai arvioi miten ratkaisu sopii olemassa olevaan järjestelmään mutta nämä oli rajattu tämän työn aihealueen ulkopuolelle. Jatkoanalyysi kuuluu työn jatkokehityksen aihe-ehdotuksiin.

Takaisinmallinnuksen avulla luodun vaatimusmäärittelyn dokumentaatioon kuvatut vaatimukset luokiteltiin sen mukaan minkä tyyppisiä muutoksia ne aiheuttivat ohjelmaan. Luokittelussa käytettiin apuna alaluvussa 3.1 Valmisohjelmiston hyödyntäminen esiteltyä ohjelmiston konfiguraation ja kustomoinnin käsitteitä. Näiden ohjelmistotoi-



mittajan tarkasti määrittämien käsitteiden avulla pystytään luomaan pohja keskustelulle ja ohjelmiston kehittämissuunnitelmalle. Toteutettavan kehittämissuunnitelman onnistumisen kannalta on merkittävää, että kaikki ohjelmiston kanssa toimivat tahot, prosessien omistajat, ohjelmiston kehittämisestä vastaavat sekä kehitystyöstä vastaavat ymmärtävät samalla tavoin kehitystyössä tehtyjen ratkaisuiden merkityksen kokonaisuuden kannalta.

## 4 Yhteenveto

Työn tuloksena kuvattiin projekti- ja portfoliohallinnan ohjelmiston nykytila. Kuvaus toteutettiin dokumentoimalla liiketoiminnan ja toiminnalliset vaatimukset. Näin jälkikäteen arvioiden nykytilan kuvauksessa onnistuttiin yli odotusten. Kuvauksen avulla pystyttiin erottamaan projektin ja portfoliohallinnan ohjelmiston omat toiminnallisuudet toimeksiantajan organisaation toimesta tehdyistä konfiguraatioista ja kustomoinneista. Yhteenvedon avulla osoitettiin toimeksiantajan organisaation prosessien piirteet, jotka vaativat konfiguraatioita tai kustomointeja valmisohjelmiston perustoiminnallisuuksien laajentamiseksi. Lisäksi kustomoinnit pystyttiin yhdistämään ohjelmiston tukemiin prosesseihin ja käyttäjiin sekä ohjelmiston kehittämisen ja ylläpidon kustannuksiin. Yhteenvedon avulla organisaatio pystyy laatimaan ohjelmiston kehittämissuunnitelman ja kohdistamaan kehittämisen halumaansa suuntaan.

Työn aiheen rajaus oli työn toteutuksen kannalta onnistunut. Rajaus loi työn toteutukselle selkeät reunaehdot ja tavoitteet, niin tietopohjan kuin vaatimusdokumentaation ja vaatimusten yhteenvedon toteutuksen osalta. Aihe antoi mahdollisuuden keskittyä tietopohjassa perinpohjaisesti vaatimustenhallinnan osa-alueisiin. Vaatimustenhallinta saatiin yhdistettyä työssä ohjelmistokehittämiseen ja ylläpitoon, eikä se näin jäänyt yksinäiseksi tietopohjan osa-alueeksi, ilman käytännön yhteyttä. Työn tekemisen ja tekijän näkökulmasta tietoperusta ja työn toteutus osuus saatiin yhtenäiseksi, toisiaan tukevaksi kokonaisuudeksi.

Työn toteutuksessa oli haastavaa saada toiminnalliset vaatimukset dokumentoitua oikealla tarkkuustasolla. Jälkikäteen kuvatuissa toiminnallisissa vaatimuksissa ei kaikilta osin pystytty riittävän ylätasolla olevaan kuvauksen tarkkuustasoon. Varsinkin raportoinnissa hyödynnettävien luokittelutekijöiden osalta kuvauksessa mentiin lähes spesifikaatioiden luonnin puolelle. Myös toteutustavan huomioimatta jättäminen oli haastavaa dokumentoinnissa. Toimeksiantaja ja tekijä totesivat tämän olevan hyväksyttävää, sillä luokittelutekijöiden osalta vaatimusten dokumentoinnin tarkkuustaso ei vaikuttanut niiden ymmärrettävyyteen. Vaatimusten dokumentoinnissa yksi tärkeä tavoite on vaatimusten ymmärrettävyys. Haasteet antoivat mukavaa lisäharjoitusta vaatimusten kir-

joittamisessa ja oikean kirjoitusmuodon löytämisessä. Tämä kehitti tekijän taitoa kirjoittaa vaatimukset oikeassa muodossa.

Syksyllä 2012 opinnäytetyön prosessin etenemisen haasteena oli puuttuva ohjelmiston nykytilankuvaus, joka tarvittiin kehittämissuunnitelman luonnissa. Nykytilan kuvauksen puute johtui suoraan ohjelmiston kehittämistiimin työtavoista ja vallitsevasta työku-  
tuurista. Opinnäytetyön prosessi eteni vasta, kun myös tekijä oli valmis haastamaan omat työtapansa kehittämistiimin jäsenenä, tiimissä vallitsevan työku-  
tuurin sekä kat-  
somaan tilannetta ulkopuolisen silmin. Tämän jälkeen prosessissa päästiin tasolle, jossa pystyttiin etsimään uusia ratkaisuja ja lähestymistapoja puuttuvaan nykytilan kuvauk-  
seen. Lopulta työhön löydettiin sopiva toteutustapa eli vaatimusten kuvaus. Vaatimus-  
ten kuvaus tuki lopulta molempien osapuolten tavoitteita työn suhteen. Työn tekijä pääsi kehittämään osaamistaan vaatimustenhallinnan osa-alueella ja toimeksiantaja sai kehittämissuunnitelmaan tarvittavan taustamateriaalin.

Opinnäytetyön prosessin aikana tekijän valmiudet etsiä lähdemateriaalia ja taidot hyö-  
dyntää tietoa eri lähteistä kehittyivät. Työssä vaadittiin tietopohjan oppien soveltamista  
käytäntöön ja tämän kautta tekijän osaaminen niin vaatimusten kartoittamisessa, do-  
kumentoinnissa sekä validoinnissa kehittyivät. Myös vaatimusten analysoinnin taito  
kehittyi takaisinmallinnuksen työskentelytavassa ja vaatimusten toteutustavan ana-  
lysoinnissa, vaikka vaatimusten analysointia ei sinänsä toteutettu työssä.

Työssä ei pelkästään dokumentoitu ohjelmiston nykytilaa kuvaavia liiketoiminnan ja  
toiminnallisia vaatimuksia vaan mahdollistettiin ohjelmiston kehittämistiimin työntapo-  
jen ja dokumentointikäytäntöjen muutos. Työtapojen ja dokumenttikäytäntöjen muutos  
ei kuulunut työn tavoitteisiin. Tästä huolimatta voidaan arvioida, että työn tekijän kyvyt  
oman ja työyhteisön toiminnan parantamiseen paranivat opinnäytetyön prosessin aika-  
na. Samoin tekijä harjaantui ongelmakohtien havaitsemisessa sekä sopivien ratkaisujen  
etsimisessä.

Toimeksiantajalta saadun palautteen mukaan työssä toteutettu taustamateriaali ja tuo-  
tokset ovat olleet hyödyllisiä ja tarpeellisia. Toimeksiantaja on ottanut tuotokset alka-  
neen ohjelmiston jatkokehittämisen materiaaliksi. Toimeksiantaja on myös huolehtinut,

että luotu vaatimuskäsitelmä on otettu ohjelmiston kehittämisessä osaksi sovel-lusdokumentaatiota ja sitä ylläpidetään muun dokumentaation osana.

#### 4.1 Johtopäätökset

Vaatimusten käsittelyn prosessien aktiviteeteilla on suuri merkitys eri ohjelmistokehit-tämisen ja ylläpidon vaiheiden onnistumisessa. Vaatimusmäärittely on tapa luoda eri osapuolten välille yhteinen ymmärrys ohjelmistoa koskevista tarpeista ja odotuksista. Niin ohjelmistokehittämisessä kuin ohjelmiston ylläpidonkin vaiheessa pyritään ohjel-miston avulla ratkaisemaan olemassa olevaa ongelmaa tai tukemaan jotain toimintaa. Mikäli ongelman syvintä olemusta tai ymmärrystä tuettavasta toiminnasta ei saada selvi-tettyä vaatimusmäärittelyssä, ei määrittelyn pohjalta toteutettu ohjelmisto voi näitä vaa-timuksia tai toiveita täyttää.

Monet ohjelmistojen ongelmat sekä käyttäjien tyytymättömyys ohjelmistoja kohtaan on peräisin huonoista vaatimusmäärittelyistä. Ohjelmisto, joka sisältää ongelmia tai on käy-tettävyydeltään hankala, ei tuota siltä toivottua lisäarvoa organisaatiolle. Ohjelmisto saattaa jopa aiheuttaa enemmän kustannuksia, esimerkiksi kuluneena työaikana, kuin hyötyä. Ohjelmistojen ylläpidon kustannukset voivat nousta jopa 50 % ohjelmistokehit-tämisen aikaisissa vaatimuksissa tehdyistä virheistä johtuen. Vaatimusten virheet johtu-vat huonoista tai virheellisistä vaatimusmäärittelyistä. Näistä syistä johtuvat virheet ovat estettävissä, mikäli kehittämisessä suhtaudutaan vaatimusmäärittelyyn tarvittavalla va-kavuudella ja vaatimusmäärittelyyn kuuluvat päätökset tehdään ennen vaatimusten to-teuttamista.

Nykyiset uudet kommunikaation muodot vaikuttavat myös vaatimusten käsittelyyn. Vaatimusten hallinnassa on pystyttävä huomioimaan uusista kommunikaatiokanavista tulevat vaatimukset ja toiveet. Nämä muuttuneet kommunikaation muodot luovat haas-teen varsinkin vaatimusten analysointiin ja niiden arvon määrittelyyn. Analyysin ja ar-von määrittelyksen toimintatapoja on muutettava, sillä onnistuminen voidaan saavuttaa vain, jos ymmärretään tarpeet vaatimuksen takana. Uusien kommunikaatiokanavien kautta kerättyjen vaatimusten taustalla olevat tarpeet on opittava löytämään uudella tavalla.

Ohjelmiston nykytilan arvioinnissa ohjelmistoa on tarkasteltava kokonaisvaltaisesti eri näkökulmista. Prosessien näkökulmasta voidaan sanoa, että eri prosessit hyödyntävät ohjelmistosta eri tietoja. Ohjelmiston näkökulmasta taas kaikki useat eri prosessit hyödyntävät samoja tietoja, katsontakulma ja tietojen hyödyntäjät poikkeavat. Ohjelmistoa ei voida kehittää pitkäjänteisesti, ilman että vaikutukset kaikkiin prosesseihin huomioidaan ja muutoksista keskustellaan kunkin prosessin edustajan kanssa. Ohjelmiston hyödyntämisen tarkempi analyysi on työn jatkokehitysehdotuksissa. Lisäksi työssä suositellaan analyysia ohjelmistoratkaisun hyödyistä sekä sen kustannuksista tai seurauksista. Näin voidaan arvioida vastaako ohjelmiston tuki sille asetettuja tarpeita.

