

Quality Function Deploymentin käyttö vakuutusjärjestelmän uusimisen vaihtoehtojen vertailussa

Kalle Aalto



Tekijä(t) Kalle Aalto	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Opinnäytetyön otsikko Quality Function Deploymentin käyttö vakuutusjärjestelmän uusimisen vaihtoehtojen vertailussa	Sivu- ja liitesivumäärä 32 + 2
Opinnäytetyön otsikko englanniksi Use of Quality Function Deployment in comparing insurance system renewal possibilities	
<p>Quality Function Deployment (QFD) on japanilaisten kehittämä laatutyön menetelmä, jonka avulla tuote voidaan alusta asti suunnitella täyttämään mahdollisimman hyvin asiakastarpeet. Perinteisessä valmistavassa teollisuudessa menetelmää käytetään useimmiten nelivaiheisena ja vaiheina ovat tuotteen suunnittelu, osien suunnittelu, prosessien suunnittelu ja valmistuksen suunnittelu. Menetelmän tärkein ja eniten käytetty osa on tuotteen suunnitteluvaiheessa käytetty laaduntalomatriisi, jolla asiakastarpeet muunnetaan tuotteen ominaisuuksiksi. Laaduntalo sisältää myös kilpailija-analyysia ja sen lopputuloksena tuotteen ominaisuuksille saadaan tärkeysjärjestys.</p> <p>Quality Function Deployment on sovellettavissa ohjelmistotuotantoon, mutta pitää sitä muokata vastaamaan ohjelmistotuotannon fyysisten tuotteiden valmistuksesta poikkeavia toimintatapoja. Tähän tarkoitukseen on kehitetty monia eri malleja, jotka poikkeavat toisistaan siinä miten laajasti ne sisältävät ohjelmistotuotannon vaiheita ja minkälaisia kaavioita ja matriiseja niissä käytetään. Tässä työssä käsitellyistä menetelmistä Blitz QFD ja Method\1 keskittyvät ainoastaan asiakastarpeiden johtamiseen ohjelmiston tärkeimmiksi toiminnoiksi. Software QFD sisältää mallina koko ohjelmistoprojektin vaiheet, mutta on hyvin teoreettinen ja osin tekemisen vaiheistukseltaan normaaleista ohjelmistotuotantomenetelmistä poikkeava. Software Quality Deployment sisältää asiakastarpeiden johtamisen ohjelmiston ominaisuuksiksi ja edelleen ohjelmiston rakenteen tarkan suunnittelun.</p> <p>Keskinäinen Vakuutusyhtiö Fenniassa tehtiin 2015 esiselvitysprojekti yritysvakuuttamisen järjestelmien uusimisesta. Käyttämällä esiselvitysprojektissa kerättyjä vaatimuksia syötteenä johdettiin niistä Blitz QFD-mallia mukaillen järjestelmän tärkeimmät toiminnot. Tärkeysjärjestystä voidaan käyttää hyväksi vaihtoehtojen vertailussa. Fenniassa systeemyömallit toimivat hyvin, mutta QFD voisi olla hyödyllinen vaatimusmäärittelysten priorisoinnissa.</p>	
Asiasanat laatutyö, tietojärjestelmät, määrittely	

Author(s) Kalle Aalto	
Degree programme Business Information Technology	
Report/thesis title Use of Quality Function Deployment in comparing insurance system renewal possibilities	Number of pages and appendix pages 32 + 2
<p>Quality Function Deployment (QFD) is a quality development method originally created by the Japanese. It can be used from early design phases to make a product to fulfil customer needs as well as possible. In traditional manufacturing industry it is used mostly in four phases, which are product design, parts deployment, process planning and production planning. The most important and most used part of this method is a House of Quality matrix, which is used in product design phase. It is used for translating customer needs into product's technical measures. House of quality includes also competitive analysis and it creates priorities for technical measures as a final outcome.</p> <p>Quality Function Deployment can be used in software development, but it needs to be tuned to make it better fit for software development methods that differ from methods in physical product manufacturing. For this purpose multiple models have been created and they differ from each other in how widely software development steps are included and what kind of matrix or charts are used. In models described in this study Blitz QFD and Method\1 concentrate only on translating customer needs into most important functions of a software. Software QFD includes all phases of a software development, but is highly theoretical and its phases order partly differ from normal software developments models. Software Quality Deployment includes translation of customer needs into software functions and furthermore into detail design of software.</p> <p>In Fennia Mutual Insurance Company a pre-study project about renewing company insurance systems was made during 2015. By using requirements gathered in that project most important functions of the system were led by using applied Blitz QFD. The importance of functions can be used for the comparison of different renewal possibilities. In Fennia system development models are working well, but QFD could be beneficial in the prioritization of requirements.</p>	
Keywords quality tools, information systems, requirements	

Sisällys

1	Johdanto	1
1.1	Tutkimuksen tausta ja tavoitteet	1
2	Quality Function Deployment	3
2.1	Nelivaiheinen menetelmä	4
2.2	Laaduntalo	5
2.2.1	Asiakastarpeet	7
2.2.2	Suunnittelumatriisi	9
2.2.3	Ominaisuudet	10
2.2.4	Riippuvuudet Mitä-Miten	10
2.2.5	Riippuvuudet Miten-Miten	10
2.2.6	Ominaisuusmatriisi	11
3	QFD ohjelmistotuotannossa	12
3.1	Software Quality Deployment	13
3.2	Method/1	15
3.3	Software QFD	17
3.4	Blitz QFD	18
3.5	Kokemuksia QFD-menetelmän käytöstä ohjelmistokehityksessä	21
3.6	Menetelmien vertailu	22
4	Fennian yritysvaruutusjärjestelmien uudistaminen	24
4.1	Esiselvitysprojekti	25
4.2	Liiketoimintavaatimusten analysointi	26
4.3	Liiketoimintavaatimusten ryhmittely	27
4.4	Tarpeiden keskinäinen priorisointi	28
4.5	Järjestelmän tärkeimpien osien johtaminen	28
5	Pohdinta	30
5.1	Johtopäätökset	30
5.2	Jatkotutkimusmahdollisuudet	30
5.3	Opinnäytetyön prosessi ja oma oppiminen	31
	Lähteet	32
	Liitteet	33
	Liite 1. Liiketoimintavaatimusten korrelaatio järjestelmän osiin	33
	Liite 2. Tarpeista ja vaatimuksista johdettu osien tärkeys	34

1 Johdanto

Vaikka tietojärjestelmiä on käytetty ja kehitetty jo vuosikymmeniä, uutisoidaan silti säännöllisesti, kuinka isot tietojärjestelmäprojektit epäonnistuvat tavoitteissaan. Uutisiin päätyvät epäonnistumiset ovat vain jäävuoren huippu ja luultavasti lähes jokaisella organisaatiolla on kokemuksia epäonnistuneista tietojärjestelmäprojekteista.

Laadun parantaminen ja siihen liittyvät menetelmät lähtivät kehittymään Japanissa Toisen maailmansodan jälkeen, kun niukoilla resursseilla yritettiin saada aikaiseksi kilpailukykyisiä tuotteita. Erityisesti japanilainen autoteollisuus on tullut kuuluisaksi luotettavista ja vähävikaisista tuotteistaan. Pitkälti tämän seurauksena nämä menetelmät ovat erityisesti teollisuudessa, mutta myös muilla toimialoilla otettu käyttöön ympäri maailman.

Ohjelmistokehityksessä ketterien menetelmien käyttö on yleistynyt viime aikoina nopeasti. Nämä menetelmät pohjautuvat monelta osin laatutyön menetelmien käyttöön. Osa menetelmistä on lainattu suoraan laatutyömenetelmistä esimerkkinä Kanban-korttien käyttö. Perusajatus vain tarpeellisen toteuttamisesta on sama kuin laatutyössä, jossa puhutaan hukkan poistamisesta. Myös ketterä on käsitteenä ollut kauan käytössä yhtenä ajatusmallina laatutyön soveltamisessa.

Suurin osa laatutyökaluista on tarkoitettu havaittujen ongelmien tai virheiden korjaamiseen ja niiden uudelleenilmenemisen estämiseen. Quality Function Deployment on Japanista lähtöisin oleva laatutyömenetelmä, jolla pyritään jo suunnitteluvaiheen aikana kehittämään tuote tai palvelu vastaamaan mahdollisimman hyvin asiakkaiden tarpeita. Juuri asiakastarpeiden täyttäminen on yksi niistä asioista, joissa monet tietojärjestelmäprojektit epäonnistuvat, joten QFD voisi menetelmänä auttaa tietojärjestelmäprojekteja onnistumaan useammin.

1.1 Tutkimuksen tausta ja tavoitteet

Keskinäinen vakuutusyhtiö Fennia on uusinut viimeisen vuosikymmenen aikana vanhoja ydintietojärjestelmiään. Tähän mennessä on korvattu asiakkaiden hoitoon, kotitalous- ja ajoneuvovakuuttamiseen sekä korvauskäsittelyyn liittyvät järjestelmät. Vanhat järjestelmät ovat käytössä vielä yritysten vakuuttamisessa ja niiden uusimisen suunnittelu on tullut ajankohtaiseksi.

Fenniassa tehtiin 2015 koko vuoden kestänyt esiselvitysprojekti yritysten vakuuttamisen järjestelmien uusimisvaihtoehtoista. Projektissa kartoitettiin erilaisia vaihtoehtoja ja käytiin

läpi liiketoimintaprosesseja, nykyisten järjestelmien toimintoja että uusia tarpeita järjestelmälle. Projektissa käytiin läpi erilaisia vaihtoehtoja järjestelmän uusimiseksi ja tarkemmin analysoitiin kahta varteenotettavinta vaihtoehtoa.

Ennen Fenniaan siirtymistä työskentelin melkein kymmenen vuotta autoteollisuuden alihankkijan palveluksessa. Autoteollisuudessa laatutyö on systemaattisessa käytössä joka puolella ja mielenkiintoni kohteena on miten hyvin alun perin fyysisten tuotteiden suunnitteluun kehitettyjä menetelmiä voidaan hyödyntää tietojärjestelmien kehittämisessä.

Tutkimuksen tavoitteena on perehtyä Quality Function Deployment (QFD) menetelmän käyttöön ja sen soveltamiseen ohjelmistojen suunnittelussa. Toisena tavoitteena on tutkia, voisiko QFD-menetelmää hyödyntää Fennian vakuutusjärjestelmien uudistamisen vaihtoehtojen vertailussa.

Tutkimus pyrkii vastaamaan seuraaviin kysymyksiin:

- Mikä on QFD?
- Miten QFD-menetelmää on sovellettu ohjelmistokehityksessä?
- Soveltuisiko QFD-menetelmä tietojärjestelmä vaihtoehtojen vertailuun?
- Voidaanko vaatimusmäärittämisestä menetelmän avulla johtaa järjestelmän tärkeimmät tietotekniset ominaisuudet?

Tutkimuksessa rajaudutaan QFD-menetelmän ensimmäiseen vaiheeseen ja tarkoituksena ei ole tutkia sen myöhempiä vaiheita. Tutkimus rajautuu tarkastelemaan pelkästään vakuutusjärjestelmän vaatimuksia.

2 Quality Function Deployment

Tämä luku käy läpi Quality Function Deployment-menetelmän historian ja käyttötarkoituksen. Luvussa esitellään myös menetelmän soveltamista teollisuudessa sekä menetelmän käytön perusteet.