## 4.2 Jatkokehitysehdotukset

Nykytilan arvioinnissa dokumentoitujen vaatimusten kriittinen arviointi, niiden tarpeellisuuden määrittäminen ja priorisointi ovat kehittämissuunnitelmassa tulevia toimenpiteitä. Ohjelmistoon toteutetuista vaatimuksista ja ohjelmiston toiminnallisuuksista on hyvä karsia pois ne toiminnallisuudet ja vaatimukset, jotka eivät tue organisaation nykyisiä tarpeita tai tahtotilaa. Ohjelmiston kehittämistyön kannalta on tarpeen määrittää, missä määrin valmisohjelmistoa halutaan konfiguroitavan ja kustomoitavan, jotta se tukisi prosesseja halutulla tavalla. Samalla on tarkasteltava kriittisesti, kuinka paljon ohjelmistoa halutaan muutettavan, jotta se tukisi prosesseja paremmin, ja mikä verran organisaatio on valmis joustamaan prosesseissaan ohjelmiston toiminnallisuuksien suuntaan. Näin vaatimusten toteutukseen saadaan onnistuneeseen ja kestäväan lopputulokseen vievät reunaehdot ja rajoitteet. Samalla ohjelmiston ylläpitävään kehittämiseen saadaan toivottu kehityssuunnitelman pohja. Tämän jälkeen voidaan lähteä suunnittelemaan kehittämisen pitkäaikaisempia linjauksia.

Vaatimustenhallinta ja vaatimusdokumentaation käyttöönotto oli rajattu tämän työn ulkopuolelle. Jatkokehittämisessä olisikin ensiarvoisen tärkeää, että vaatimustenhallinnan aktiviteetit ja vaatimusmuutosten hallinta otetaan mukaan ohjelmiston ylläpidon työtapoihin. Näin myös työssä tuotettu vaatimusdokumentaatio pysyy päivitettyinä ja ajantasaisena. Työssä toteutettu vaatimusmäärittely tehtiin valmiiksi, mutta se on vain aikansa kuva. Vaatimusmäärittely vanhenee heti kun tulee ensimmäinen uusi vaatimus, jota ei dokumentoida. Työn valmistumisen jälkeen vaatimusdokumentaation ylläpito ja

päivitys voi alkaa vaatimustenhallinnan aktiviteettien mukaisesti. Jatkon ylläpidon onnistumisen kannalta on hyvä kehittää vaatimusmuutosten hallinnan prosessi ja varsinkin uusien vaatimusten vastaanottaminen ja analysointi.

Lisäksi on hyvä sopia kehittämistyössä yhteisesti käytettävä tapa muutosten hallintaan sekä kuvata prosessi, jonka mukaisesti toiveet hyväksytään toteutettaviksi muutoksiksi. Myös ohjelmistoa ja sen tukemia toimia koskevat palautteet on hyvä yhdistää vaatimustenhallintaan.

Työtapojen ja dokumentointitapojen muutos ei kuulunut työn alkuperäiseen toimeksiantoon, vaan tekijä huomasi mahdollisuuden työn edetessä. Dokumentointitapojen muutoksella mahdollistetaan ohjelmiston omistaville tahoille parempi ymmärrys ohjelmistoon tehtävistä muutoksista ja tulevaisuudessa tehtävien muutosten vaikutuksista. Näin muutoshallinnan yhteydessä voidaan aidosti arvioida muutoksen vaikutukset koko ohjelmiston ja käyttäjien kannalta.

Työssä ei dokumentoitu ei-toiminnallisia vaatimuksia. Vaatimuskumentaation jatkokehittämisenä ei-toiminnallisten vaatimusten keräys ja dokumentointi on seuraava askel. Vaatimusten dokumentointitapojen muutoksella mahdollistetaan ohjelmiston omistaville tahoille parempi ymmärrys ohjelmistoon tehtävistä muutoksista ja ohjelmistoon tulevaisuudessa tehtävien muutosten vaikutuksista.

## Lähteet

Beynon-Davis, P. 2009. Business information systems. Palgrave Macmillan. Hampshire.

Grant, T. 2012. Software Requirements are where we define value. Luettavissa: [http://blogs.forrester.com/tom\\_grant/12-01-04-software\\_requirements\\_are\\_where\\_we\\_define\\_value](http://blogs.forrester.com/tom_grant/12-01-04-software_requirements_are_where_we_define_value). Luettu: 24.1.2013

Haikala, I., Mikkonen, T. 2011. Ohjelmistotuotannon käytännöt. Talentum. Helsinki.

Haikala, I., Merijärvi, J. 2003. Ohjelmistotuotanto. Talentum. Helsinki.

Harsu, M. 2003. Ohjelmien ylläpito ja uudistaminen. Talentum. Helsinki.

Harsu M. 2012. Takaisinmallinnus Luettavissa: <http://www.cs.tut.fi/~evo/kalvot/takmalli2.pdf>. Luettu: 24.1.2013.

Hamlet, D., Maybee, J. 2001. The Engineering of Software. Technical Foundations for the Individual. Addison Wesley Longman, Inc. Boston.

Higgins, T. 2011. New Forrester Report on a Requirement Revolution. Luettavissa: [http://www.blueprintsys.com/blog/new\\_forrester\\_report\\_on\\_a\\_requirements\\_revolution/](http://www.blueprintsys.com/blog/new_forrester_report_on_a_requirements_revolution/). Luettu 24.1.2013.

ITIL. ITIL-sanasto ja lyhenteet. Suomenkielinen. ITIL® Suomenkielinen sanasto, v1.0, 29 heinäkuuta 2011. Luettavissa: <http://www.itil-officialsite.com/nmsruntime/saveasdialog.aspx?IID=1214>. Luettu: 10.3.2013.

Johdatus ohjelmistotuotantoon. Luettavissa: [cs.joensuu.fi/~jimmonen/jot\\_moniste/jot\\_moniste\\_121.html](http://cs.joensuu.fi/~jimmonen/jot_moniste/jot_moniste_121.html). Luettu: 10.3.2013.

Kokonaisarkkitehtuuri sanasto. Kansallinen digitaalinen kirjasto. Luettavissa:  
<http://www.kdk2011.fi/fi/kokonaisarkkitehtuuri/sanasto>. Luettu 10.3.2013.

Kotonya, G., Sommerville, I. 2002. Requirements engineering: processes and techniques. John Wiley & Sons. Chichester.

Kupila, P. 2013. Tietotekniikka synnyttää hermoraunioita. Sähköiset raportointityökalut ovat käteviä – hallinnolle. Asiantuntijatyötä niillä ei helpoteta. Talouselämä 7/2013.

Ohjelmistotoimittaja 2013. Upgrade Assessment. Espoo.

Paakki, Juha. 2011. Ohjelmiston vaatimusmäärittely. Luettavissa:  
<http://www.cs.helsinki.fi/u/paakki/Vaatimus-11-Luentokalvot-1.pdf>. Luettu: 23.1.2013.

Pohl, K., Rupp, C. 2011. Requirements Engineering Fundamentals. A Study Guide for the Certified Professional for Requirements Engineering Exam. Foundation Level IREP compliant. Rocky Nook Inc. Santa Barbara.

Robertson, S., Robertson, J. 2011. Mastering the Requirements Process. Addison-Wesley. Upper Saddle River (NJ).

Schwartz, M. 2001. Reverse-Engineering. Luettavissa:  
[http://www.computerworld.com/s/article/65532/Reverse\\_Engineering](http://www.computerworld.com/s/article/65532/Reverse_Engineering). Luettu: 28.1.2013.

Tietojärjestelmän suunnittelu. Luettavissa:  
[www2.kyamk.fi/~zjaan/kurssit/tjs/Osa2.ppt](http://www2.kyamk.fi/~zjaan/kurssit/tjs/Osa2.ppt). Luettu 10.3.2013.

Vuori, M. 2009. Vaatimusmäärittelyn huonoimmat käytännöt. Systemityö 9/2009. Luettavissa: <http://www.pcuf.fi/sytyke/lehti/kirj/st20092/ST092-20A.pdf>. Luettu: 15.1.2013.



Wieggers, K.E. 2000. When Telepathy Won't Do: Requirements Engineering Key Practices. Process Impact Cutter IT Journal, vol. 13, no. 5 (May 2000) Luettavissa: <http://www.processimpact.com/articles/telepathy.html>. Luettu: 23.1.2013.

Wigser, S. 2012. Maximizing Requirements Value Throughout the Product Lifecycle. Luettavissa: <http://blogs.seapine.com/2012/04/maximizing-requirements-value-throughout-the-product-lifecycle/>. Luettu: 23.1.2013.

# Liitteet

## Liite 1. Keskeiset käsitteet

**Asiakasvaatimus** (customer requirements) on suoraan asiakkaan tarpeista lähtenyt vaatimus, jonka kuvauksessa käytetään harvoin sovellusalueen sanastoa. Asiakasvaatimuksen sijaan voidaan käyttää termiä ominaisuus. (Haikala & Mikkonen 2011, 61.)

**Konfiguraatio** (configuration) ovat usein toteutettu ohjelmiston käyttöliittymän kautta, ohjelmiston olemassa olevia toiminnallisuuksia ja ominaisuuksia hyödyntäen (Ohjelmistotoimittaja 2013).

**Kustomointi** (customization) ovat sellaiset lisäykset ohjelmiston peruskoodiin, jota ei ole suunniteltu ohjelmiston perustoiminnallisuuksiin tai se ei ole ohjelmistotoimittajan julkaisema ohjelmiston osa, kuten tietokantaan lisätyt triggerit, taulut, indeksit, proseduurit, funktiot ja näkymät (Ohjelmistotoimittaja 2013).

**Ohjelmien evoluutio** (software evolution) tarkoittaa ohjelman kasvua ja kehittymistä eli toiminnallisuuksien lisäämistä ja parantamista (Harsu 2003, 20 - 21).

**Ohjelmien takaisinmallinnus** (software reverse engineering) Takaisinmallinnuksessa tutkitaan olemassa olevaa ohjelmaa ja jäljitetään ohjelman suunnitteluratkaisu tai uudelleendokumentoidaan ohjelma (Harsu 2003, 117).

**Ohjelmisto** (software) on looginen tietokoneohjelmista, niiden käyttämistä tiedostoista ja niiden dokumentaatiosta koostuva kokonaisuus (Johdatus ohjelmistotuotantoon).

**Ohjelmistovaatimus** (software requirements) on ohjelman toiminto, joka avulla asiakasvaatimukset toteutetaan ja annetaan käyttäjälle käytettäväksi (Haikala & Mikkonen 2011, 61).

**Ominaisuus** (feature) tai asiakasvaatimus on joukko toiminnallisuuksia joiden avulla ratkaistaan asiakkaan ongelma (Haikala & Mikkonen 2011, 62).

**Räätälöity ohjelmisto** (bespoke software, custom software) on kokonaan itse tehtynä tai jollakin toisella kolmannella osapuolella teetetty ohjelmisto, joka vastaa tilaajaorganisaation vaatimuksia (Beynon-Davis 2009 349).

**Sovellus** (application) on ohjelmisto, joka tarjoaa käyttäjälle tarkoitettua palvelua (ITIL).

**Sovellusohjelma** (application software) kts. sovellus.

**Takaisinmallinnus** kts. Ohjelmien takaisinmallinnus.

**Tekninen vaatimus** (technical requirements) Ohjelmiston toteutuksessa ohjelmistovaatimukset jakautuvat pienemmiksi teknisiksi vaatimuksiksi (Haikala & Mikkonen 2011, 63).

**Tietojärjestelmä** (information system) on kokonaisuus, joka koostuu ohjelmistosta, tiedonsiirtolaitteista, tietojenkäsittelylaitteista ja järjestelmää käyttävistä ihmisistä sekä käyttöön tehdyistä ohjeista (Tietojärjestelmän suunnittelu).

**Tietokoneohjelmisto** (computer software) kts. ohjelmisto.

**Toiminnallinen määrittely** (functional specification) on vaatimusmäärittelyssä tuotettava dokumentti, joka sisältää asiakas- ja ohjelmistovaatimukset. (Haikala & Mikkonen 2011, 68).

**Vaatimus** (requirement) kuvaa asioita, joita tuotteella voi tehdä sekä sen laadullisia ominaisuuksia. Vaatimukset jaotellaan toiminnallisiin vaatimuksiin, ei-toiminnallisiin vaatimuksiin sekä reunaehtoihin tai rajoitteisiin. (Haikala & Mikkonen 2011, 61).

**Vaatimusten jäljitettävyys** (requirements traceability) kuvaa kykyä seurata vaatimuksen elinkaarta sen määrittelystä toteutukseen asti (Haikala & Mikkonen 2011, 63).

**Vaatimusmäärittely** (requirements definition) sisältää vaatimusten käsittelyprosessin ensimmäiset vaiheet, joiden aikana kartoitetaan sidosryhmien asiakastarpeet, analysoidaan, dokumentoidaan ja validoidaan vaatimukset (Haikala & Mikkonen 2011, 66).

**Vaatimustenhallinta** (requirements management) sisältää vaatimusten käsittelyprosessin jälkimmäiset vaiheet ja se sisältää vaatimusten muutosten dokumentointi, riippuvuuksien seuranta sekä raportointi (Haikala & Mikkonen 2011, 66).

**Vaatimusten käsittely** (requirements engineering) prosessi sisältää vaatimusmäärittelyn ja vaatimusten hallinnan (Haikala & Mikkonen 2011, 66).

**Valmisohjelmisto** (software package, off-the-shelf software (OTS)) on ohjelmistotoimittajalta hankittu ohjelmisto, jonka ominaisuudet ja toiminnallisuudet on kehitetty tukemaan liiketoiminta-alueen aktiviteetteja ja tehtäviä (Beynon-Davis 2009 349).

Liite 2. Projektisuunnitelma

ID	Task Name	Start	Finish	Duration	% Complete	touko 2012				kesä 2012				heinä 2012				elo 2012				syys 2012				loka 2012				marras 2012			
						29.4	6.5	13.5	20.5	27.5	3.6	10.6	17.6	24.6	1.7	8.7	15.7	22.7	29.7	5.8	12.8	19.8	26.8	2.9	9.9	16.9	23.9	30.9	7.10	14.10	21.10	28.10	4.11
1	Opinnäytetyön suunnittelu	28.5.2012	3.12.2012	190d	100%																												
2	Suunnitelman laatiminen alkaa	28.5.2012	28.5.2012	0d	100%																												
3	Tavoitteen ja lopputuloksen määrittäminen	1.6.2012	24.6.2012	24d	100%																												
4	Aikataulutuksen suunnittelu	1.6.2012	24.6.2012	24d	100%																												
5	Opinnäytetyösopimus allekirjoitettu ja suunnitelma hyväksytty työpaikalla	4.7.2012	4.7.2012	0d	100%																												
6	Teorian ja tietoperustan valinta	1.6.2012	11.7.2012	41d	100%																												
7	Tiivistelmän ja johdannon ensimmäisen version kirjoitus	23.7.2012	27.7.2012	5d	100%																												
8	Opinnäytetyösuunnitelma valmis	27.7.2012	27.7.2012	0d	100%																												
9	Uudelleensuunnittelu alkaa	2.11.2012	2.11.2012	0d	100%																												
10	Uudelleensuunnittelu	2.11.2012	25.11.2012	24d	100%																												
11	Johdannon ja tiivistelmän kirjoitus uuden aiheen mukaiseksi	23.11.2012	3.12.2012	11d	100%																												
12	Toimeksiantaja hyväksyi uuden opinnäytetyösuunnitelman	26.11.2012	26.11.2012	1d	100%																												
13	Ohjaaja hyväksyi uuden opinnäytetyösuunnitelman	3.12.2012	3.12.2012	1d	100%																												
14	Työn uusi suunnitelma valmis ja hyväksytty	3.12.2012	3.12.2012	0d	0%																												

ID	Task Name	Start	Finish	Duration	% Complete	joulu 2012				tammi 2013				helmi 2013				maalisk 2013				huhti 2013							
						2.12	9.12	16.12	23.12	30.12	6.1	13.1	20.1	27.1	3.2	10.2	17.2	24.2	3.3	10.3	17.3	24.3	31.3	7.4	14.4	21.4	28.4		
14	Työn uusi suunnitelma valmis ja hyväksytty	3.12.2012	3.12.2012	0d	0%	◆																							
15	<b>Teoria ja tietoperusta</b>	<b>5.12.2012</b>	<b>3.5.2013</b>	<b>149d</b>	<b>100%</b>	▶																							
16	Teoriaan ja tietoperustaan tutustuminen ja kirjoitus alkaa	5.12.2012	5.12.2012	0d	100%	◆																							
17	Lähteisiin tutustuminen	5.12.2012	17.12.2012	13d	100%	■																							
18	Valmiin tietoperustan karsiminen ja muokkaus uuteen aiheeseen sopivaksi	2.1.2013	21.2.2013	51d	100%	■																							
19	Tietoperustan kirjoitus	2.1.2013	22.1.2013	21d	100%	■																							
20	Tietoperustan ensimmäinen versio valmis	30.1.2013	30.1.2013	0d	100%	◆																							
21	Tietoperustan muokkaus ja viimeistely	1.2.2013	7.4.2013	66d	100%	■																							
22	Tietoperusta valmis	3.5.2013	3.5.2013	0d	100%	◆																							

ID	Task Name	Start	Finish	Duration	% Complete	joulu 2012				tammi 2013				helmi 2013				maalis 2013				huhti 2013	
						2.12	9.12	16.12	23.12	30.12	6.1	13.1	20.1	27.1	3.2	10.2	17.2	24.2	3.3	10.3	17.3	24.3	31.3
23	Työn toteutus	1.2.2013	11.4.2013	69d	100%																		
24	Työn lopullinen suunnittelu alkaa	1.2.2013	1.2.2013	0d	100%																		
25	Toteutustavan valinta	1.2.2013	10.2.2013	10d	100%																		
26	Toteutuksen rajausten määrittäminen	7.2.2013	12.2.2013	6d	100%																		
27	Toteutus alkaa	15.2.2013	15.2.2013	0d	100%																		
28	Ohjelmistoa hyödyntävien prosessien kartoitus	15.2.2013	21.2.2013	7d	100%																		
29	Toiminnallisten vaatimusten dokumentointi	25.2.2013	7.3.2013	11d	100%																		
30	Liiketoiminnan vaatimusten dokumentointi	11.3.2013	25.3.2013	15d	100%																		
31	Vaatimusten yhteenvedon kokoaminen	1.4.2013	5.4.2013	5d	100%																		
32	Tuotosten katselmoinnit	15.3.2013	2.4.2013	19d	100%																		
33	Tuotosten viimeistely	1.4.2013	8.4.2013	8d	100%																		
34	Toimeksiantaja hyväksynyt tuotokset	11.4.2013	11.4.2013	0d	100%																		

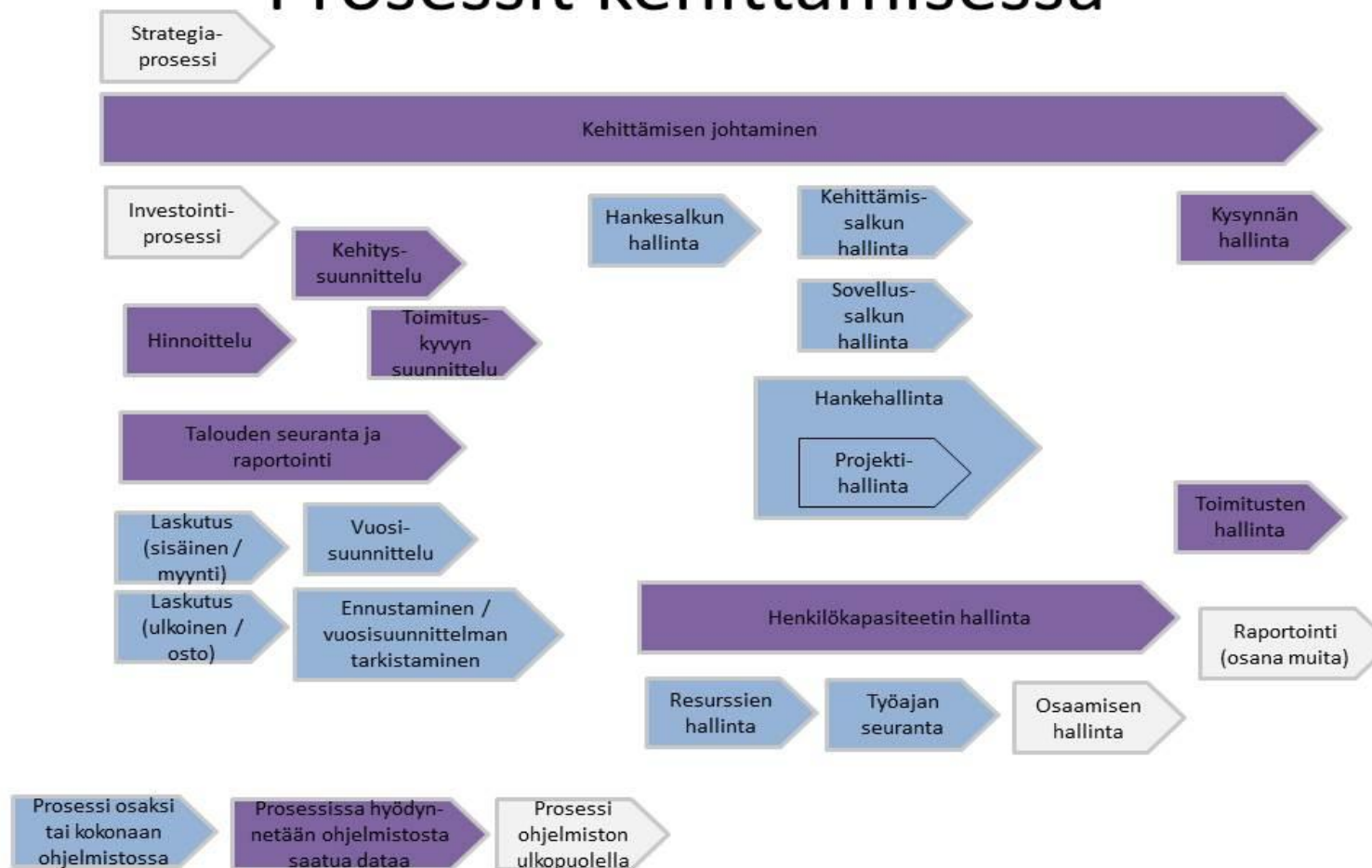




Liite 3. Ohjelmistotoimittajan näkemys ohjelmiston konfiguraatioista ja kustomoinneista