Uutta liiketoimintaa syntyy yleensä siksi, että tyytymättömien asiakkaiden tarpeet huomataan. Uusi tarjoama näiden tarpeiden tyydyttämiseksi voi olla innovatiivinen ja ainutlaatuinen palvelu tai tuote. Riippumatta siitä kuinka hyvin yritys täyttää asiakkaidensa alkupe-
räiset tarpeet, sen pitää olla jatkuvasti herkkä vastaamaan asiakkaiden uusiin tarpeisiin. (Koski 2003, 6.)

Asiakastytyväisyyttä voidaan tarkkailla kyselyillä ja asiakasvalitusten ja -palautteen perusteella. Usein nämä menetelmät eivät paljasta asiakkaiden todellisia tarpeita, koska ne on tehty myynnin ja markkinoinnin tarpeisiin. Asiakkaan todellisten tarpeiden ymmärtäminen vaatii suuria panostuksia, jotka on tarkkaan suunniteltava. (Koski 2003, 6.)

Quality Function Deployment (QFD) on menetelmä, joka pyrkii ottamaan huomioon laadun eri ulottuvuudet jo tuotekehityksen aikana ja sisällyttämään laadun tuotteeseen alusta asti (Lillrank, 1990). Perinteiset laatujärjestelmät ja -menetelmät pyrkivät minimoimaan negatiivisen laadun tuotteessa, kun taas QFD lisää tuotteen arvoa maksimoimalla positiivisen laadun määrän. Menetelmässä keskitytään asiakkaan tarpeiden ymmärtämiseen ja niiden muuntamiseen tuotteen halutuiksi ominaisuuksiksi. QFD ei ole alun perin vaatimusmäärittelyn menetelmä vaan systemaattinen metodi, jolla asiakastarpeet voidaan muuntaa tarkoituksiksi tuoteominaisuuksiksi. (Kivinen 2008, 34.)

Shigeru Mizuno ja Yoji Akao esittelivät ensimmäiset ajatukset QFD-menetelmästä Japanissa 1960-luvulla. Ensimmäinen menetelmän soveltaja oli Bridgestone Tire Company, jossa sitä käytettiin 1966 Kiyotaka Oshiumin toimesta kalanruotokaavion muodossa. (Alrabghi 2013, 7.)

QFD ei herättänyt aluksi juurikaan kiinnostusta, kunnes 1972 sitä käytettiin öljytankkerin suunnittelussa Mitsubishi Heavy Industries Ltd:n Koben telakalla. Kalanruotokaavio osoitautui epäkäytännölliseksi, koska sillä ei voi esittää useampaa syy-yhteyttä. Kalanruotokaavio vaihdettiin laatutauluiksi, joissa rivit olivat asiakastytyväisyyttä lisääviä tekijöitä ja kolumnit mitattavia syitä. (Alrabghi 2013, 7.)

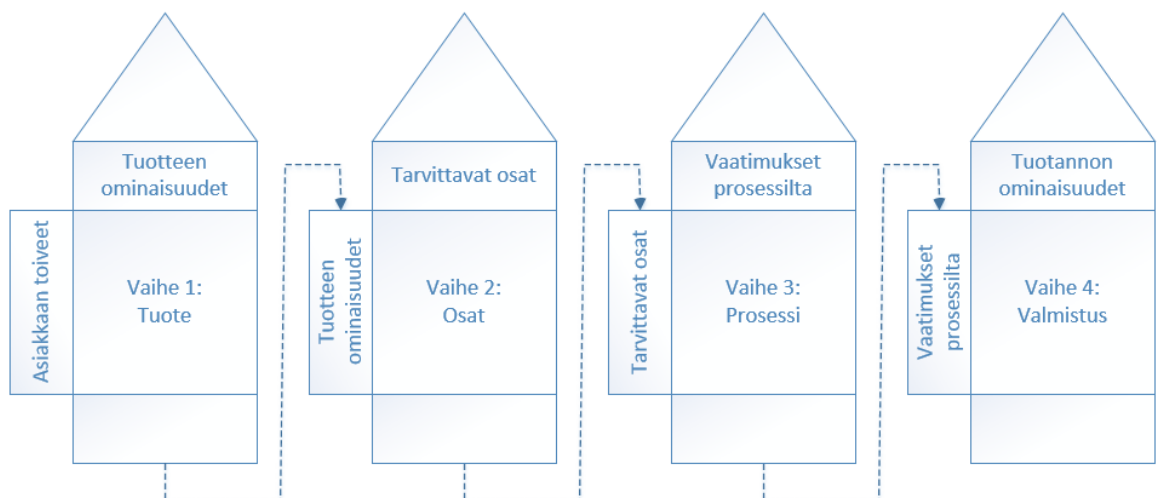
Toyota Autobody otti käyttöön QFD:n vuonna 1978 ja samana vuonna julkaistiin myös ensimmäinen kirja QFD:sta Mizunon ja Akaon kirjoittamana. Se käännettiin englanniksi vuonna 1994. Ensimmäinen vuosittainen QFD-seminaari pidettiin Japanissa 1983. Samana vuonna QFD esiteltiin Yhdysvalloissa sekä Euroopassa. American Society of Quality Control julkaisi artikkelin Akaon työstä 1983. (Alrabghi 2013, 8.)

Yleisimmät QFD:n käyttötavat ovat matriisien matriisi ja nelivaiheinen analyysi. Matriisien matriisi koostuu noin 30 taulusta, matriisista tai muusta kaaviosta. Tämän vuoksi sen käyttöä pidetään vaativana ja suuritöisenä. Nelivaiheisen mallin kehittivät Hauser ja Clausing ja esimerkiksi American Supplier Institute on ottanut sen käyttöön. (Alrabghi 2013, 22.)

2.1 Nelivaiheinen menetelmä

QFD on menetelmänä luonnostaan joustava ja sitä käytetään monin eri tavoin. Tyypillinen erityisesti valmistavassa teollisuudessa käytettävä QFD sisältää neljä vaihetta

1. tuotteen suunnittelu
2. osien suunnittelu
3. prosessin suunnittelu
4. valmistuksen suunnittelu.



Kuvio 1. QFD:n nelivaiheinen menetelmä (mukailtu Alrabghi 2013, 23)

Kuviossa 1 esitetyn nelivaiheisen mallin jokaisessa vaiheessa käytetään matriisia muuntamaan asiakasvaativukset yrityksen määräyksiksi alkuvaiheen suunnittelusta tuotannon suunnitteluun asti. Jokainen vaihe tai matriisi esittää tarkemman näkökulman tuotteen

vaatimuksista. Jokainen matriisi koostuu pystysuuntaisista "Miten"-kolumneista ja vaakasuuntaisista "Mitä"-riveistä. "Mitä"-rivit ovat asiakasvaatimuksia ja "Miten"-kolumnit keinoja saavuttaa asiakasvaatimukset tai teknisiä ominaisuuksia. Jokaisessa vaiheessa vain tärkeimmät "Miten"-kolumnit siirretään seuraavan vaiheen uusiksi "Miksi"-riveiksi. Ensimmäistä vaihetta kutsutaan yleisesti laaduntaloksi, mutta laaduntalo-matriisi voidaan tehdä jokaisessa vaiheessa. Iso osa QFD:tä käyttävistä organisaatioista keskittyy ainoastaan ensimmäiseen vaiheeseen vaikka kaikissa vaiheissa käytetään melkein samoja rakenteita ja analyysimenetelmiä. (Alrabghi 2013, 23.)

Ensimmäinen vaiheen tuotesuunnittelu, jota myös laaduntaloksi kutsutaan, tehdään markkinointiosaston johtamana. Tässä vaiheessa asiakastarpeet muunnetaan tuotteen suunnittelusta riippumattomiksi mitattaviksi ominaisuuksiksi. Lisäksi voidaan huomioida ja dokumentoida myös takuuasiat, tuotteen mitat, kilpailevien tuotteiden mitat ja organisaation tekninen kyvykkyys saavuttaa asiakastarpeet. Kerättävän tiedon laatu ja oikeellisuus tässä vaiheessa on elintärkeää koko prosessin onnistumiselle. (Alrabghi 2013, 24.)

Toiseen vaiheen osien suunnittelu on tuotekehitysvetoista. Tuotekonseptit suunnitellaan niin, että ne täyttävät vaatimukset tärkeimmille ominaisuuksille. Osat, jotka ovat tärkeimpiä asiakastarpeiden saavuttamiseksi, asetetaan tärkeysjärjestykseen ja annetaan syötteeksi seuraavalle vaiheelle. (Alrabghi 2013, 24.)

Kolmannen vaiheen prosessisuunnittelu tehdään prosessikehityksen johdolla. Kriittiset ominaisuudet ja parametrit muunnetaan yksityiskohtaiseksi valmistusprosesseiksi, prosessin kontrolleiksi ja kehityskohteiksi. Tärkeimmät prosessin kontrollit ja kehityskohteet annetaan syötteeksi seuraavalle vaiheelle. (Alrabghi 2013, 24.)

Neljännessä vaiheessa tehdään tuotannon kontrollit ja ohjeistus tuotannon valvomiseksi sekä huoltosuunnitelmat ja työntekijöiden koulutussuunnitelmat. Erityisesti tässä vaiheessa on tärkeää tehdä laadunvarmistussuunnitelmat ja toimintamallit vikatilanteisiin. Vaihe tehdään laadunvarmistuksen johtamana yhteistyössä valmistuksen kanssa. (Alrabghi 2013, 25.)

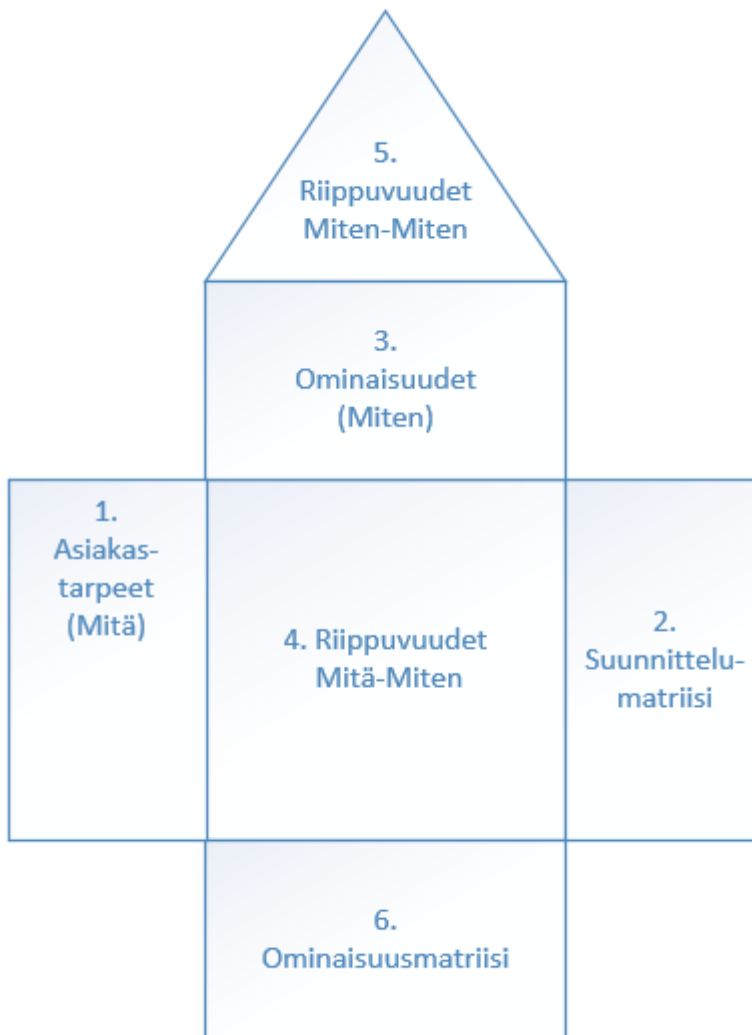
2.2 Laaduntalo

Laaduntalo nimitys tulee matriisin ulkonäöstä, joka muistuttaa harjakattoista taloa. Laaduntalo on QFD:n perussuunnittelutyökalu ja se on ensimmäinen ja tärkein QFD-matriiseista. Se suhteuttaa asiakkaan tarpeet ylätasoon mitattaviksi teknisen suunnittelun vaatimuksiksi käyttämällä suunnittelumatriisia. Laaduntalon lopputuloksena ovat tärkeim-

mät tekniset ominaisuudet ja niiden suhde asiakastarpeisiin sekä kilpailukykyanalyysi. Tämä hyödyttää suunnittelijoita, koska tekniset vaatimukset voidaan johtaa takaisin asiakastarpeisiin. (Alrabghi 2013, 25.)