Muutos	Muutoksen selite	Muutoksen luonne
Laskutus	<p>Laskutus sisältää SAP -järjestelmään välitettävät kustannus- ja veloitustiedot. Projekteilla objektille voidaan määrittää aikaan sidonnaisesti yksi tai useampi maksaja ja muut laskutuksessa käytettävät tiedot. Projektin tehtävällä on mahdollista määrittää tehtävän maksaja. Resurssien osalta objektilla on laskutuksessa käytettävät tiedot. Objektien tietojen historoinnilla on mahdollistettu takautuvien korjausten teko.</p> <p>Kahta kustannus matriisiä hyödynnetään järjestelmään rakennetussa laskutusprosessissa erottamaan järjestelmän sisäinen kustannus ja laskutus kustannus. Toinen on projektien kustannusmatriisi, jota käytetään OOB -tapaan. Toinen on hallintolisämatriisi, jossa olevat hinnat lisätään projektien kustannusmatriisin hintoihin.</p> <p>Useita proseduureja herätetään järjestelmän taustajojen kautta, jotta tuntikorttien sisältämä data saadaan prosessoitua luotuihin kustannustauluihin laskutuksen sääntöjen mukaisesti matriiseja hyödyntäen, eri aikajaksojen laskutussääntöjen mukaan.</p>	kustomointia, konfigurointia
Hankkeiden budjetti	<p>Budjetointi tehdään hanketasolle ja valutetaan projekteille. Jotta hankkeiden budjetit eivät tuplaannu, on järjestelmään rakennettu negaatioprojekti toiminnallisuus. Negaatioprojekti kuvaa negatiivista hankkeen projektien budjettien summaa. Toiminnallisuus on luotu triggereiden ja ajastettujen pollaus prosessien avulla, jotka luovat ja päivittävät negaatioprojekteja.</p>	kustomointia, konfigurointia
Muut kuin henkilötöiden kustannukset	<p>Muut kuin henkilötöiden toteutuneet kustannukset tuodaan tiedossa SAP – järjestelmästä ohjelmistoon. Kustannuksia voi myös syöttää manuaalisesti. Tämä on toteutettu ohjelmiston objektien konfiguroinnilla, sekä prosesseilla jotka käsittelevät ja validoivat datan ja siirtävät sen tietokannan tauluihin.</p>	kustomointia, konfigurointia
Resursointikokonaisuudet	<p>Hankkeiden resurssien allokointi tehdään resursointikokonaisuuksiin. Triggereiden ja ajastettujen, pollaavien prosessien avulla seurataan allokointeja sekä resursointikokonaisuuksien resurssien kiinnitysoikeuksia. Vain resursointikokonaisuuteen luvattujen resurssien voi kiinnittää resursointikokonaisuuteen kuuluvien projekteihin.</p>	kustomointia, konfigurointia
Kustannusarvot	<p>Objektien konfiguroinnilla sekä prosesseilla on mahdollistettu käyttäjille kustannusarvion luonti automatisoidusti halutun rakenteen ja kustannusten osalta.</p>	kustomointia, konfigurointia

# Prosessit kehittämisessä



Liite 5. Vaatimusdokumentaatio

**Liiketoiminnalliset vaatimukset**

Tunniste*	Vaatimuksen nimi*	Vaatimuksen kuvaus*	Käyttäjryhmä	Prosessi /Toiminto	Lähde ja pvm *	Prioriteetti*	Tila*
LTV001	Vuosisuunnittelu ja ennustaminen hanketasolla	Vuosisuunnittelu ja ennustaminen tehdään hanketasolla. Vuosisuunnitteluun kuuluville hankkeille tehdään vuosiennuste budjetit. Budjetin avulla baselinetaan hanketasolla, yrityksittäin vuosisuunnittelun euromäärä. Vuosisuunnittelun tiedot viedään suunnittelun työkaluun.	Kehittämisen johtamisen tuki	Vuosisuunnittelu	Luotu havaittujen toiminnallisten vaatimusten pohjalta 11.4.2013	1-Välttämätön	Hyväksytty
LTV002	Hankkeen kustannusten arviointi ja seuranta	Hankkeen kustannusarvion pitää voida luoda automaattisesti valittujen tietojen perusteella. Kustannusarviosta luotavalla budjetilla on luokittelutekijöitä jotka kuvaavat budjetin tyyppiä. Hankkeen kustannukset eivät ole yhtä kuin hankkeeseen kuuluvien projektien kustannukset. Hankkeille on saatava raportointiin myös ei tuntiperusteiset SAP -järjestelmässä olevat toteutuneet kustannukset ostettujen palveluiden osalta. Kustannukset jotka eivät tule SAP -järjestelmästä on voitava lisätä projektille suoraan manuaalisesti.	Hankevastaava, projektipäällikkö	Hankehallinta, Projektinhallinta	Luotu havaittujen toiminnallisten vaatimusten pohjalta 11.4.2013	1-Välttämätön	Hyväksytty
LTV003	Hankkeen lisätiedot ja luokittelutekijät	Hankkeella on oltava lisätietoina tilanne, päätöspiste, kustannuspaikka (useita), hanketieto, riippuvuudet, liiketoiminnan työ sekä määrättyjä luokittelutekijä tietoja.	Hankevastaava, kehittämisen johtamisen tuki	Hankehallinta	Luotu havaittujen toiminnallisten vaatimusten pohjalta 11.4.2013	1-Välttämätön	Hyväksytty
LTV004	Alueperusteiset oikeudet hanketietoihin	Eri alueiden hankeavastavat eivät saa nähdä toisten alueiden hanketietoja.	Hankevastaava	Hankehallinta	Luotu havaittujen toiminnallisten vaatimusten pohjalta 11.4.2013	1-Välttämätön	Hyväksytty
LTV005	Hanke ja projekti kuuluvat salkkuun	Hanke ja projekti kuuluvat niiden kustannuspaikan mukaisesti salkkuun. Kustannuspaikat on jaettu salkkuihin ja hanke sekä projekti saa automaattisesti salkun päämaksajan (kustannuspaikan) mukaan.	Palvelupäällikkö	Hankehallinta	Luotu havaittujen toiminnallisten vaatimusten pohjalta 11.4.2013	1-Välttämätön	Hyväksytty
LTV006	Projektin lisätiedot ja luokittelutekijät	Hankkeella on oltava lisätietoina tilanne Ja luokittelutekijöinä VSOP -vaihe sekä Kehitysprosessin vaihe. Tehtävällä on oma luokittelutekijä tehtävän luokka.	Projektipäällikkö	Projektinhallinta	Luotu havaittujen toiminnallisten vaatimusten pohjalta 11.4.2013	1-Välttämätön	Hyväksytty
LTV007	Projektin kustannusten arviointi ja seuranta	Projektin kustannusarvion pitää voida luoda automaattisesti projektisuunnitelman pohjalta. Kunnossapidon projekteilla on oltava budjetin tyyppi luokittelutekijä. Vain SAP -järjestelmässä olevia kulu kustannuksia on voitava syöttää projektin seurantaan toteutuneiksi kustannuksiksi.	Projektipäällikkö	Projektinhallinta	Luotu havaittujen toiminnallisten vaatimusten pohjalta 11.4.2013	1-Välttämätön	Hyväksytty

LTV008	Resursointikokonaisuus	Resursointikokonaisuus on voitava tunnistaa resursointikokonaisuudeksi. Siihen kuuluvat projektit ja hankkeet on voitava tunnistaa sekä hankkeen tai projektin tiedoista on löydettävä mihin resursointikokonaisuuteen se kuuluu. Lisäksi resursointikokonaisuuksien vastaavilla on oltava käyttöoikeuksiin resursointikokonaisuuteen varattujen henkilöiden kiinnittämiseen toteuttaville projekteille.	Reko -vastaava	Resurssinhallinta	Luotu havaittujen toiminnallisten vaatimusten pohjalta 11.4.2013	1-Välttämätön	Hyväksytty
LTV009	Resursointi resursointikokonaisuuden kautta	Resursointikokonaisuudesta tehdään yksi- (resurssipyynnö) tai kaksivaiheinen (roolipyynnöt ja resurssipyynnöt) resursointi kapasiteetin hallintaan kuuluvien henkilöiden osalta. Resursointikokonaisuuden vastaava saa kiinnittää esimiesten lupaamia tulevaisuuteen vahvistettuja henkilöitä resursointikokonaisuuteen kuuluviin projekteihin.	Reko -vastaava, esimiehet	Resurssinhallinta	Luotu havaittujen toiminnallisten vaatimusten pohjalta 11.4.2013	1-Välttämätön	Hyväksytty
LTV010	Resurssien lisätiedot	IT:n toimihenkilöt sekä konsultit kuuluvat roolipohjaiseen kapasiteetin hallintaan. Kapasiteetin hallinta tapahtuu henkilön roolitiedon lisäksi organisaatiotiedon kautta.	Esimiehet	Resurssinhallinta	Luotu havaittujen toiminnallisten vaatimusten pohjalta 11.4.2013	1-Välttämätön	Hyväksytty
LTV011	Henkilötyön laskutuksen yleiset ehdot	Henkilötyön laskutuksen aikataulun mukaan toimitetaan SAP -järjestelmään laskutuksen liittämätiedostot jokaisen kuukauden sovittuna päivänä. Ohjelmisto on laskutuksen master ja laskutuksen korjaukset tehdään projekteille kirjattuja toteumia huottaamatta.	Taloudesta vastaavat toimijat	Laskutus	Luotu havaittujen toiminnallisten vaatimusten pohjalta 11.4.2013	1-Välttämätön	Hyväksytty
LTV012	Henkilötyön hinnoittelu	Laskutuksessa on oltava mahdollisuus asiakaskohtaisiin hintoihin. IT henkilöillä laskutettava tuntihinta on rooliin sidonnainen. Tuntihintaa on voitava muuttaa tehtäväkohtaisesti. Alihankkijoilla tuntihinta on alihankkijan kanssa tehdyssä sopimuksessa sovittu ja tuntihinnan lisäksi veloitetaan vakiohintainen hallintolisä. Tuntihintaa ja hallintolisää vastaa nimike, joka jakautuvat tehtävän työn mukaan.	Taloudesta vastaavat toimijat	Laskutus	Luotu havaittujen toiminnallisten vaatimusten pohjalta 11.4.2013	1-Välttämätön	Hyväksytty
LTV013	Henkilötyön laskutuksen tiedot	Tilausnumeroa välitetään SAP -järjestelmään henkilötyön laskutuksessa laskutusliittymässä, sisäisen laskennan liittymässä sekä oman työn aktivoinnin liittymässä. Tilausnumero kertoo onko veloitus kulua vai aktivoitavaa. Laskutuksessa välittyy lisäksi palvelutuotenumero ja sisäisen laskennan veloituksissa tilinumero, jonne veloitukset kohdistuvat. Näitä palvelutuotetta ja tilinumeroa vastaavia tilinumeroita sekä tilausnumeroita käytetään konsulttilaskujen tilioinnissa.	Taloudesta vastaavat toimijat	Laskutus	Luotu havaittujen toiminnallisten vaatimusten pohjalta 11.4.2013	1-Välttämätön	Hyväksytty
LTV014	Projektin maksajatiedot, Resurssien tulonsaantitiedot	Projektin maksajat voi olla yksi tai useampi kustannuspaikka. Maksaja on voitava määrittää projektilla tehtävätasolla. Jokaisella laskuttavalla henkilöllä on kustannuspaikka tieto joka välittyy laskutukseen tulonsaajana. Lisäksi konsulteilla on lisätietona erilaisia laskutukseen liittyviä vastuuhenkilöitä.	Taloudesta vastaavat toimijat	Laskutus	Luotu havaittujen toiminnallisten vaatimusten pohjalta 11.4.2013	1-Välttämätön	Hyväksytty
LTV015	P.paveluiden laskutus	P.paveluiden laskutus toteutetaan kappalemääräisenä, nimike määrittää tulonsaajan ja hinta on asiakaskohtaisesti määritettävä.	Taloudesta vastaavat toimijat	Laskutus	Luotu havaittujen toiminnallisten vaatimusten pohjalta 11.4.2013	1-Välttämätön	Hyväksytty

## Tietojärjestelmän toiminnalliset vaatimukset

Tunniste*	Vaatumuksen nimi*	Vaatumuksen kuvaus*	Käyttäjäryhmä	Prosessi / Toiminto	Lähde ja pvm*	Prioriteetti*	Tila*	Toteutus-erä	Vapaa sarake
TJV001	Hankeen kuuluminen vuosisuunnitteluun	Hankkeilla pitää olla luokittelutekijä, jolla ilmoitetaan kuuluuko hanke vuosisuunnitteluun. Luokittelutekijän arvot: "Kyllä", "Ei", "Ei otettu kantaa". Tietoa ei historoida.	Kehittämisen johtamisen tuki	Vuosisuunnittelu	LTV001	1-Välttämätön	Hyväksytty	Erä1	Attribuutti, Konfiguraatio
TJV002	Hanke ehdolla seuraavaan vuosisuunnitteluun	Hankkeella pitää olla luokittelutekijä, jonka avulla ilmoitetaan onko hanke ehdolla seuraavaan vuosisuunnitteluun. Hanke joko on ehdolla tai ei ole ehdolla seuraavaan vuosisuunnitteluun. Tietoa ei historoida.	Kehittämisen johtamisen tuki	Vuosisuunnittelu	LTV001	1-Välttämätön	Hyväksytty	Erä1	Attribuutti, Konfiguraatio
TJV003	Vuosisuunnitteluun kuuluville hankkeille vuosienbudjetit	Valitun alueen vuosisuunnitteluun kuuluville hankkeille pitää voida luoda automaattisesti voimassa olevasta kustannusarviosta vuosisuunnittelun budjetti. Budjetti nimetään "vvvv vuosienbudjetin tavoite". Budjettiin viedään kunkin hankkeen kustannusarviossa olevat vuosisuunnitteluun / raamiin kuuluvat kustannuslajit ja niille suunnitellut ennusteet. Budjetteja ei muokata jälkikäteen, korjaukset tehdään uusiin budjetteihin.	Kehittämisen johtamisen tuki	Vuosisuunnittelu	LTV001	1-Välttämätön	Hyväksytty	Erä1	Objekti, Taustaprosessi, Konfiguraatio
TJV004	Vuosisuunnitteluun kuuluvien hankkeiden vuosienbudjettien tarkastus kvartaaleittain	Valittuun vuosisuunnitteluun kuuluville hankkeille pitää voida luoda automaattisesti voimassa olevasta kustannusarviosta tarkastettu vuosisuunnittelun budjetti. Tarkastettu budjetti luodaan erikseen määritettyinä ajankohtana. Tarkastuksen tehdään kvartaaleittain. Budjetti nimetään "vvvv vuosienbudjetin Qx tarkistettu". Budjettiin viedään kunkin hankkeen kustannusarviossa olevat vuosisuunnitteluun / raamiin kuuluvat kustannuslajit ja niille suunnitellut ennusteet. Budjetteja ei muokata jälkikäteen, korjaukset tehdään uusiin budjetteihin	Kehittämisen johtamisen tuki	Vuosisuunnittelu	LTV001	1-Välttämätön	Hyväksytty	Erä1	Objekti, Taustaprosessi, Konfiguraatio
TJV005	Vuosisuunnitteluun kuuluvien hankkeiden vuosienbudjetit	Vuosisuunnitteluun kuuluvien hankkeiden vuosienbudjetit pitää viedä ohjelmistosta suunnittelun työkaluun. Liittymätiedoston tulee sisältää määrittämissä kuvatut tiedot.	Kehittämisen johtamisen tuki, vuosisuunnittelussa mukana olevat tahot	Vuosisuunnittelu	LTV001	1-Välttämätön	Hyväksytty	Erä1	SQL kysely toimii liittymänä Konfiguraatio

TJV006	Hankkeen kustannusarvion luonnin automatisointi	Hankkeen kustannusarvion luontiin pitää saada avuksi automatiikkaa. Automatiikan avulla haetaan menneen ajan (valittuun kk asti) ennuste eurot hankkeisiin kuuluvien projektien toteumista ja tulevat ennusteet hankkeen edellisestä kustannusarviosta. Lisäksi kuluvan vuoden summat pitää voida säilyttää halutessaan samana kustannuslaji ja kulu/aktivoitava tasolla.	Hankevastaava	Hankehallinta	LTV002	1-Välttämätön	Hyväksytty	Erä1	Objekti, Tausta-prosessi, Funktio Konfiguraatio & Kustomointi
TJV007	Hankkeen ennusteet eivät ole yhtä kuin siihen kuuluvien toteuttavien projektien ennusteet	Hankenäkymässä ennusteiden summautuminen on estettävä, niin että hankkeen ennusteet eivät ole yhtä kuin siihen kuuluvien projektien ennusteet. Hankkeella on omat ennusteet (kustannusarviot ja budjetit) ja projekteilla on omat ennusteet (kustannusarviot ja budjetit).	Hankevastaava, hankkeen tietojen seuraajat	Hankehallinta	LTV002	1-Välttämätön	Hyväksytty	Erä1	Triggeri Kustomointi
TJV008	Kustannukset SAP -järjestelmästä	Ostettujen palveluiden kustannukset pitää tuoda SAP -järjestelmästä. Kustannukset poimitaan SAP -järjestelmästä valittujen kulutilien perusteella ja tuodaan ohjelmistoon. Toteutuneet kustannukset kohdistetaan projekteille tilausnumeron ja kulutilin perusteella. Mikäli projektia ei voida päätellä, on kohdistus pystyttävä tekemään manuaalisesti.	Hankevastaava	Hankehallinta	LTV002	1-Välttämätön	Hyväksytty	Erä1	Liittymä, Funktio Konfiguraatio & Kustomointi
TJV009	Hankkeen budjetin lisätieto päätös	Korjaukset tehdään negatiivisilla vienneillä. Liittymätiedoston tulee sisältää määrityksessä kuvatut tiedot.	Hankevastaava	Hankehallinta	LTV002	1-Välttämätön	Hyväksytty	Erä1	Attribuutti Konfiguraatio
TJV010	Hankkeen budjetin lisätieto muu tyyppi	Hankkeen budjetissa pitää olla lisätieto hankkeen budjetin muu tyyppi. Lisätieto kuvaa budjetin muuta, ei päätöspisteen mukaista, tyyppiä.	Hankevastaava	Hankehallinta	LTV002	1-Välttämätön	Hyväksytty	Erä1	Attribuutti Konfiguraatio
TJV011	Hankkeen budjetin lisätieto hankkeen budjetin tyyppi	Hankkeen budjetilla pitää olla lisätieto hankkeen budjetin tyyppi listausta varten. Hankkeen budjetin tyyppi kentän valintalistan arvot päivittyvät automaattisesti sen mukaan kumman tyyppin käyttäjä on valinnut; budjetin päätöksen (TJV009) vai hankkeen budjetin muu (TJV010) tyyppin.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV002	1-Välttämätön	Hyväksytty	Erä2	Attribuutti, Triggeri Kustomointi
TJV012	Oikeudet hanketietoihin	Eri hankeavastavilla on oltava oikeudet vain valittuihin hankkeisiin ja näkymä niiden tietoihin. Oikeudet annetaan aina myös uusiin hankkeisiin.	Hankevastaava	Hankehallinta	LTV004	1-Välttämätön	Hyväksytty	Erä1	Triggeri, Kustomointi
TJV013	Hanke kuuluu salkkuun	Hanke kuuluu sen kustannuspaikan mukaan salkkuun. Kustannuspaikat on jaettu salkkuihin ja hanke saa salkun päämaksajansa mukaan.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV005	1-Välttämätön	Hyväksytty	Erä1	Triggeri, Kustomointi

TJV014	Hankkeella useampia kustannuspaikkoja	Hankkeen ennusteet ja toteumat on voitava jakaa % -perusteisesti useammalle kustannuspaikalle raportoinnissa ja ennustamisessa. Osuudet ovat aikaan sidonnaisia ja ne historoidaan.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Objekti, Konfiguraatio
TJV015	Hankkeen tiedot tallennetaan hankkeelle	Hanketietoihin kirjatut tiedot on voitava tallentaa niitä vastaaviin kenttiin ohjelmistoon. Hankkeet perustetaan tietojen perusteella ja perustuksessa käytetyt kirjatut hankkeen tiedot tallennetaan myös ohjelmistoon.	Hankevastaava	Hankehallinta	LTV003	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Objekti, Konfiguraatio
TJV016	Hankkeiden väliset riippuvuudet	Hankkeelle on voitava antaa tiedoksi riippuvuudet muihin hankkeisiin. Tietoina annetaan hankkeen nimi johon riippuvuus on, riippuvuuden kuvaus ja ajankohta. Tietoa ei historoida.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Objekti, Konfiguraatio
TJV017	Hankkeella sisäinen työ	Hankkeelle on voitava antaa tiedoksi sisäisen työn määrä aikajaksittain. Tietoa ei historoida.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Objekti, Konfiguraatio
TJV018	Hankkeen päätöspiste tiedon tallennus	Hankkeelle pitää voida lisätä päätöspisteet. Päätöspisteestä tallennetaan päätöspiste (ennalta määritetyt päätöspisteet valintalistalta), muu päätöspiste ja päivämäärä. Päätöspisteelle annetaan lisätietoja. Päätöspisteitä voi antaa useampia, ne tallennetaan ja niitä voi päivittää.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Objekti, Taustaprosessi, Konfiguraatio
TJV019	Hankkeen tilanne tiedon tallennus	Hankkeelle pitää voida lisätä status / tilanne tieto. Tilanne sisältää sanallisen kuvauksen sekä liikennevalon värin (ennalta määritetyt värit valintalistalta) neljästä eri statuksesta (aikataulu, kustannus, sisältö ja yleinen). Tilanne annetaan tilanapäivällä. Niitä voi antaa useampia, ei tallenneta ja niitä voi päivittää.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Objekti, Konfiguraatio
TJV020	Hankkeen strategian mukaisuus	Hankkeella pitää voida antaa tieto strategian mukaisuudesta. Annettujen arvojen mukaan lasketaan automaattisesti strategian mukaisuuden numeerinen arvo.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Objekti, Konfiguraatio
TJV021	Hankkeen luokittelutekijä: Kehityshankkeen luokittelu	Hankkeella on oltava lisätietokenttä, johon valitaan valintalistasta hankkeen kehityshankkeen luokittelu tieto. Vain kehittämisen hankkeilla. Ei historoida.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Attribuutti, Konfiguraatio
TJV022	Hankkeen luokittelutekijä: Kehitysprosessin vaihe	Hankkeella on oltava lisätietokenttä, johon voidaan valita valintalistasta kehitysprosessin vaihe (ennalta määritetty vaihe valintalistalta) jossa hanke on. Tieto päivittyy päätöspisteen (TJV017) mukaan mutta kenttää voi myös päivittää manuaalisesti. Vain kehittämisen hankkeilla. Ei historoida.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Attribuutti, Konfiguraatio