Laaduntalo koostuu kuudesta pääosasta

1. asiakastarpeet (Mitä)
2. suunnittelumatriisi
3. ominaisuudet (Miten)
4. riippuvuudet Mitä-Miten
5. riippuvuudet Miten-Miten
6. ominaisuusmatriisi.



Kuvio 2. Laaduntalon pääosat (mukailtu Alrabghi 2013, 25)

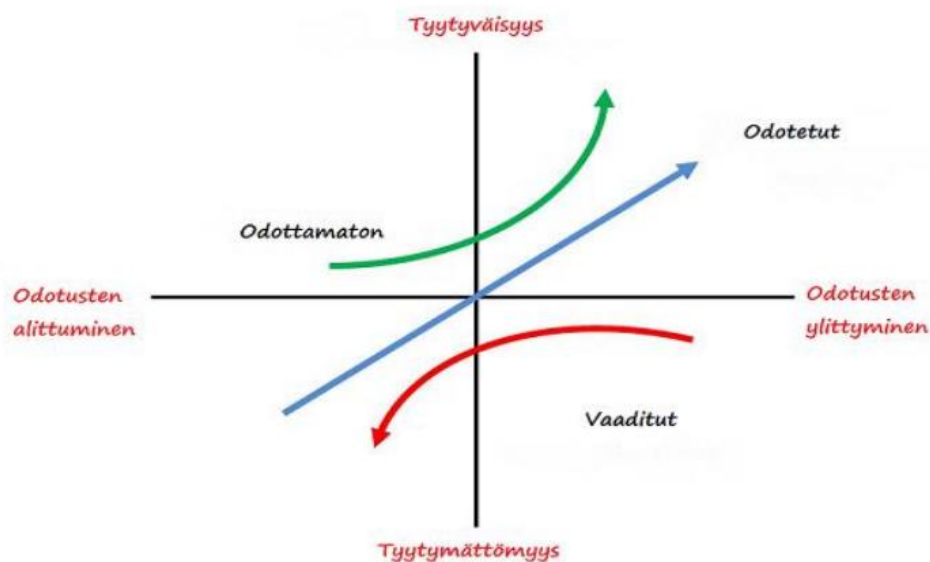
2.2.1 Asiakastarpeet

Asiakastarpeiden määrittely on vaativa tehtävä, joka koostuu useasta osasta. Ensimmäiseksi tuotteen asiakkaat täytyy tunnistaa. Asiakastarpeet pitää kerätä ja selvittää niiden tärkeys suhteessa toisiinsa. On tärkeää huomata, jos asiakastarpeet muuttuvat ja tarpeiden kehittymistä ajan mittaan kannattaa seurata. (Alrabghi 2013, 28.)

Asiakkaita on yleensä kolmen tyyppisiä: sisäisiä asiakkaita, väliasiakkaita ja loppuasiakkaita. Sisäisiä asiakkaita ovat omistajat, johto ja työntekijät, kun taas esimerkiksi jälleenmyyjät ovat väliasiakkaita. Loppuasiakkaita ovat esimerkiksi palvelun vastaanottajat tai ostajat. (Alrabghi 2013, 28.)

Asiakkaiden tarpeita voidaan kartoittaa monella eri menetelmällä. Asiakkaat kuvailevat tarpeitaan usein paremminkin "kuinka" tarve täytetään, kuin "mitä" tarve on. Todellisen tarpeen selvittämiseksi on tällöin osattava jatkaa "miksi" kysymyksillä, kunnes todellinen tarve on ymmärretty. (Alrabghi 2013, 28.)

Noriaki Kano kehitti oppilaidensa kanssa mallin, joka suhteuttaa asiakastytyväisyyden siihen, miten hyvin tuotteen ominaisuudet täyttävät asiakastarpeet. Asiakastarpeet jaotellaan vaadittuihin, odotettuihin ja odottamattomiin. Kuviossa 3 on esitetty miten Kano-mallissa erilaisten asiakastarpeiden täyttäminen vaikuttaa asiakkaan tyytyväisyyteen. (Alrabghi 2013, 30.)



Kuvio 3. Kano-malli (mukailtu Alrabghi 2013, 30)

Odotetut tarpeet ovat niitä, joita asiakkaat osaavat vaatia ja niiden määrä parantaa asiakastytyvyyttä. Odotettuja tarpeita on helppo mitata ja yleensä kaikissa keskenään kilpailuvissa tuotteissa löytyy samoja odotettuja ominaisuuksia, joten niitä on helppo käyttää vertailussa. (Alrabghi 2013, 30.)

Vaaditut tarpeet ovat sellaisia, joiden ilman muuta oletetaan kuuluvan tuotteeseen. Ne eivät lisää asiakastytyvyyttä, mutta niiden puuttuminen tai toimimattomuus aiheuttaa tyytymättömyyttä. Normaalisti ne ovat näkymättömiä tarpeita, mutta niiden poisjättäminen tuo tarpeet näkyviin ja johtaa tyytymättömyyteen ja reklamaatioihin. Näitä tarpeita ei normaalisti oteta QFD-matriisiin mukaan, jos ei ole välttämätöntä keskittyä niihin. (Alrabghi 2013, 30.)

Odottamattomat tarpeet ovat uusia tai poikkeavia ominaisuuksia, jotka saavat asiakkaat innostumaan ja suuresti arvostamaan näitä ominaisuuksia. Odottamattomien ominaisuuksien sisällyttäminen tuotteeseen voi johtaa merkittävään kilpailuetuun ja suurempaan markkinaosuuteen. Myös odottamattomat tarpeet ovat normaalisti näkymättömiä, koska asiakkaat eivät osaa vaatia niitä. Kun ne sisällytetään tuotteeseen, ne tulevat näkyviksi ja saavat aikaan asiakastytyvyyden lisääntymistä. Näiden ominaisuuksien poisjättäminen ei kuitenkaan aiheuta tyytymättömyyttä, koska asiakkaat eivät osaa vaatia tai odottaa niitä. (Alrabghi 2013, 31.)

Kuviossa 3 suora linja esittää odotettuja tarpeita. Asiakkaiden tyytyväisyys kasvaa, kun odotetut tarpeet ylitetään ja huononee, jos ne alitetaan. QFD on vahvimillaan odotettujen tarpeiden parissa. Sillä saadaan strategiset edut näkyviksi. Vaatimukset kuitenkin muuttuvat ajan myötä. Odottamattomat ominaisuudet muuttuvat ensin odotetuiksi ja sitten vaadituiksi, kun asiakkaat alkavat olettaa näiden automaattisesti sisältyvän tuotteeseen. (Alrabghi 2013, 32.)

QFD:ssä käytetään yleensä vaadittuja ominaisuuksia. Odottamattomat ominaisuudet ovat uniikkeja ja asiakkaat eivät osaa niitä toivoa. Asiakkaiden toiveista voidaan löytää vihjeitä odottamattomista ominaisuuksista, mutta yleensä niiden löytämiseksi tarvitaan luovia ideoita, uusi lähestymistapoja tai teknologisia muutoksia (Koski 2003, 12). QFD ei menetelmänä tarjoa niiden löytämiseen juuri apua (Alrabghi 2013, 32).

Asiakasvaatimuksia on monenlaisia ja niiden tärkeys vaihtelee. Menestyvän tuotteen luomiseksi kaikki vaatimukset täytyy ottaa huomioon ja tasapainottaa keskenään. QFD:n seuraava vaihe määrittelee suhteellisen tärkeyden asiakasvaatimuksille. Suhteellinen tärkeys näyttää kuinka tärkeää vaatimuksen täyttäminen asiakastytyvyyden suhteen on.

Tuotteen kehityksessä keskitytään kaikista tärkeimpiin asiakasvaatimuksiin ja epäolennaiset vaatimukset jätetään huomiotta. Asiakkaita pyydetään arvioimaan vaatimusten tärkeys käyttämällä pisteytystä esimerkiksi yhdestä viiteen tai yhdestä kymmeneen. Pisteytyksiä käytetään myöhemmin suunnittelumatriisissa. (Alrabghi 2013, 33.)

2.2.2 Suunnittelumatriisi

Suunnittelumatriisi antaa vastauksia siihen miksi joidenkin ominaisuuksien kehittämiseen kannattaa keskittyä. Kun asiakastarpeet on kartoitettu, keskitytään suunnittelumatriisissa asiakastarpeiden mitattavuuteen. Suunnittelumatriisissa tuotetta verrataan kilpailijoiden tuotteisiin asiakasvaatimusten suhteen. (Alrabghi 2013, 34.)

Aluksi on löydettävä kilpailijoiden tuotteet, jotka ovat kehitettävän tuotteen kanssa samankaltaisia. Kilpailuedun saavuttamiseksi on elintärkeää selvittää omat heikkoudet ja vahvuudet suhteessa kilpailijoihin kaikkien asiakasvaatimusten suhteen. Tämä voidaan tehdä pyytämällä asiakkaita arvioimaan oman ja kilpailijatuotteiden suoriutumista asiakasvaatimuksista suhteessa toisiinsa. On tärkeää, että asiakkaat arvioivat vain sellaisia tuotteita, joista heillä on riittävää käyttökokemusta. Täysin uuden tuotteen kohdalla kilpailijavertailun tekeminen voi olla hankalaa. (Alrabghi 2013, 35.)

Kilpailija-arvioinnin pohjalta tiedetään oma suorituskyyky verrattuna kilpailijoihin jokaisen asiakasvaatimuksen kohdalla. Jokaiselle asiakasvaatimukselle annetaan tavoite samalla skaalalla kuin asiakkaat arvioivat tuotteita. Tavoitteen pitää olla realistinen ja ottaa huomioon käytettävissä olevat resurssit, teknologiat ja aikataulu sekä suunniteltu kustannustaso. Näiden tavoitteiden asetus määrittää minkälaisia toimenpiteitä tehdään, jotta asiakasvaatimukset täytetään paremmin. Toimenpiteitä ovat esimerkiksi (Alrabghi 2013, 35.)

- paranna QFD toteutuksen kautta
- säilytä nykyinen asema
- kopioi kilpailijaa
- vähennä.

Edellisten vaiheiden pohjalta voidaan määritellä myyntipisteitä. Myyntipiste kuvailee kykyä myydä tuotetta sen perusteella miten hyvin se täyttää kunkin asiakasvaatimuksen. Kun oma ja kilpailevat tuotteet ovat kaikki huonoja jonkin asiakasvaatimuksen suhteen, on syynä usein teknologinen pullonkaula, jonka ratkaisemisella voidaan tuotetta parantaa ja saavuttaa ainutlaatuinen myyntiväittämä. (Alrabghi 2013, 35.)

2.2.3 Ominaisuudet

Tämä osa Laaduntalosta keskittyy teknisiin ominaisuuksiin. Asiakastarpeet muunnetaan teknisiksi ominaisuuksiksi. Niitä kutsutaan termillä "Miten", koska ne ovat vastauksia siihen miten asiakkaan tarpeet voidaan tyydyttää. On tärkeää huomioida, että tekniset ominaisuudet eivät ole ratkaisuja. Jokaiseen asiakastarpeeseen pitää liittyä ainakin yksi tekninen ominaisuus. (Alrabghi 2013, 38.)

Jokaiselle ominaisuudelle pitää määrittää yksikkö. Yksikkö voi olla esimerkiksi fyysinen suure kuten pituus tai tilavuus. Yksikkö voi olla myös esimerkiksi prosessivaiheiden määrä tai laatupoikkeamien määrä miljoona tuotetta kohden. Jokaiselle yksikölle määritellään myös suunta, jota kohti mentäessä ominaisuus paranee. Mahdollisia suuntia on kolme: enemmän on parempi, vähemmän on parempi tai lähimpänä tavoitearvoa on parempi. (Alrabghi 2013, 38.)

2.2.4 Riippuvuudet Mitä-Miten

Matriisi, jossa tunnistetaan asiakastarpeiden ja ominaisuuksien väliset riippuvuudet on tärkeä vaihe, koska sen perusteella tehdään johtopäätökset siitä mihin asioihin tuotteen suunnittelussa keskitytään. (Alrabghi 2013, 38.)

Erilaisia riippuvuuden vahvuuksia on neljä: ei riippuvuutta, heikko/mahdollinen riippuvuus, keskinkertainen riippuvuus ja vahva riippuvuus. Riippuvuuksille annetaan painoarvot (0, 1, 3, 9), mutta usein ne esitetään matriisissa symboleina. (Alrabghi 2013, 38.)

2.2.5 Riippuvuudet Miten-Miten

Ominaisuuksien välisten riippuvuuksien esittämiseen käytetty matriisi, Laaduntalon katto on usein sen vähiten käytetty osa. Silti sen mahdolliset edut ovat suuria. Matriisia käytetään analysoitaessa mitkä ominaisuuksista tukevat toisiaan ja mitkä ovat keskenään ristiiriitaisia. (Alrabghi 2013, 40).

Ominaisuuksien välillä on viittä eri tyyppiä riippuvuuksia: vahva positiivinen, positiivinen, ei vaikutusta, negatiivinen ja vahva negatiivinen. Riippuvuudet esitetään matriisissa symboleilla plus-plus, plus, tyhjä, miinus ja miinus-miinus. Negatiiviset riippuvuudet osoittavat tilanteita, jotka tulevat olemaan rajoitteita suunnittelussa ja vaativat kompromisseja, lisäsuunnittelua tai läpimurtojen löytämistä. Liian monet positiiviset riippuvuudet vihjaavat, että joko asiakastarpeissa tai ominaisuuksissa on sama asia kahteen kertaan. (Alrabghi 2013, 40.)

2.2.6 Ominaisuusmatriisi

Ominaisuusmatriisin tarkoituksena on selvittää ominaisuuksien alustava suhteellinen tärkeys aiemmista vaiheista saatujen tietojen perusteella. Jotta kilpailuasetelmasta saataisiin parempi kuva, verrataan ominaisuuksien kyvykkyksiä kilpaileviin tuotteisiin. Myös tekniset rajoitteet, jotka estävät ominaisuuksien tavoitearvojen saavuttamisen voidaan esittää. Lopuksi lasketaan ominaisuuksien strateginen tärkeys ja tärkeimmät ominaisuudet valitaan seuraavan QFD-vaiheen syötteiksi. (Alrabghi 2013, 41.)

3 QFD ohjelmistotuotannossa

Tämä luku keskittyy QFD:n soveltamiseen ohjelmistotuotantoon. Luvussa käydään läpi miten ohjelmistotuotanto eroaa perinteisemmästä teollisuudesta ja miten se vaikuttaa QFD-menetelmän soveltamiseen. Malleja QFD:n soveltamiseksi ohjelmistotuotantoon on monia ja tässä luvussa esitellään muutamia. Lopuksi vertaillaan esiteltyjä soveltamismalleja ja käydään läpi kokemuksia QFD:n käytöstä ohjelmistotuotannossa.

Ohjelmistotuotanto eroaa muutamissa asioissa merkittävästi fyysisten tuotteiden valmistamisesta. Ohjelmistot ovat abstrakteja, monimutkaisia ja ainutlaatuisia. Ohjelmistoja kehitetään, ei valmisteta. Kun ohjelmisto on kehitetty valmiiksi, voidaan sitä tuotteena monistaa loputtomasti ilman merkittäviä kustannuksia, koska ohjelmistotuotteilla ei ole juuri lainkaan materiaalikustannuksia. Tästä syystä aika ja tieto tulevat paljon tärkeämmiksi ohjelmistotuotteen kokonaisprosessissa. Koska ohjelmistokehitys on luovaa ja inhimillistä toimintaa, se vaatii kommunikaatiota. Kommunikaation tarve merkitsee sitä, että monet ohjelmistokehityksen tehtävät eivät ole pilkottavissa pienemmiksi osiksi. Kaikkea prosessiin vaikuttavia tekijöitä on vaikea hallita. Koska kaikki järjestelmät ovat ainutlaatuisia ja teknologia kehittyi jatkuvasti, ei ole suoraviivaista toistaa aikaisempien projektien onnistumisia uusissa projekteissa. (Koski 2003, 35.)

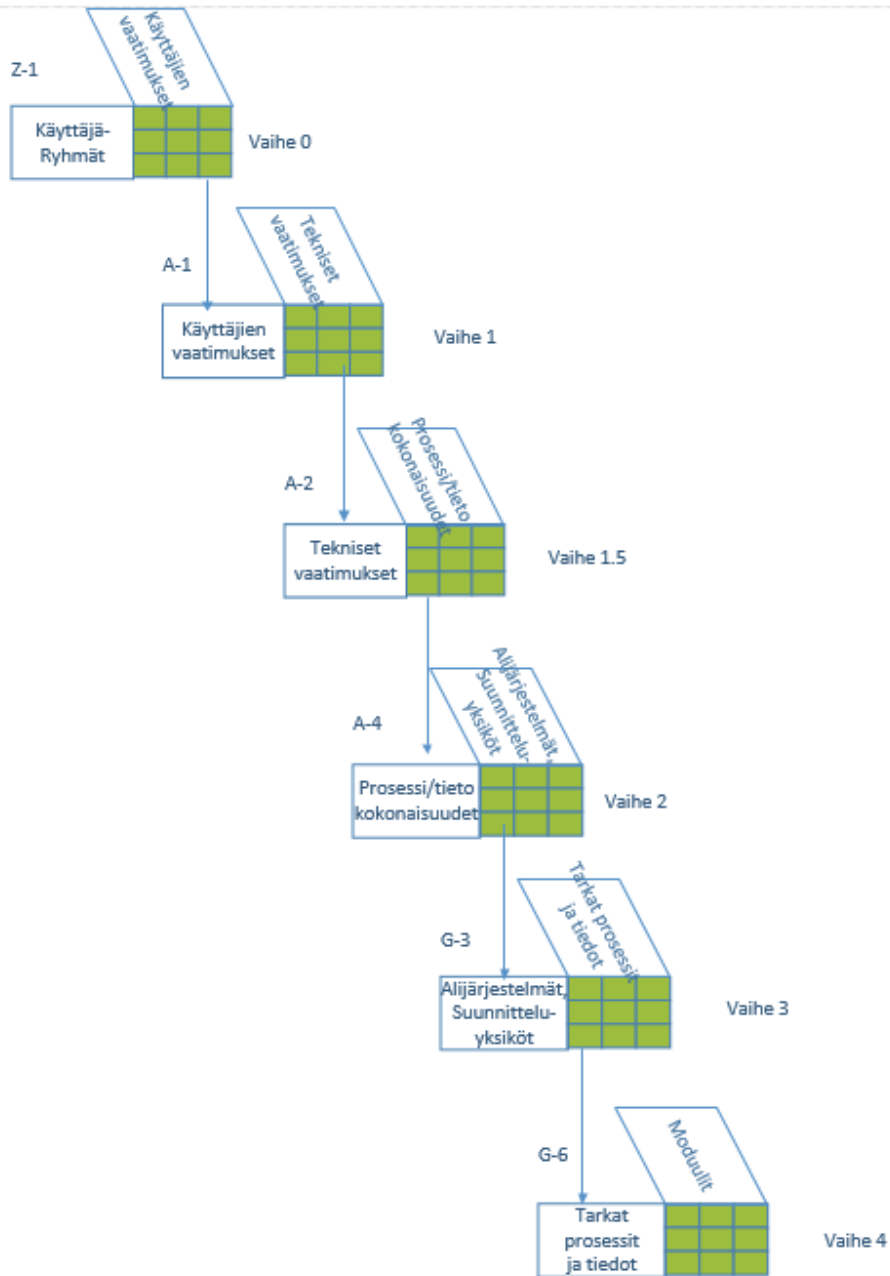
QFD menetelmää on onnistuneimmin hyödynnetty valmistavassa teollisuudessa. Sen hyödyntäminen ohjelmistotuotannossa edellyttää muutamia mukautuksia. QFD:n ydin on asiakkaan tarpeiden täyttäminen ja se sopii mainiosti vaatimusmäärittelyn prosessiin. Vaatimusmäärittelyyn pitää sisällään sidosryhmien ohjelmistoon kohdistamien vaatimusten tunnistamisen ja ylläpidon. Koska ohjelmistossa ei ole fyysisiä ominaisuuksia QFD vaatii joitakin muutoksia soveltuakseen ohjelmistotuotantoon. (Koski 2003, 36.)

QFD:n menetelmien muokkaaminen ohjelmistotuotannon käyttöön valmistavan teollisuuden sijaan lähti liikkeelle 1984, kun japanilaiset alkoivat tutkia sen käyttömahdollisuuksia sulautettujen ohjelmistojen kehityksessä. Digital Equipment Corporation (DEC) ilmoitti neljä vuotta myöhemmin hyödyntävänsä QFD:tä ohjelmistojen kehityksessä. 1987 Richard Zultner kuvasi miten QFD voidaan integroida ohjelmistojen suunnitteluun. Betts yhdisti QFD periaatteiden käytön koko ohjelmistotuotannon elinkaareen 1989. Työskennellessään Hewlett Packardilla hän sovelsi QFD:tä yrityksen laatu-tietojärjestelmä projektissa. Menetelmiä jatkokehitettiin 1996 vaatimusten hallinnassa ja käyttöön otettiin termi Software Quality Function Deployment (SQFD). SQFD muuntaa valmistavassa teollisuudessa käytetyt perinteiset QFD menetelmät ohjelmistotuotantoon sopiviksi. SQFD:tä ovat

ohjelmistotuotannossa käyttäneet Hewlett Packardin lisäksi AT&T, IBM, Texas Instrument ja SAP. (Kivinen 2008, 34-35.)

3.1 Software Quality Deployment

Zultner oli eräs ensimmäisistä, joka kehitti ja julkaisi mallin QFD:n soveltamisesta ohjelmistojen kehityksessä. Hän käytti mallistaan nimeä Software Quality Deployment (SQD). Mallin lähestymistavan perusta on matriisien käyttö eri asioiden välisten riippuvuuksien yksityiskohtien selvittämisessä. Mallia voidaan soveltaa ohjelmistokehityksessä monella eri kompleksisuuden tasolla. Vaihtoehtoisesti voidaan käyttää vain neljää perusmatriisia, monimutkaisemmissa tapauksissa kolmekymmentä tai jopa 150 matriisia. Matriisien dimensioita ovat esimerkiksi kustannukset, asiakastarpeet ja ympäristön rakenne. Malli käyttää nelivaiheista lähestymistapaa, johon on lisätty muutama ylimääräinen vaihe. Mallin vaiheistus on esitetty kuviossa 4. (Koski 2003, 36.)