TJV023	Hankkeen luokittelutekijä: Kehityssuunnitelma	Hankkeella on oltava lisätietokenttä, johon voidaan valita valintalistasta kehityssuunnitelma, jota hanke toteuttaa. Vain kehittämisen hankkeilla. Ei historoita.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1-Välttämätön	Hyväksytty	Erä1	Attribuutti, Konfiguraatio
TJV024	Hankkeen luokittelutekijä: Vaikutusalue	Hankkeella on oltava lisätietokenttä, johon voidaan valita valintalistasta hankkeen vaikutusalue. Vain kehittämisen hankkeilla. Ei historoita.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1-Välttämätön	Hyväksytty	Erä1	Attribuutti, Konfiguraatio
TJV025	Hankkeen luokittelutekijä: Projektisalkku	Hankkeella on oltava lisätietokenttä, johon voidaan valita valintalistasta hankkeen projektisalkku. Vain kehittämisen hankkeilla. Ei historoita.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1-Välttämätön	Hyväksytty	Erä1	Attribuutti, Konfiguraatio
TJV026	Hankkeen luokittelutekijä: Mittarihanke	Hankkeella on oltava lisätietokenttä, jolla ilmoitetaan onko hanke mittarihanke. Vain kehittämisen hankkeilla. Ei historoita.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1-Välttämätön	Hyväksytty	Erä1	Attribuutti, Konfiguraatio
TJV027	Hankkeen luokittelutekijä: Prioriteetti	Hankkeella on oltava lisätietokenttä, johon voidaan kirjoittaa hankkeen Prioriteetti. Vain kehittämisen hankkeilla. Ei historoita.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1-Välttämätön	Hyväksytty	Erä1	Attribuutti, Konfiguraatio
TJV028	Hankkeen luokittelutekijä: Kehityssalkku	Hankkeella on oltava lisätietokenttä, johon voidaan valita valintalistasta hankkeen kehityssalkku. Vain kehittämisen hankkeilla. Ei historoita.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1-Välttämätön	Hyväksytty	Erä1	Attribuutti, Konfiguraatio
TJV029	Hankkeen luokittelutekijä: Kehityssalkun tarkenne	Hankkeella on oltava lisätietokenttä, johon voidaan valita valintalistasta hankkeen Kehityssalkun tarkenne. Vain kehittämisen hankkeilla. Ei historoita.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1-Välttämätön	Hyväksytty	Erä1	Attribuutti, Konfiguraatio
TJV030	Hankkeen luokittelutekijä: Sovellussalkku	Hankkeella on oltava lisätietokenttä, johon voidaan valita valintalistasta hankkeen sovellussalkku. Vain kunnossapidon ja pienkehittämisen hankkeilla. Ei historoita.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1-Välttämätön	Hyväksytty	Erä1	Attribuutti, Konfiguraatio
TJV031	Hankkeen lisätieto: Tila	Hankkeella on oltava pakollinen tieto, jolla kerrotaan hankkeen tila. Valintalistasta arvot Uusi, Käynnissä, Keskeytetty, Valmis. Oletusarvona tila on uusi. Ei historoita.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1-Välttämätön	Hyväksytty	Erä1	Attribuutti, Konfiguraatio
TJV032	Hankkeen lisätieto: Varavastuullinen	Hankkeella on oltava lisätietokenttä, johon voidaan valita valintalistasta hankkeen varavastava toimiva henkilö. Henkilölle annetaan toimintaan tarvittavat oikeudet oikeusmäärittelyn kautta. Ei historoita.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1-Välttämätön	Hyväksytty	Erä1	Attribuutti, Konfiguraatio
TJV033	Hankkeen lisätieto: Liiketoimintavastaava	Hankkeella on oltava lisätietokenttä, johon voidaan valita valintalistasta hankkeen liiketoimintavastaava toimiva henkilö. Ei historoita.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1-Välttämätön	Hyväksytty	Erä1	Attribuutti, Konfiguraatio



TJV034	Hankkeen lisätieto: Hankkeen johtoryhmän puheenjohtaja	Hankkeella on oltava vapaamuotoinen tekstikenttä, johon voidaan kirjoittaa johtoryhmä puheenjohtajan nimi. Ei historoia.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1-Välttämätön	Hyväksytty	Erä1	Attribuutti, Konfiguraatio
TJV035	Hankkeen lisätieto: Hankkeen johtoryhmä	Hankkeella on oltava vapaamuotoinen tekstikenttä, johon voidaan luetella johtoryhmän jäsenet. Ei historoia.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1-Välttämätön	Hyväksytty	Erä1	Attribuutti, Konfiguraatio
TJV036	Hankkeen lisätieto: Hankkeen materiaalin tallennuspaikka	Hankkeella on oltava materiaalin tallennuspaikka tieto, johon voidaan tallentaa hankkeen sivuston url. Ei historoia.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1-Välttämätön	Hyväksytty	Erä1	Attribuutti, Konfiguraatio
TJV037	Hankkeen lisätieto: Hankkeen tavoite	Hankkeella on oltava vapaamuotoinen tekstikenttä hankkeen tavoitteen dokumentointiin, ei historoia.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1-Välttämätön	Hyväksytty	Erä1	Attribuutti, Konfiguraatio
TJV038	Hankkeen lisätieto: Huomionarvoista	Hankkeella on oltava vapaamuotoinen tekstikenttä hankkeen tavoitteen lisätiedoksi tai muille huomionarvoisille asioille, ei historoia.	Hankevastaava, hanketietojen hyödyntäjät	Hankehallinta	LTV003	1-Välttämätön	Hyväksytty	Erä1	Attribuutti, Konfiguraatio
TJV039	Projektin tilanne tiedon tallennus	Projektille on voitava lisätä tilanne tieto, joka sisältää valmiusastetta kuvaavan prosentin (projektipäällikön arvio), tilannetta kuvaavan liikennevalon värin, sanallisen kuvauksen tilanteesta sekä tilanpäivä päivämäärätiedon. Niitä voi antaa useampia, niitä ei voi päivittää jälkikäteen ja ne historoidaan.	Projektipäällikkö, projektitietojen hyödyntäjät	Projektinhallinta	LTV006	1-Välttämätön	Hyväksytty	Erä1	Objekti, Taustaprosessi Konfiguraatio
TJV040	Projektin luokittelutekijä: Elinkaaren VSOP vaihe	Projektilla on oltava pakollisena tietona sen VSOP vaihe tieto. VSOP vaiheet ovat valmistelu, suunnittelu, ohjaus, päättäminen, oletusarvona on valmistelu. Käytetään kehittämisen ja pienkehittämisen projekteilla. Ei historoia.	Projektipäällikkö, projektitietojen hyödyntäjät	Projektinhallinta	LTV006	1-Välttämätön	Hyväksytty	Erä1	Attribuutti, Konfiguraatio
TJV041	Projektin luokittelutekijä: Kehitysprosessin vaihe	Projektilla on oltava valittavana tieto sen Kehitysprosessin vaiheesta. Tieto valitaan valintalistasta (ennalta määritetty vaihe valintalista). Ei historoia.	Projektipäällikkö, projektitietojen hyödyntäjät	Projektinhallinta	LTV006	1-Välttämätön	Hyväksytty	Erä1	Attribuutti, Konfiguraatio
TJV042	Projektin tehtävän luokittelutekijä: Tehtävän luokittelu	Projektin tehtävällä on oltava lisätieto tehtävän luokittelu, jossa voidaan valita valintalistasta minkä tyyppistä työtä tehtävällä tehdään. Käytössä kehittämisen ja pienkehittämisen projekteissa. Ei historoia.	Projektipäällikkö	Projektinhallinta	LTV006	1-Välttämätön	Hyväksytty	Erä1	Attribuutti Konfiguraatio
TJV043	Projektin budjetin lisätieto: Kunnossapitoprojektin budjetin tyyppi	Projektin budjetilla on oltava lisätieto budjetin tyyppi. Käytetään kunnossapito- ja pienkehitysprojekteilla. Tietoa hyödynnetään raportoinnissa.	Projektipäällikkö	Projektinhallinta	LTV007	1-Välttämätön	Hyväksytty	Erä1	Attribuutti Konfiguraatio

TJV044	Projektin kustannusarvion luonti automatisoidusti työsuunnitelmasta tai varauksista	Projektin kustannusarvion luontiin pitää saada avuksi automatiikkaa. Automatiikan avulla haetaan suunnitellut kustannukset joko työsuunnitelmasta tai resurssivarauksista. Rooleille suunnitellun työn tai varausten on noustava mukaan kustannusarvion suunniteltuihin kustannuksiin / ennusteeseen oikealle kustannuslajille. Ryhmitely tehdään kulu -aktivoivat tiedon sekä kustannuslajin mukaan. Hinnoissa käytetään laskutuksessa käyttämiä hintoja 1 kk:den tarkkuudella.	Projektipäällikkö	Projektinhallinta	LTV007	1-Välttämätön	Hyväksytty	Erä1	Objekti, Taustaprosessi, Funktio Konfiguraatio & Kustomointi
TJV045	Projektille toteutuneiden kustannusten syöttö	Projektille on voitava syöttää kustannuksia toteutuneiksi. Toteutuneiksi kustannuksiksi syötetään vain kustannukset, jotka eivät ole tuntiperusteisia ja niitä ei suunnitella tehtäväsuunnitelmassa. Kustannukset kohdistetaan tehtävälle ja tietyille kustannuslajille. Syöttäessä on voitava valita päiviä jonne kustannus kohdistuu raportoinnissa, mille kustannuslajille kustannus kohdistuu ja sen euromäärä.	Projektipäällikkö	Projektinhallinta	LTV002	1-Välttämätön	Hyväksytty	Erä1	Objekti, taustaprosessi Konfiguraatio
TJV046	Projekti kuuluu palvelupäällikön salkkuun	Projekti kuuluu sen kustannuspaikan mukaan palvelupäällikön salkkuun. Kustannuspaikat on jaettu salkkuihin ja projekti saa salkun sen päämaksajan mukaan.	Projektipäällikkö, projektitietojen hyödyntäjät	Projektinhallinta	LTV005	1-Välttämätön	Hyväksytty	Erä1	Triggeri, Kustomointi
TJV047	Resursointikokonaisuus projektin tunnistaminen	Resursointikokonaisuus projekti on pystyttävä tunnistamaan / erottamaan muista projekteista.	Reko-vastaava	Resurssinhallinta	LTV008	1-Välttämätön	Hyväksytty	Erä1	Konfiguraatio
TJV048	Resursointikokonaisuuteen kuuluvat projektit ja hankkeet	Resursointikokonaisuuteen kuuluvat projektit ja hankkeet on pystyttävä tunnistamaan. Resursointikokonaisuuteen kuuluvat projektit ja hankkeet on pystyttävä hankemaan resursointikokonaisuuden kautta.	Reko-vastaava	Resurssinhallinta	LTV008	1-Välttämätön	Hyväksytty	Erä1	Konfiguraatio
TJV049	Hankkeen resursointikokonaisuus	Hankkeelta on pystyttävä näkemään mihin resursointikokonaisuuteen se kuuluu.	Hankevastaava, Reko-vastaava	Resurssinhallinta	LTV008	1-Välttämätön	Hyväksytty	Erä1	Attribuutti, Konfiguraatio
TJV050	Projektin resursointikokonaisuus	Projektilta on pystyttävä näkemään mihin resursointikokonaisuuteen se kuuluu.	Projektipäällikkö, Reko-vastaava	Resurssinhallinta	LTV008	1-Välttämätön	Hyväksytty	Erä1	Attribuutti, Konfiguraatio
TJV051	Projektin resursointikokonaisuus määräytyy hankkeen kautta	Projektilla on oltava sama resursointikokonaisuustieto kuin hankkeella. Kun projekti kiinnitetään hankkeeseen, se saa automaattisesti saman resursointikokonaisuuden kuin hankkeella on.	Hankevastaava, Reko-vastaava	Resurssinhallinta	LTV008	1-Välttämätön	Hyväksytty	Erä1	Triggeri, Kustomointi