Kuvio 4. SQD vaiheet (mukailtu Koski 2003, 36)

QFD alkaa asiakastarpeiden listaamisella laaduntalomatriisiin. Ohjelmistoprojekteissa asiakastarpeita ovat vaatimukset käyttäjiltä ja muilta sidosryhmiltä. Yleensä ohjelmiston on palveltava useiden eri sidosryhmien tarpeita. Nämä sidosryhmät on tunnistettava, ymmärrettävä ja priorisoitava ennen kuin työskentely laaduntalon kanssa voidaan aloittaa. SQD esittelee uuden matriisin käyttäjien ja sidosryhmien luokitteluun ja tunnistukseen. (Koski 2003, 36.)

Kun sidosryhmät on tunnistettu, aloitetaan vaatimusten, tarpeiden ja uhkien kerääminen. Tähän käytetään haastatteluita, kyselyitä, tiimi analyysiä, ryhmähaastatteluita, ongelma-raportteja ja virhelokeja sekä palautetta käytössä olevista järjestelmistä. Nämä jalostetaan

yksiselitteisiksi käyttäjäodotuksiksi. Käyttäjäodotukset järjestetään vaatimushierarkiaksi ja vaatimusten ja sidosryhmien väliset riippuvuudet analysoidaan omassa matriisissaan. (Koski 2003, 37.)

Käyttäjien vaatimukset käännetään teknisiksi vaatimuksiksi laaduntalomatriisia käyttämällä. Prosessi on hyvin samankaltainen kuin valmistettavien tuotteiden tapauksessa. Zultner tosin ehdottaa, että matriisiin ei tulisi esittää "mitä" ja "miten" kuten normaalissa tapauksessa vaan ohjelmistojen kanssa parempi vaihtoehto olisi käyttää "miksi" ja "mitä". Vaatimusten painoarvot lasketaan normaalisti, käyttäen tarvittaessa säätökertoimia. Teknisten vaatimusten painoarvot lasketaan perinteisellä tavalla. (Koski 2003, 38.)

Sen sijaan, että jatkettaisiin seuraavaksi QFD prosessin Osa-vaiheeseen, SQD:ssä tehdään seuraavaksi teknisten vaatimusten kääntäminen tieto- ja prosessimalleiksi. Teknisten vaatimusten suhde tietoon ja prosessiin analysoidaan matriisilla. Matriisissa esitetään mitä tietoja tarvitaan, mutta ei oteta kantaa toteutustapaan. Samaan tapaan prosesseista esitetään prosessoinnin tarpeet, mutta ei miten se toteutetaan. (Koski 2003, 38.)

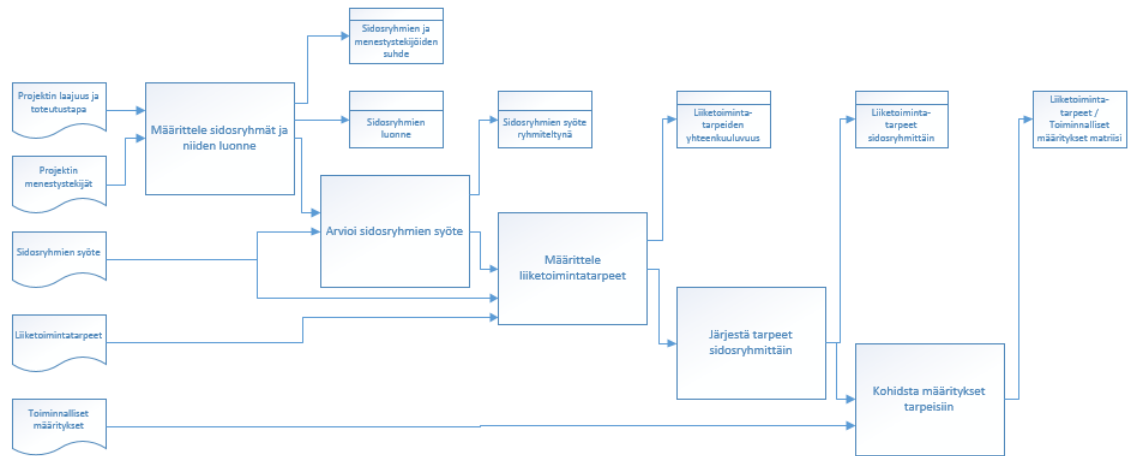
SQD sisältää matriisin tietojen ja prosessien suhteiden kuvaamiseen. Tämän tarkoitus on varmistaa, että tieto- ja prosessimallit ovat ristiriidattomia. Muutkin samankaltaiset matriisit voivat olla hyödyllisiä. Oliomallinnuksessa voidaan käyttää olio-entiteettimatriisia ja tapahtumapohjaisessa mallinnuksessa tapahtuma-entiteettimatriisia. Nykyaikaiset työkalut, kuten UML-mallinnusohjelmistot voivat tehdä saman asian vaivattomammin. (Koski 2003, 39.)

SQD analyysia voidaan jatkaa myös seuraaviin vaiheisiin. QFD voidaan sovittaa ohjelmistoihin korvaamalla materiaali tiedolla, kustannukset ajalla ja toiminnot prosesseilla. SQD:ssä on monia matriiseja, jotka voivat olla käyttökelpoisia eri asioiden jäsentämiseen ja kommunikointiin. Esimerkiksi uuden konseptin sopivuutta voidaan analysoida teknisiä vaatimuksia vasten tai konseptin sisältämät riskit voidaan osoittaa. SQD lähestymistapaa voidaan säätää erilaisten ohjelmistoprojektien tarpeita vastaavaksi. (Koski 2003, 39.)

3.2 Method/1

Accenture soveltaa QFD-menetelmää Method/1 ohjelmistoprosessissaan. QFD:n soveltamisen painopiste on suunnittelussa ja analyysissa, joiden aikana projektin laajuus ja sen tuottama arvo määritellään. Analyysi- ja suunnitteluvaihe sisältää viisi vaihetta, joiden jälkeen toteutusta johdetaan suunnitellussa saatujen priorisointien pohjalta. Kuviossa 5 esitetään vaiheet, niiden käyttämät pohjatiedot ja vaiheiden tuotokset. Tehtävät matriisit

muistuttavat Laaduntaloa, mutta niissä ei käytetä kattoa, jossa määriteltäisiin ominaisuuksien väliset riippuvuudet. (Krogstie 1999, 4-5.)



Kuvio 5. Method/1 vaiheet (mukailtu Krogstie 1999, 6)

Ensimmäisessä vaiheessa määritellään sidosryhmät ja niiden luonne. Vaihe tuottaa matriisin, jossa kuvataan sidosryhmien ja projektin menestystekijöiden suhde. Vaihe auttaa projektiryhmää ymmärtämään projektin kannalta tärkeitä sidosryhmiä ja syyt näiden sidosryhmien tärkeyteen. (Krogstie 1999, 4-5.)

Toisessa vaiheessa kerätään ja arvioidaan sidosryhmien syöte. Sidoryhmien syöte ryhmitellään määritysten keräyksen aikana syötetauluun. Syötetaulu tuottaa syötteen tieto-, prosessi- ja oliomalleille sekä toiminnallisille ja ei-toiminnallisille määrityksille. (Krogstie 1999, 4-5.)

Kolmannessa vaiheessa määritellään liiketoimintatarpeiden ongelmat ja mahdollisuudet, jotka sovellus voi ratkaista. Liiketoimintatarpeiden yhteenkuuluvuuskaavio esittää alunäkymän liiketoimintatarpeiden rakenteesta. (Krogstie 1999, 4-5.)

Seuraavaksi ryhmitellään liiketoimintatarpeet sidosryhmätyypeittäin. Projektin on arvioitava kuinka tärkeää tarpeiden täyttäminen on eri sidosryhmille. Sidoryhmien tyytyväisyystaso tarjoaa keinot tarpeiden priorisointiin. (Krogstie 1999, 4-5.)

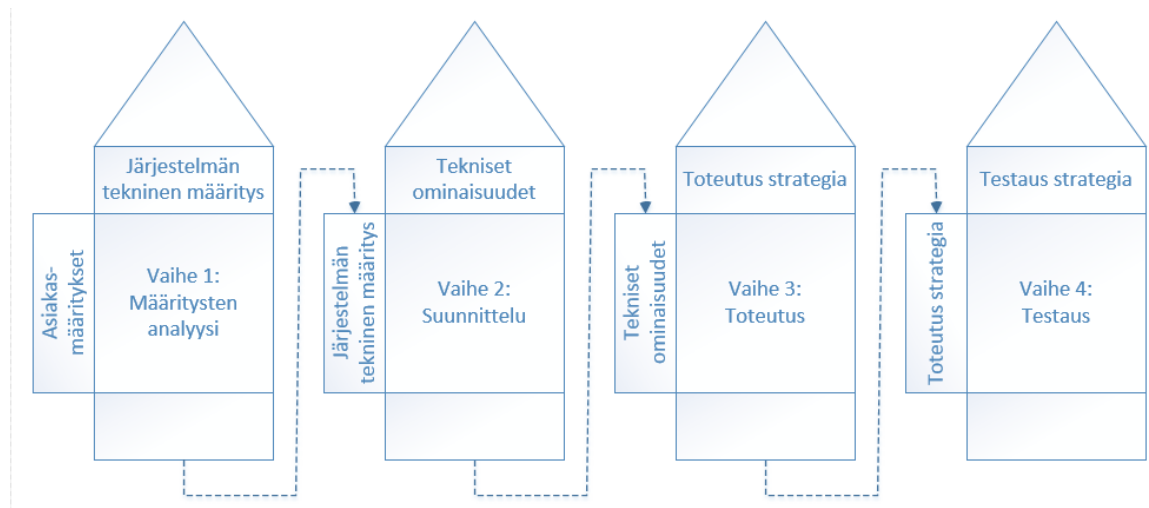
Viimeisessä vaiheessa kohdistetaan vaatimukset tarpeisiin. Liiketoimintatarpeiden ja toiminnallisten määritysten matriisi priorisoi toiminnalliset vaatimukset sen mukaan kuinka hyvin ne auttavat liiketoimintatarpeiden saavuttamisessa. Samaa toimintatapaa voidaan käyttää ei-toiminnallisiin vaatimuksiin ja projektin vaatimuksiin. Vaiheen tuloksena on

kolme matriisia, jotka sisältävät liiketoimintatarpeiden riippuvuuden toiminnallisista, ei-toiminnallisista ja projektin määräyksistä. Jos projektissa toteutettavan tietojärjestelmän on tarkoitus pohjautua valmisohjelmistoon, niin tässä vaiheessa tehdään vertailu myös siitä miten hyvin eri vaihtoehtoina olevat valmisohjelmistot täyttävät liiketoimintatarpeet ilman mukautusta tai räätälöintiä. Tämän pohjalta tehdään valinta siitä minkä valmisohjelmiston pohjalta järjestelmä toteutetaan. (Krogstie 1999, 4-5.)

Analyysi- ja suunnitteluvaiheiden jälkeen toteutusvaiheessa johdetaan projektin tuottamaa arvoa asiakkaalle. Vaatimusten priorisointi palvelee ohjeena läpi järjestelmäkehitys prosessin. Prioriteetit ja arvon tuottaminen määrittelevät mihin tehtäviin projektin kannattaa keskittää voimavaransa. (Krogstie 1999, 4-5.)

3.3 Software QFD

Software QFD (SQFD) on normaalista nelivaiheisesta mallista modifioitu malli, jossa vaiheet on vaihdettu ohjelmistotuotantoon sopiviksi. kuviossa 6 on esitetty mallin vaiheet ja niiden eteneminen. SQFD alkaa vaatimusten analysointi vaiheella. Mallin ajatuksena on, että teknisten määritysten täytyy perustua asiakkaan vaatimusmäärittämiin. Sellaiset tekniset määritykset, jotka eivät perustu asiakkaan vaatimuksiin poistetaan. (Alrabghi 2013, 71.)



Kuvio 6. SQFD vaiheet (mukailtu Alrabghi 2013, 72)

Seuraavaa vaihetta kutsutaan suunnitteluvaiheeksi, ja se korvaa nelivaiheisen QFD mallin osien suunnittelu vaiheen. Toiminnalliset ja ei-toiminnalliset vaatimukset toimivat ohje-
nuorana, kun suunnitellaan ohjelmiston arkkitehtuuri, moduulirakenne, tietomalli ja käyttö-

liittymä. Vaiheesta syntyy matriisi, jossa ovat riippuvuudet teknisten määritysten ja yksityiskohtien suunnittelun välillä. (Alrabghi 2013, 72.)

Nelivaiheisen QFD mallin prosessivaihe korvataan toteutusvaiheella. Yksityiskohtainen suunnitelma toimii syöteenä toteutusstrategialle. Toteutusvaiheessa valitaan ohjelmointikieli ja käytettävät työkalut yksityiskohtien suunnittelun perusteella. (Alrabghi 2013, 72.)

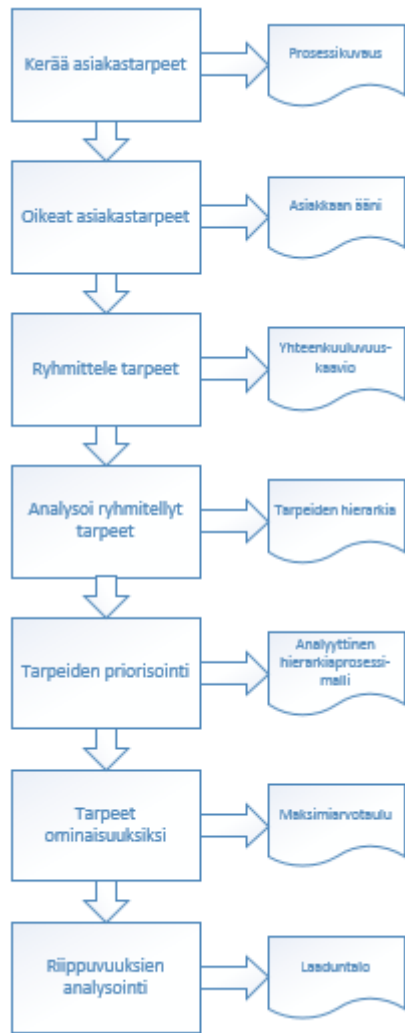
QFD mallin neljäs ja viimeinen vaihe on valmistus. SQFD korvaa sen testausvaiheella. Testausvaiheessa matriisia käytetään testistrategian luomiseen toteutusstrategian pohjalta. Testausvaiheessa laaditaan testausmenetelmät ja testaus suunnitellaan löytämään virheet ohjelmasta. (Alrabghi 2013, 72.)

SQFD esittää korkean tason mallin ohjelmistotuotantoprosessista. Se ei anna tarkempia ohjeita siitä kuinka mallia pitäisi käytännössä toteuttaa. Myös vaiheiden järjestys eroaa jossain määrin todellisista ohjelmistoprojekteista. Esimerkiksi testaussuunnittelu aloitetaan yleensä paljon aikaisemmin. (Koski 2003, 43-44.)

3.4 Blitz QFD

Blitz QFD soveltuu käytettäväksi nopeassa ohjelmistokehityksessä. Sen tarkoituksena on saavuttaa mahdollisimman suuret hyödyt mahdollisimman pienellä panostuksella menettämättä. Ohjelmistotuotannossa on tarve tehdä tuote sekä nopeasti että laadukkaasti samanaikaisesti. Mahdollisimman aikainen markkinoille pääsy on menestyksen edellytys ja tuotteen täytyy täyttää asiakkaiden tarpeet. Ainoastaan muutamilla tehtävillä ohjelmiston kehityksen lukemattomista tehtävistä on vahva tai suora yhteys tärkeimpiin asiakasvaatimuksiin. Tämän takia Blitz QFD keskittyy löytämään ne muutamat ominaisuudet, jotka tuottavat suurimman arvon asiakkaille ja aikataulun kannalta kriittisimmät toiminnot. (Koski 2003, 43-44; Alrabghi 2013, 62.)

Blitz QFD keskittyy pelkästään asiakastarpeiden ymmärtämiseen, analysointiin ja priorisointiin. Se koostuu seitsemästä vaiheesta, joiden järjestys ja niiden tuotokset on esitetty kuviossa 7. Blitz QFD vaiheissa ei juurikaan käytetä matriiseja vaan vaiheiden tuotokset ovat yleensä tauluja tai rakenteellisia esitysmalleja. (Koski 2003, 25.)



Kuvio 7. Blitz QFD vaiheet

Blitz QFD alkaa asiakastarpeiden keräämisellä. Tässä vaiheessa tarkoitus on mennä sinne, missä kehitettävää tuotetta tullaan käyttämään ja tarkkailla asiakkaita ja heidän tarpeitaan todellisessa käyttöympäristössä ja -tilanteissa. Näiden ymmärtäminen antaa paremman mahdollisuuden löytää lisäarvoa asiakkaalle tuottavat asiat. Havaintojen perusteella tehdään kuvaus asiakkaan prosessista. (Alrabghi 2013, 63.)

Seuraavassa vaiheessa analysoidaan edellisessä vaiheessa kerätyt havainnot ja tarpeet. Asiakkaat eivät ole vaatimusmäärittely ammattilaisia, ja heidän tarpeensa antavat ehkä vain vihjeitä oikeista tarpeista. Analyysin tehtävänä on selvittää todelliset tarpeet asiakkaiden käyttäminen sanojen ja ilmausten takana. Analyysin lopputulos koostetaan asiakkaan ääni tauluun. (Alrabghi 2013, 63.)

Seuraavaksi ryhmitellään asiakastarpeet yhteenkuuluvuuskaavioon. Yleensä asiakkaat tekevät tämän, koska asiakkaiden tekemä vaatimusten ryhmittely auttaa ymmärtämään sitä miten asiakkaat hahmottavat kokonaisuuden. (Alrabghi 2013, 63.)

Seuraavaksi asiakkailta saadut yhteenkuuluvuuskaaviot analysoidaan. Tässä kohtaa on tärkeä saada ymmärrys asiakkaiden ajattelutavasta ja löytää mahdolliset piilevät vaatimukset sekä muuttaa vaatimukset mitattaviksi. Tarpeet ryhmitellään hierarkiakaaviin. (Alrabghi 2013, 63.)

Asiakastarpeiden priorisointiin käytetään analyttistä hierarkiaproessia (AHP). Tarpeiden priorisoinnilla löydetään ne tarpeet, jotka tuottavat eniten arvoa asiakkaille. Priorisointi tehdään parivertailuna niin, että jokaista tarvetta verrataan kaikkiin muihin vuorollaan ja jokaisessa parissa annetaan kerroin sille, kuinka paljon tärkeämpi toinen on. (Alrabghi 2013, 64.)

AHP:n parivertailu toteutetaan vertaamalla jokaista vaihtoehtoa kaikkiin muihin, kunnes kaikki vaihtoehdot ovat vertailtu keskenään. Vertailussa käytetään arvoja 1, 3, 5, 7 ja 9. Arvo 1 tarkoittaa, että vaihtoehdot ovat keskenään yhtä tärkeitä, kun taas 9 tarkoittaa sitä että ensimmäinen vertailtavista vaihtoehdoista absoluuttisesti tärkeämpi kuin toinen. Vaihtoehdot sijoitetaan matriisiin ja matriisin soluihin laitetaan vertailun tulos. Matriisin rivit kerrotaan keskenään ja saadusta tulosta otetaan neljäsjuuri. Saadut tulokset normalisoidaan ja tulokseksi saadaan tarpeiden painoarvot. (Villanen 2013, 3.)

Taulukossa 1 on esimerkki AHP:n parivertailusta. Jokainen vaihtoehto on itsensä kanssa tasavertainen, joten matriisin lävistäjän arvot ovat aina 1. A on tärkeämpi kuin B ja sille on annettu arvoksi 5. Vastaavasti, kun verrataan B:tä A:han, annetaan arvoksi käänteisluku 1/5. (Villanen 2013, 3.)

Taulukko 1. Vaihtoehtojen priorisointi parivertailulla (mukailtu Villanen 2013, 3)

Vaihtoehdon tärkeys	A	B	C	D	Rivin tulo	Tulon neäljäs juuri	Normalisoitu painoarvo
A	1	5	6	7	210,000	3,807	0,614
B	1/5	1	4	6	4,800	1,480	0,239
C	1/6	1/4	1	4	0,167	0,639	0,103
D	1/7	1/6	1/4	1	0,006	0,278	0,045
					Summa:	6,204	1

Tärkeimmät asiakastarpeet johdetaan ominaisuuksiksi. Maksimiarvo taulukkoa käytetään tähän tarpeeseen. Ominaisuudet, jotka tuottavat suurimman arvon asiakkaalle ovat ne, jotka määrittävät onnistumisen tai epäonnistumisen. Näihin ominaisuuksiin keskitytään toteutusvaiheessa. (Alrabghi 2013, 64.)

Viimeisessä vaiheessa analysoidaan maksimiarvoa tuottavien ominaisuuksien riippuvuudet, jotta niiden toteutuksessa vältetään epäonnistumiset. Tähän tarkoitukseen käytetään Laaduntaloa, josta voidaan täyttää vain tärkeimmät tarpeet, ominaisuudet ja niiden väliset riippuvuudet. Tämän vaiheen työkaluina voidaan käyttää normaaleja projektin työnositus-, tehtävälislaus- ja riskianalyysityökaluja. (Alrabghi 2013, 64.)

3.5 Kokemuksia QFD-menetelmän käytöstä ohjelmistokehityksessä

QFD-menetelmän käytöstä ohjelmistokehityksessä on tehty jonkin verran tutkimuksia. Herzworm ja Schockert toteavat, että QFD täyttää välin, joka useimmissa ohjelmistokehitysmenetelmissä on jätetty huomiotta. Tämä väli on asiakkaan ja toteutuksen väli. QFD:n avulla toteutustiimi voi johtaa tuotteen ominaisuudet asiakkaan tarpeista sen sijaan, että asiakkaan pitäisi osata antaa tarkkoja tavoitteita. Herzworm ja Schockert mukaan menetelmän käytön on todettu erityisesti parantavan poikkiteollista kommunikaatiota, kirkastavan asiakkaan ja käyttäjän tarpeita kehittäjille, vähentävän toimituksen jälkeisiä muutoksia sekä auttavan saavuttamaan kannattavan tuotteen ja tyytyväiset asiakkaat. QFD:n huonoiksi puoliksi voidaan heidän mukaansa laskea monimutkaisuus ja runsas ajan käyttö, jota tarvitaan mallin toteuttamisessa. (Herzworm & Schockert, 2003.)