TJV052	Resursointikokonaisuuden vastaavan bookkaus oikeudet	Resursointikokonaisuuden vastaavalla on oltava oikeudet kiinnittää resursointikokonaisuudessa tulevaisuuteen vahvistetusti varattua (hard bookattu) resurssi resursointikokonaisuuteen kuuluvaan toteuttavaan projektin projektiryhmään alustavalla varauksella (soft book).	Reko-vastaava	Resurssinhallinta	LTV009	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Proseduuri, Triggeri, Kustomointi
TJV053	Resurssipyynnön lisätieto	Resurssipyynnössä (Requisitionin) on oltava tieto onko pyyntö tehty resursointikokonaisuudesta vai toteuttavasta projektista.	Henkilöesimies	Resurssinhallinta	LTV009	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Attribuutti, Konfiguraatio
TJV054	Resurssipyynnön luominen vain resursointikokonaisuudelta	Resurssipyynnöjä saa luoda vain resursointikokonaisuuksista. Pyyntöjen luominen hanketta toteuttavasta projektista joka ei ole myös res.kok, on estettävä.	Reko-vastaava	Resurssinhallinta	LTV009	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Taustaprosessi, funktio, proseduurin Kustomointi
TJV055	Roolipyyntö	Roolipyynnöstä on lähetettävä tieto järjestelmässä resursointivastaavalle. Resursointivastaava saa pyynnössä tiedon projektista, jonka roolita-son kapasiteettitarve odottaa hyväksyntää. Resursointivastaava käsittelee varauksen sekä sulkeen pyynnön (Action Itemin).	Resursointivastaava	Resurssinhallinta	LTV009	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Prosessi, Konfiguraatio
TJV056	Roolipyynnön tila	Roolipyynnön käsittelyn pitää päivittää projektilla olevaa tietokenttää, josta projektipäällikkö näkee käsittelyn tilan. Hyväksynnän jälkeen projektille jää hyväksyjäkenttään tieto hyväksyjästä.	Reko-vastaava	Resurssinhallinta	LTV009	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Attribuutti, Prosessi Konfiguraatio
TJV057	Roolipyynnössä kuvaus roolitarpeesta	Roolipyynnössä, roolin varaustietojen lisäksi on kuvaus roolitarpeesta. Roolipyynnön käsittelijä näkee nämä tiedot roolipyynnön lähettävän projektin roolin varaustiedoista.	Resursointivastaava	Resurssinhallinta	LTV009	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Attribuutti, Konfiguraatio
TJV058	Resurssien organisaatitieto	Resurssien organisaation tieto on oltava aikasidonnainen ja se historoidaan.	Henkilöesimies	Resurssinhallinta	LTV010	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Objekti, Prosessi Konfiguraatio
TJV059	SAP järjestelmään laskutuksen liittämätiedosto jonka SAP tulostaa laskuiksi	Laskutusaineisto pitää viedä SAP- järjestelmään. Laskutus tehdään SAP:ssa lähetetyn aineiston pohjalta. Laskutus perustuu tehtäväkortille tehtyihin tuntikirjauksiin. Liittymätiedoston tulee sisältää määrittämissä kuvatut tiedot.	Laskuttava IT	Laskutus	LTV011	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Liittymä
TJV060	SAP- järjestelmään oman työn aktivoinnin tiedosto jonka pohjalta SAP tekee sisäiset veloitus ja siirrot	Oman työn aktivoinnin aineisto pitää viedä pitää viedä SAP -järjestelmään. Sisäiset veloitus oman aktivoitavan työn osalta tehdään SAP:ssa lähetetyn aineiston pohjalta. Sisäiset veloitus perustuu tehtäväkortille tehtyihin tuntikirjauksiin. Liittymätiedoston tulee sisältää määrittämissä kuvatut tiedot.	Laskuttava IT	Laskutus	LTV011	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Liittymä

TJV061	SAP -järjestelmään sisäisen laskennan tiedosto jonka pohjalta SAP tekee sisäiset veloitukset ja siirrot	Sisäiset veloitukset pitää viedä pitää viedä SAP -järjestelmään. Sisäiset veloitukset tehdään SAP:ssa lähetetyn aineiston pohjalta. Sisäiset veloitukset perustuvat tehtäväkortille tehtyihin tuntikirjauksiin. Liittymätiedoston tulee sisältää määrityksessä kuvatut tiedot.	Laskuttava IT	Laskutus	LTV011	1- Välttämätön	Hyväksytty	Erä1	Liittymä
TJV062	Laskutus ja sisäiset veloitukset kuukausittain sovittuna päivänä	Laskutus ja sisäiset veloitukset tehdään kuukausittain sovittuna päivänä ja lähetetään sen jälkeen SAP järjestelmään.	Laskuttava IT	Laskutus	LTV011	1- Välttämätön	Hyväksytty	Erä1	Konfiguraatio, Kustomointi
TJV063	Laskutuksen korjaukset	Laskutuksessa tulleet virheet on pysyttävä korjaamaan takautuvasti, niin että hyvitys ja veloitus tehdään sen laskutuskuukauden tiedoilla joihin korjaus kohdistuu. Korjaus lähetetään aina korjauskuukauden laskutuksen tai sisäisen laskennan aineistoissa SAP järjestelmään. Projektien veloitusten kohdistustiedon korjaus ei saa muuttaa toteumatietoja joita korjaus koskee.	Laskuttava IT	Laskutus	LTV011	1- Välttämätön	Hyväksytty	Erä1	Konfiguraatio, Kustomointi
TJV064	Laskutuksessa käytetään asiakaskohtaisia hintoja	Eri asiakas ryhmillä käytetään laskutuksessa eri veloitushintoja. Sama henkilö voi laskuttaa eri hinnalla asiakasryhmästä riippuen.	Laskuttava IT	Laskutus	LTV012	1- Välttämätön	Hyväksytty	Erä1	Konfiguraatio, Kustomointi
TJV065	IT:n henkilötyön laskutushinta riippuu roolista	Laskutettavan IT henkilötyön veloitus perustuu rooliin, joka henkilöllä on. Roolit on jaettu hintojen mukaisiin kolmeen hintaryhmään.	Laskuttava IT	Laskutus	LTV012	1- Välttämätön	Hyväksytty	Erä1	Konfiguraatio, Kustomointi
TJV066	IT:n henkilötyön laskutuksessa henkilön oletushintaa on voitava muuttaa	Henkilön laskutushinta määräytyy tehtäväkohtaisen roolin mukaan, joka voidaan muuttaa tehtäväkohtaisesti.	Laskuttava IT	Laskutus	LTV012	1- Välttämätön	Hyväksytty	Erä1	Konfiguraatio, Kustomointi
TJV067	Alihankkijoilla henkilötyön laskutushinta määräytyy konsulttisopimuksen mukaan	Jokaiselle ohjelmiston kauttalaskuttavalle alihankkijalle on voitava lisätä konsulttikohtainen laskutushinta. Hinta on aikasidonnainen ja se historoidaan.	Laskuttava IT	Laskutus	LTV012	1- Välttämätön	Hyväksytty	Erä1	Konfiguraatio, Kustomointi
TJV068	Alihankkijoilla veloitetaan projektien asiakkailta tuntihinnan lisäksi hallintolisä	Alihankkijoiden tuntihintaan lisätään sivukulun (€ per h) osuus. Raportoinnissa hintana näytetään sivukulullista hintaa. Laskutuksessa hallintolisä ja tuntihinta erotetaan. Sivukulu on aikasidonnainen ja se historoidaan.	Laskuttava IT	Laskutus	LTV012	1- Välttämätön	Hyväksytty	Erä1	Konfiguraatio, Kustomointi
TJV069	Laskutushintaa on voitava muuttaa	Laskutuksessa henkilöillä käytettävää hintaa on pystyttävä muuttamaan takautuvasti. Laskutuksessa käytetty hinta historoidaan.	Laskuttava IT	Laskutus	LTV012	1- Välttämätön	Hyväksytty	Erä1	Konfiguraatio, Kustomointi

TJV070	Laskutushintaa vastaa nimike	Laskutushinnoilla on nimike, joiden avulla hinnat ryhmitellään SAP järjestelmässä. Hallintolisällä ja tuntiveloituksella on eri nimike. Alihankkijoilla on oma nimike ja muilla laskuttavilla henkilöillä (toimihenkilöt) nimikkeet jakautuvat tehtävän työn mukaan. Nimike välitetään liittymällä SAP -järjestelmään.	Laskuttava IT	Laskutus	LTV012	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Konfiguraatio; Kustomointi
TJV071	Projektin laskutustiedoissa on tilausnumero	IT:n, IT: kanssa vastaavalla tavalla laskuttavien ja alihankkijoiden henkilötyönlaskutus. Projektin jokaiselle maksajalle eli kustannuspaikalle on voitava merkittä tilausnumero. Tilausnumero kuvaa onko työ kuluva (2-alkuinen tilausnumero) tai aktivoitavaa (8-alkuinen tilausnumero). Tilausnumero on 8 merkinen ja se voi sisältää numeroita tai kirjaimia. Ensimmäinen merkki on aina 2 tai 8. Tieto on aikasidonnainen ja se historoidaan.	Laskuttava IT	Laskutus	LTV013	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Konfiguraatio, Kustomointi
TJV072	Aktivoivat tilausnumerot	IT:n, IT: kanssa vastaavalla tavalla laskuttavien ja alihankkijoiden henkilötyönlaskutus. Aktivoitavista tilausnumeroista on pystyttävä erottamaan oman työn aktivoitavat tilausnumero ja muiden yritysten aktivoitavan työn tilausnumerot.	Laskuttava IT	Laskutus	LTV013	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Konfiguraatio, Kustomointi
TJV073	Alihankkijoiden laskujen tilioinnissa käytetty tilinumero määräytyy projektin tietojen mukaan	Alihankkijoiden laskujen tilioinnissa käytetty tilinumero määräytyy projektin laskutuksessa käytetyn palvelutuotteen ja sisäisen laskennan tilinumeron määrittämisen mukaan.	Laskuttava IT	Laskutus	LTV013	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Konfiguraatio, Kustomointi
TJV074	Projektien sisäisen laskennan veloitukset kohdistuvat määrättyille kulutileille SAP -järjestelmässä ja laskutuksessa	Projekteille määritetään kulutili, jonne sen sisäiseen laskentaan menevät veloitukset kohdistuvat. Projektin tyyppi (kunnossapito, tuotannon tuki, pienkehitys ja kehitys) määrittää myös kulutilin jonne veloitukset kohdistuvat SAP -järjestelmässä. Tieto on aikasidonnainen ja se historoidaan.	Laskuttava IT	Laskutus	LTV013	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Konfiguraatio, Kustomointi
TJV075	Projektien laskutukseen menevät toteutumat kohdistuvat määrättyille palvelutuotteille SAP-järjestelmässä	Projekteille määritetään palvelutuote, jonne sen laskutukseen menevät veloitukset kohdistuvat SAP -järjestelmässä. Projektin tyyppi (kunnossapito ja tuotannon tuki, pienkehitys tai kehitys) määrittää myös palvelutuotteen jolle veloitukset kohdistuvat SAP -järjestelmässä. Eri laskuttavilla resurssiryhmillä on eri tilinro:t ja palvelutuotteet. Tieto on aikasidonnainen ja se historoidaan.	Laskuttava IT	Laskutus	LTV013	1- Välttämätön	<b>Hyväksytty</b>	Erä1	Konfiguraatio, Kustomointi