Koski tutki QFD:n käyttökokemuksia Vaisalassa haastatteleamalla projektien vetäjiä, jotka olivat käyttäneet QFD:tä projekteissaan. Projekteista yksikään ei ollut puhdas ohjelmistokehitysprojekti, mutta kaikissa tapauksissa joko kehitettiin myös ohjelmistoja tai tehtiin valintoja käyttöön otettavista tietojärjestelmistä. Yhdessäkään tapauksista projektiryhmän jäsenillä ei ollut aikaisempaa kokemusta QFD:stä. Myöskään erityistä koulutusta menetelmästä ei projekteissa järjestetty vaan oppiminen tapahtui menetelmää käytettäessä projektin vetäjän opastamana. Metodin käyttöönotto todettiin suhteellisen helpoksi, mutta sen toteutus voi vaatia paljon työtä. Kaikki haastatellut pitivät QFD:n käyttöä onnistuneena ja menetelmän uskottiin leviävän laajempaan käyttöön yrityksessä. (Koski 2003, 87-88.)

3.6 Menetelmien vertailu

QFD:n soveltamiseksi ohjelmistokehityksessä on kehitetty hyvin monia malleja, joista yksikään ei ole dominoivassa asemassa. Nämä mallit sisältävät kuitenkin yhteisiä lähtökoh-
tia. Kaikkien mallien painopiste on ohjelmistokehitysprosessin aikaisissa vaiheissa. Näitä aikaisia vaiheita ovat asiakkaiden ja sidosryhmien tunnistaminen sekä vaatimusmäärittäminen. (Koski 2003, 46.)

Varsinaisessa toteutusvaiheessa QFD ei ole niin hyödyllinen ja kaikki mallit eivät toteutusvaiheeseen asti edes ulotu. Ohjelmistojen abstrakti luonne voi olla syynä siihen, että toteutusvaiheessa QFD ei ole niin hyödyllinen. Ohjelmistoja suunnitellaan eikä valmisteta, joten toteutusvaihe ei ole samalla tapaa vakioitu ja toistettava kuin tuotteiden valmistuksessa. Toteutusvaiheen työhön on käytettävissä lukuisia muita toimivia malleja ja työkaluja esimerkiksi UML-mallinnus. (Koski 2003, 46.)

Tässä esitellyistä malleista Software QFD eroaa muista siinä, että se sisältää kaikki ohjelmistoprojektin vaiheet. SQD ei sinällään ota kantaa toteutukseen kuin toteamalla, että suunnitteluvaiheesta voisi jatkaa edelleen myös toteutuksen puolelle käyttämällä myös toteutusvaiheessa laaduntalometrieseja. Blitz QFD ja Method\1 eivät sisällä ollenkaan toteutusvaihetta vaan ne ovat puhtaasti analyysi- ja suunnitteluvaiheen menetelmiä. Taulukossa 2 on vertailtu miten neljän esitellyn mallin vaiheet sijoittuvat analyysin, suunnittelun, toteutuksen ja testauksen välillä.

Taulukko 2. Mallien vaiheiden jakautuminen

Malli	Analyysi	Suunnittelu	Toteutus	Testaus
Blitz QFD	5	2		
Method\1	4	1		
Software QFD	1	5		
SQD	1	1	1	1

Software QFD ja SQD perustuvat molemmat matriisien käyttöön, kun taas Method\1 ja Blitz QFD käyttävät taulukoita ja rakenteellisia kaavioita. Niissä molemmissa tehdään vasta viimeisessä vaiheessa yksinkertaistettu laaduntalomatriisi.

4 Fennian yritysvakuutusjärjestelmien uudistaminen

Tässä luvussa esitellään lyhyesti Fennia yrityksenä ja Fennian yritysvakuutusjärjestelmät sekä käydään läpi yritysvakuutusjärjestelmien uudistamisen esiselvitysprojektia. Esiselvitysprojektin tuotoksia käytetään lähtötietoina, joiden perusteella QFD-menetelmää hyödyntäen pyritään muodostamaan kuva siitä, mihin ominaisuuksiin vaihtoehtojen vertailussa pitäisi keskittyä.

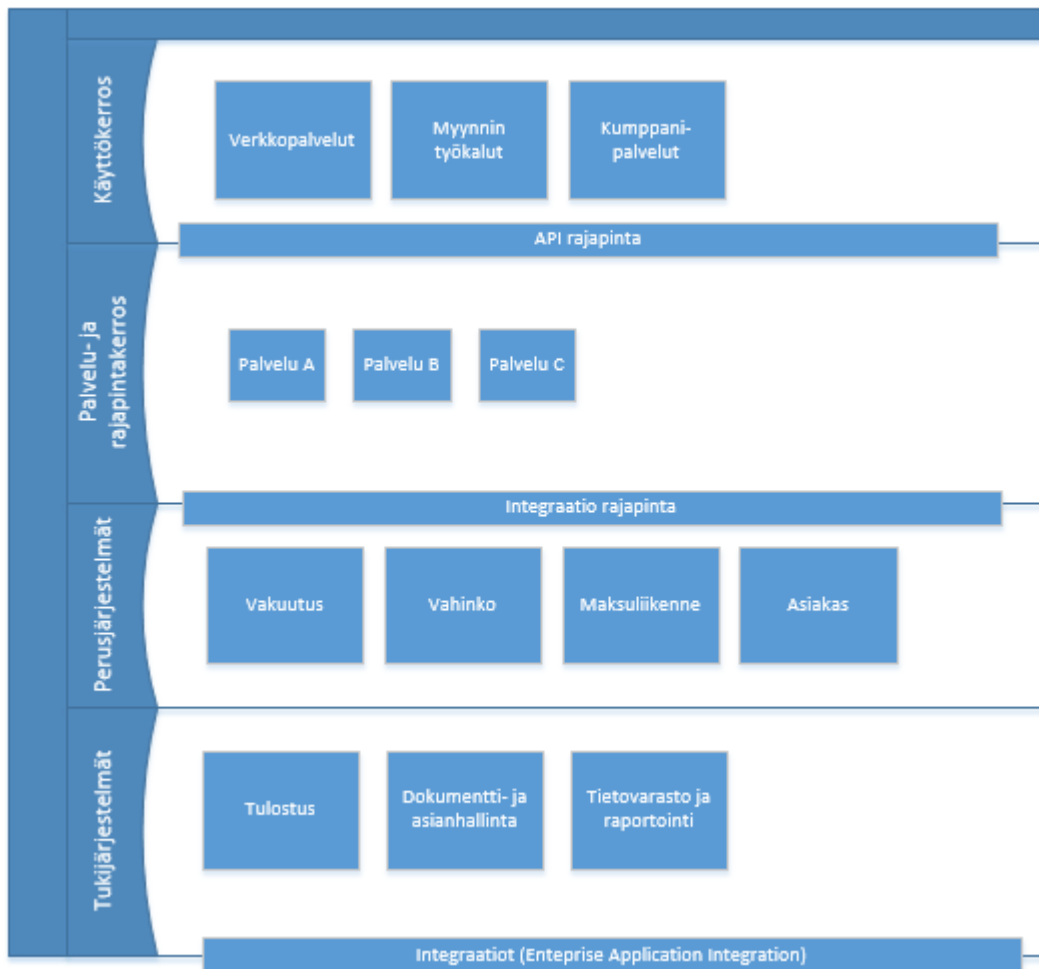
QFD on menetelmänä luonnostaan joustava ja sitä voidaan soveltaa käyttötarpeen mukaan. Niinpä tässä ei yritetä tarkasti noudattaa mitään esitellyistä malleista. Blitz QFD on esitellyistä menetelmistä sopivin tähän tarkoitukseen. Se on tarkoitettu kevyeksi vaihtoehdoksi verrattuna perinteiseen QFD-menetelmään. Tässä tutkimuksessa on tarkoitus johdattaa asiakastarpeista tietojärjestelmän tärkeät ominaisuudet, joten Blitz QFD soveltuu vaiheistukseltaan tähän hyvin. Method\1 on hyvin samankaltainen kuin Blitz QFD, joten sitäkin voitaisiin käyttää, mutta mallit eroavat toisistaan siinä, että Method\1 on suunniteltu ulkopuolisen toimittajan käyttöön ja siinä on pieniä painotuseroja tästä johtuen.

Keskinäinen Vakuutusyhtiö Fennia on vahinkovakuutusyhtiö, jonka juuret ulottuvat 1862 perustettuun Palovakuutusosakeyhtiö Fenniaan. Nykymuotoinen Fennia syntyi, kun 1984 Yrittäjainvakuutus ja Fennia yhdistyivät. Fennia tarjoaa kaikkia lakisääteisiä ja vapaaehtoisia vahinkovakuutuksia yrityksille, yrittäjille ja kotitalouksille. Fennia-konserniin kuuluu lisäksi vapaaehtoisia henki-, eläke- ja säästövakuutuksia tarjoava Henki-Fennia sekä varainhoidollisia palveluja tarjoava Fennia Varainhoito Oy. Fennian vakuutusmaksutulo vuonna 2014 oli 430 miljoonaa euroa ja yhtiön palveluksessa oli keskimäärin 1077 henkilöä. (Fennia 2014, 2-3.)

Vakuutusjärjestelmä on yksi vakuutusyhtiön ydintietojärjestelmistä. Vakuutusjärjestelmän toimintoja ovat vakuutustuotteiden hallinta ja hinnoittelu sekä vakuutus sopimusten elinkaaren hallinta. Sopimuksen elinkaari koostuu tarjouksen teosta, sopimukseksi hyväksymisestä, muutoksista sopimukseen, vuosittaisista uudistuksista ja lopuksi sopimuksen päättämisestä.

Kuvio 8 esittää Fennian vakuuttamiseen liittyvien järjestelmien loogista arkkitehtuuria. Useimpia loogisen arkkitehtuurin mukaisia järjestelmiä on Fenniassa enemmän kuin yksi ja jokin tietojärjestelmä voi sisältää useamman loogisen järjestelmän, mutta loogisesti vakuuttamiseen liittyvät toiminnot voidaan jakaa neljään perusosaan. Vakuutusjärjestelmä on integroitu kaikkiin muihin vakuutuksiin liittyviin järjestelmiin. Vakuutusjärjestelmä hakee asiakkaan tiedot Asiakasjärjestelmästä ja vie vakuutusten veloitukset Maksuliikennejärjes-

telmään, joka muodostaa laskut ja valvoo, että laskut tulevat maksetuiksi. Vahinkojärjestelmään haetaan vahingonsattumishetken tiedot vakuutuksesta vahinkokäsittelyn perustaksi.



Kuvio 8. Fennian vakuuttamiseen liittyvien järjestelmien looginen arkkitehtuuri

Nykyiset yritysvakuuttamisen järjestelmät on otettu käyttöön 1990 ja 1995. Molemmat ovat räätälöityjä järjestelmiä, joita on jatkuvasti kehitetty käyttöönoton jälkeen.

Kokonaisuudessaan nämä järjestelmät koostuvat yli miljoonasta koodirivistä. Molemmat järjestelmät alkavat olla teknisesti vanhentuneita ja niiden kehittäminen on nykyaikaisiin järjestelmiin verrattuna hidasta.