TJV076	Projektin maksava kustannuspaikka voi muuttua	Projektin kustannukset kohdistetaan laskutuksessa ja sisäisessä laskennassa kustannuspaikalle. Kustannuspaikka voi muuttua projektin elinkaaren aikana. Tieto on aikasidonnainen ja se historoidaan.	Laskuttava IT	Laskutus	LTV014	1-Välttämätön	Hyväksytty	Erä1	Konfiguraatio, Kustomointi
TJV077	Projektilla useampia kustannuspaikkoja maksajiksi projektitasolla	Projektin toteumat on voitava jakaa laskutukseen prosenttiperusteisesti useamman kustannuspaikan kesken. Tieto on aikasidonnainen ja se historoidaan.	Laskuttava IT	Laskutus	LTV014	1-Välttämätön	Hyväksytty	Erä1	Konfiguraatio, Kustomointi
TJV078	Projektille kustannuspaikka tehtävätason maksajaksi	Projektin yhden tehtävän toteumat on voitava veloittaa eri maksajalta kuin projektien muiden tehtävät. Tieto on aikasidonnainen ja se historoidaan.	Laskuttava IT	Laskutus	LTV014	1-Välttämätön	Hyväksytty	Erä1	Konfiguraatio, Kustomointi
TJV079	Resurssien kustannuspaikka tulonsaajana voi muuttua	Jokaisella resurssilla on kustannuspaikka tieto joka on tulonsaajana laskutuksen ja sisäisten veloitusten tapahtumissa. Tieto on aikasidonnainen ja se historoidaan.	Laskuttava IT	Laskutus	LTV014	1-Välttämätön	Hyväksytty	Erä1	Konfiguraatio, Kustomointi
TJV080	Resurssin lisätieto: Konsultin laskun tarkastaja	Resurssille lisätietokenttä johon valitaan resurssilistasta vastuuhenkilö, joka tarkastaa konsulttiyrityksen laskun. Ei historoida.	Laskuttava IT	Laskutus	LTV014	1-Välttämätön	Hyväksytty	Erä1	Konfiguraatio, Kustomointi
TJV081	Resurssin lisätieto: Konsultin laskun hyväksyjä	Resurssille lisätietokenttä johon valitaan resurssilistasta vastuuhenkilö, joka hyväksyy konsulttiyrityksen laskun. Ei historoida.	Laskuttava IT	Laskutus	LTV014	1-Välttämätön	Hyväksytty	Erä1	Konfiguraatio, Kustomointi
TJV082	Resurssin lisätieto: Konsultin sopimusvastaava	Resurssille lisätietokenttä johon valitaan resurssilistasta vastuuhenkilö, joka vastaa konsulttisolupimuksesta. Ei historoida.	Laskuttava IT	Laskutus	LTV014	1-Välttämätön	Hyväksytty	Erä1	Konfiguraatio, Kustomointi
TJV083	P.palveluiden laskutus kappalemääräistä	P.palveluiden laskutus osuuspankeille toteutetaan kappalemääräisenä.	P.Palvelut	Laskutus	LTV015	1-Välttämätön	Hyväksytty	Erä1	Konfiguraatio, Kustomointi
TJV084	P.palveluiden laskutustiedot	P.palveluiden laskutuksessa tulonsaaja riippuu laskutuksessa käytettävästä nimikkeestä.	P.Palvelut	Laskutus	LTV015	1-Välttämätön	Hyväksytty	Erä1	Konfiguraatio, Kustomointi
TJV085	P.palveluiden laskutustiedot	P.palveluiden laskutuksessa käytetään asiakaskohtaisia hintoja.	P.Palvelut	Laskutus	LTV015	1-Välttämätön	Hyväksytty	Erä1	Konfiguraatio, Kustomointi

Liite 6. Dokumentoitujen vaatimusten yhteenveto

Nykytila	Nykytilan /toiminnallisen vaatimuksen kuvaus	Toiminnallisen vaatimuksen ID	Toteutuksen kuvaus
Vuosisuunnittelu ja ennustaminen hanketasolla	Vuosisuunnittelu ja ennustaminen hanketasolla pitää sisältää hankkeiden merkitsemisen vuosisuunnitteluun kuuluviksi tai ehdolla seuraavaan vuosisuunnitteluun, vuosienuste budjettien luomisen vuosisuunnitteluun kuuluville hankkeille sekä vuosienuste budjettien tarkastukset kvartaaleittain. Lisäksi vuosisuunnittelu ja ennustaminen hanketasolla sisältää vuosisuunnitteluun kuuluvien hankkeiden budjettien siirron suunnittelun työkaluun.	TJV001 - TJV005	Konfiguraatio
Hankkeiden ja projektien kustannusarvion automatisoitu luominen	Projektin kustannusarvion saa luotua automatisoidusti työsuunnitelmasta tai varauksista. Hankkeen kustannusarvion saa luotua automatisoidusti niin, että menneen ajan (valittuun kk asti) kustannukset lasketaan hankkeeseen kuuluvien projektien toteumista ja tulevat kustannukset hankkeen edellisestä kustannusarviosta. Lisäksi kuluvan vuoden summat voi halutessaan säilyttää samana.	TJV006, TJV044	Konfiguraatio & kustomointi
Hankkeen ennusteet eivät ole yhtä kuin siihen kuuluvien projektien ennusteet	Hankkeella on omat ennusteet ja projekteilla omat ennusteet. Hankkeenäkymässä ennusteiden summautuminen on estetty ns. negatiivisilla projekteilla, niin että näkyvässä hankkeen ennusteeseen ei summata siihen kuuluvien projektien ennusteita.	TJV007	Kustomointi
Toteutuneet kustannukset tuodaan SAP -järjestelmästä	Ostettujen palveluiden kustannukset tuodaan SAP -järjestelmästä kuukausittain poimimalla valittujen kulutilien tapahtumat. Toteutuneet kustannukset kohdistetaan projekteille kustannuksen tilausnumeron ja kulutilin perusteella. Mikäli projektia ei voida päätellä, tehdään kohdistus manuaalisesti. Mahdolliset korjaukset tehdään negatiivisilla viennillä.	TJV008	Konfiguraatio & kustomointi
Hankkeen lisätiedot ja luokittelutekijät	Hankkeen lisätiedot ja luokittelutekijät ovat: päätöspiste, tilanne (status) tieto, strategian mukaisuus, kehityshankkeen luokittelu, kehitysprosessin vaihe, kehityssuunnitelma, vaikutusalue, projektisalkku, mittarihanke, prioriteetti, kehityssalkku, kehityssalkun tarkenne, sovellussalkku, tila, vastuullinen, liiketoimintavastaava, hankkeen johtoryhmän puheenjohtaja, hankkeen johtoryhmä, hankkeen materiaalin tallennuspaikka, hankkeen tavoite, huomionarvoista, liiketoiminnan sisäinen työ sekä hankkeiden väliset riippuvuudet. Lisäksi hankkeelle tallennetaan hankekortille kerätyt tiedot erillisiin hankekortin tiedot objektin kenttiin.	TJV015 - TJV038	Konfiguraatio
Hankkeella useita kustannuspaikkoja	Hankkeen toteumat ja ennusteet jaetaan tarvittaessa % perusteisesti useammalle kustannuspaikalle. Osuudet ovat aikaan sidonnaisia.	TJV013	Konfiguraatio
Hankkeen budjetin lisätiedot	Hankkeen budjetissa on kaksi budjetin tyyppi lisätietoa; budjetin päätöspiste sekä hankkeen budjetin muu tyyppi. Lisäksi listausta varten on tieto hankkeen budjetin tyyppi, jonka valintalistan arvot päivittyvät automaattisesti sen mukaan kumman tyyppitiedon käyttäjä on valinnut; budjetin päätöspisteen vai hankkeen budjetin muu tyyppiin.	TJV009 - TJV011	Kustomointi

Organisaatioperusteiset oikeudet hanketietoihin	Eri yritysten hankevastaavilla on oikeudet vain oman alueensa hankkeisiin ja näkymä niiden tietoihin. He voivat myös luoda omista hankkeistaan portfolioita sekä antaa muille käyttäjille niihin oikeuksia.	TJV012	Kustomointi
Hanke ja projekti kuuluvat salkkuun	Hanke ja projekti kuuluvat niiden kustannuspaikan mukaisesti palvelupäällikön salkkuun. Kustannuspaikat on jaettu salkkuihin ja hanke sekä projekti saa automaattisesti salkun päämaksajan (kustannuspaikan) mukaan.	TJV013, TJV046	Kustomointi
Projektin lisätiedot ja luokittelutekijät	Projektin lisätiedot ja luokittelutekijät ovat: tilanne tiedon tallennus, elinkaaren VSOP vaihe, kehitysprosessin vaihe, tehtävän luokittelutekijä tehtävän luokittelu sekä kunnossapitoprojektin budjetin lisätieto tyyppi.	TJV039 - TJV043	Konfiguraatio
Toteutuneiden kustannusten syöttö projektille	Projektille syötetään manuaalisesti sellaisia kustannuksia toteumaksi, jotka eivät tule SAP -järjestelmästä (kohta: toteutuneet kustannukset SAP – järjestelmästä) tai eivät ole tuntikirjauksiin perustuvia. Kustannukset kohdistetaan tehtävälle ja tietyille kustannuslajeille, päivälle jonne kustannus kohdistuu raportoinnissa. Mahdolliset korjaukset tehdään negatiivisilla vienneillä.	TJV045	Konfiguraatio
Resurssintikokonaisuus	Resurssintikokonaisuuden toiminnallisuuksiin kuuluu eri kokonaisuuksia liittyen resurssintikokonaisten tunnistamiseen, niiden linkitykseen projekteihin ja hankkeisiin sekä resurssintikokonaisten vastaavien käyttöoikeuksiin resurssintikokonaisuuden varattujen henkilöiden kiinnitykseen toteuttaville projekteille.	TJV047 - TJV052	Konfiguraatio & kustomointi
Kaksivaiheinen resurssointi	Kaksivaiheiseen resurssointiin kuuluu rooli kapasiteetin hyväksymispyyntö sekä pyynnön lisätiedot. Lisäksi resurssipyynnön (Requisitionin) lisätiedot sekä esto, että resurssipyyntöjä voi luoda vain resurssintikokonaisuudesta.	TJV053- TJV057	Konfiguraatio & kustomointi
Laskutus ja henkilöiden organisaatiotiedon tallennus	Laskutus pitää sisällään kaikki henkilötyön laskutuksen toiminnallisuudet jotka näkyvät ohjelmistossa henkilöiden (resurssien) tiedoissa ja niiden historoinnissa, projektien tiedoissa ja niiden historoinnissa, tuntityön hinnoissa, niiden muutoksissa, laskutettavan työn hinnoittelussa sekä laskutuksen takautuvien korjausten mahdollistamisena. Kuvaukseen on yhdistetty myös resurssienhallintaan kuuluva henkilöiden organisaatiotiedon historointi, sillä sen toteutus on osa henkilöiden laskutustietojen historoinnin toteutusta.	TJV058 - TJV085	Konfiguraatio & kustomointi