4.1 Esiselvitysprojekti

Fenniassa tehtiin esiselvitysprojekti yritysvakuutusjärjestelmien uudistamisesta vuonna 2015. Projekti kesti koko vuoden ja projektityön eri vaiheisiin osallistui kaikkiaan noin 30 henkilöä. Projektissa käytiin läpi koko yritysvakuuttamisen liiketoimintaprosesseja, nykyisten järjestelmien toimintaa sekä mahdollisia järjestelmien uudistamisvaihtoehtoja. Projek-

tissa kerättiin alustavat liiketoimintavaatimukset koko liiketoimintaprosessia tukeville järjestelmille painottuen kuitenkin uudistettavien vakuutusjärjestelmien vaatimuksiin. Esiselvitysprojekti vastaa hyvin Blitz QFD-menetelmän vaihetta 1 Kerää asiakastarpeet.

Projektissa tehtiin myös eri uudistamisvaihtoehtojen vertailuja ja vaihtoehtoista kaksi todettiin sellaisiksi, että niiden kanssa kannattaa edetä tarkempaan selvitystyöhön. Esiselvitysprojektin puitteissa ei analysoitu vaihtoehtoja niin syvästi, että valintaa voitaisiin sen pohjalta tehdä, mutta kahta jatkoon pääsnyttä lukuun ottamatta muut vaihtoehdot voitiin sulkea pois mahdottomina tai huonoina vaihtoehtoina.

4.2 Liiketoimintavaatimusten analysointi

Esiselvitysprojektin tuloksena syntyi yhteensä 82 kpl liiketoimintavaatimusta. Projektin tarkoituksena ei ollut koota kattavaa vaatimusmäärittelyä esimerkiksi vakuutusjärjestelmän toiminnoista vaan liiketoimintavaatimukset koostuvat joko tärkeistä nykytoiminnoista tai uusista tarpeista ja kehittämiskohteista. Liiketoimintavaatimukset ovat hyvin eritasoisia. Laajimmat vaatimukset ovat koko liiketoimintaprosessin kehittämistarpeita ja yksityiskohtaisimmat kohdistuvat yhden järjestelmän yksittäiseen toimintoon. Liiketoimintavaatimuksista 30 kpl on sellaisia, että ne kohdistuvat pääosaltaan vakuutusjärjestelmään.

Vakuutusjärjestelmään kohdistuvat liiketoimintavaatimuksia voidaan kaikkia pitää Kano-mallin odotettuina tarpeina. Vaatimusmäärittelyt eivät ole kattavat vaan pitävät sisällään sellaisia tarpeita, joita hyvältä vakuutusjärjestelmältä odotetaan, mutta jotka eivät ole mitenkään itsestään selviä perustoiminnallisuuksia. Näin ollen nämä vaatimukset soveltuvat Laaduntalossa käytettäväksi. Vaatimuksissa yksikään ei ole luokiteltavissa Kano-mallin vaadituksi tarpeeksi, joten kaikki 30 on perusteltua ottaa mukaan analyysiin.

Kaikki vakuutusjärjestelmää koskevat 30 liiketoimintavaatimusta ovat hyvin tarkkoja ja yksityiskohtaisia. Vaatimukset tehneet liiketoiminnan edustajat ovat kokeneita vaatimusmäärittelijöitä ja useimmille on pitkä kokemus järjestelmien kehittämisestä. Vanhan järjestelmän uusimisen kyseessä ollessa on helppoa tehdä tarkkoja vaatimuksia, koska olemassa olevan järjestelmän toiminnot tunnetaan tarkasti. Myös kehitystarpeista verrattuna nykyisiin toimintoihin on selkeä tuntuma. Niinpä vaatimusten analyysillä ei tässä tapauksessa ole löydettävissä mitään sellaisia tarpeita, jotka piiloutuisivat käytettyjen sanavalintojen taakse tai liiketoiminnan edustajat eivät olisi osanneet todellista tarvetta tuoda esille.

4.3 Liiketoimintavaatimusten ryhmittely

Esiselvitysprojekti käynnistyi siitä näkökulmasta, että järjestelmien tekninen elinkaari alkaa olla loppuillaan, joten projektin lähtökohtana ei ollut oikeastaan niinkään liiketoiminnalliset kehitystarpeet vaan puhtaasti järjestelmän tekninen uudistaminen. Tämän takia järjestelmä uudistukselle ei ole olemassa varsinaisia liiketoimintatarpeita, mutta liiketoimintavaatimukset ryhmittelemällä voidaan löytää yhdistäviä asioita. Nämä voisivat varsinaisessa uudistusprojektissa olla liiketoimintatarpeita, jotka projektissa pyritään ratkaisemaan pelkän teknisen uudistamisen lisäksi. Liiketoimintatarpeet voisi priorisoida ja sen perusteella saataisiin liiketoimintavaatimukset asetettua tärkeysjärjestykseen sen perusteella kuinka ne auttavat saavuttamaan priorisoidut liiketoimintatarpeet.

Vakuutusjärjestelmään kohdistuvat vaatimukset voidaan ryhmitellä neljään ryhmään:

- tärkeät toiminnot
- tuotekehityksen sujuvoittaminen
- rajapintojen tarjoaminen
- automaation lisääminen

Tärkeät toiminnot ryhmä sisältää vaatimuksia, jotka ovat toteutettu nykyisissä järjestelmissä ja ne toimivat pääsääntöisesti hyvin. Osassa näistä vaatimuksista on myös jonkin verran parannustoi-veita nykytoiminnallisuuteen nähden. Pelkästään nämä vaatimukset toteuttamalla päästäisiin tilanteeseen, jossa uudistettu järjestelmä sisältäisi nykyisten järjestelmien toiminnot ilman merkittävää parannusta niihin. Jos uudistaminen tehtäisiin puhtaasti järjestelmän tekninen uudistaminen ainoana tavoitteena, riittäisi näiden vaatimusten toteuttaminen.

Tuotekehityksen sujuvoittaminen ryhmässä on vaatimuksia, jotka toteuttamalla tuotekehitys ja tuotehuolto sekä liiketoimintasäännösten ylläpito nopeutuisi ja helpottuisi. Näiden vaatimusten toteuttaminen mahdollistaisi myös uuden tyyppisiä ominaisuuksia tuotteille.

Rajapintojen tarjoaminen ryhmässä on vaatimuksia, joiden toteuttaminen mahdollistaisi vakuutusjärjestelmän tietojen ja toimintojen käyttämisen muista järjestelmistä tai palveluis- ta käsin.

Automaation lisääminen ryhmän vaatimukset ovat kehityskohteita, joilla ihmisten manuaalisesti tekemiä toistuvia ja paljon työaikaa vieviä toimenpiteitä muutettaisiin järjestelmän automaattisesti suorittamiksi.

4.4 Tarpeiden keskinäinen priorisointi

Tarpeet voidaan priorisoida toisiinsa nähden, kuten Blitz QFD:n vaiheessa neljä tehdään. Priorisointi tehdään parivertailuna ja jokaisessa parissa annetaan numeroarvo joka kertoo tarpeiden suhteellisesta tärkeydestä toisiinsa nähden. Taulukossa 3 on esitetty tarpeiden keskinäinen priorisointi AHP:n parivertailulla. Tärkeimmät toiminnot on priorisoitu muita paljon tärkeämmäksi, koska ne ovat mahdollistava tekijä myös muille tarpeille. Ilman tärkeimpiä toimintoja muita tarpeita ei voi toteuttaa. Tärkeät toiminnot painoarvo on 0,673 eli se vastaa siis 2/3-osaa vertailtujen tarpeiden tuottamasta arvosta.

Taulukko 3. Tarpeiden keskinäinen priorisointi

Tarpeen tärkeys	Automaation lisäys	Tärkeät toiminnot	Rajapintojen tarjoaminen	Tuotekehityksen sujuvuus	Rivin tulo	Tulon neljäosuuri	Normalisoitu painoarvo
Automaation lisäys	1	0,1	1	0,3	0,048	0,467	0,073
Tärkeät toiminnot	7	1	7	7	343,000	4,304	0,673
Rajapintojen tarjoaminen	1	0,1	1	0,2	0,029	0,411	0,064
Tuotekehityksen sujuvuus	3	0,1	5	1	2,143	1,210	0,189
					Summa:	6,392	1

4.5 Järjestelmän tärkeimpien osien johtaminen

Liitteessä 1 on esitetty ensimmäinen hahmotelma vaatimusten ja järjestelmän osien välisistä riippuvuuksista. ilman tarpeiden keskinäisten priorisointien painostusta. Järjestelmän osat ovat toiminnallisia kokonaisuuksia tai moduuleja, joita vakuutusjärjestelmissä tyypillisesti on. Osiin jako perustuu kirjoittajan omiin kokemuksiin vakuutusjärjestelmistä ja ei ole suoraan minkään yksittäisen järjestelmän rakenne vaan yleinen esitys vakuutusjärjestelmän toimintojen jaottelusta.

Kun vaatimusten ja järjestelmän osien välisiin riippuvuuksiin lisätään vielä tarpeiden priorisoinnista saadut painokertoimet, saadaan johdettua ne osat, jotka ovat tarpeiden täyttämiseksi tärkeimmät. Painotettu matriisi on esitetty liitteessä 2. Taulukossa 4 on esitetty järjestelmän osat ja niiden saamat pisteet tärkeysjärjestyksessä.

Taulukko 4. Järjestelmän osien tärkeysjärjestys

Järjestelmän osa	Pisteytys
Veloitukset	26
Sopimusrakenne	20
Integraatiot	17
Sopimushallinta	16
Muutos kesken vakuutuskauden	15
Maksunlaskenta	14
Tuoterakenne	9
Tuotehallinta	7
Sopimuksen päättäminen	7
Sopimuksen voimaansaatto	7
Käyttöliittymä	6
Uudistus	4
Rajapinnat ja palvelut	3
Tarjouksen teko	0

Tärkeimmiksi osiksi pisteytyksen perusteella tulevat sopimuksien rakenne ja hallinta, integraatiot sekä maksujen laskenta ja niiden veloittaminen. Vaihtoehtoisesti tarpeiden tärkeyden pohjalta olisi voinut johtaa vaatimuksille tärkeysjärjestyksen. Tässä tapauksessa tarpeet syntyivät kuitenkin vaatimusten ryhmittelystä ja vaatimukset eivät olleet sellaisia, että ne olisivat auttaneet saavuttamaan useamman kuin yhden tarpeen.

5 Pohdinta

5.1 Johtopäätökset

Tämän työn ensimmäisenä tavoitteena oli perehtyä Quality Function Deployment-menetelmän käyttöön ja erityisesti sen soveltamiseen ohjelmistotuotannossa. Luvussa 2 käytiin läpi QFD:n hyödyntämistä valmistavassa teollisuudessa. Luvussa 3 käytiin läpi ohjelmistotuotannon eroja tuotteiden valmistamiseen sekä erilaisia malleja siitä miten QFD:tä on mahdollista soveltaa myös ohjelmistotuotannossa ja kokemuksia sen soveltamisesta.

Toisena tavoitteena oli kokeilla menetelmän hyödyntämismahdollisuuksien tutkiminen Fennian yritysvaluuttamisen järjestelmien uudistamisvaihtoehtojen vertailussa. Luvussa 4 johdettiin esiselvitysprojektin tuottaminen vaatimusmääritysten pohjalta järjestelmän tärkeimmät osat. Työssä käytetyt vaatimukset eivät ole koko järjestelmän vaatimukset, mutta toisaalta QFD:n perusajatuksena on jokaisesta vaiheesta siirtää seuraavaan vain tärkeimmät asiat. Esiselvitysprojektissa kerätyt vaatimukset olivat liiketoiminnan tärkeinä pitämiä, joten ne voisivat hyvinkin olla ne vaatimukset, joiden perusteella vertailu kannattaisi tehdä. Menetelmänä QFD soveltuisi hyvin vaatimusten johtamiseksi ominaisuuksien tärkeysjärjestykseen ja mahdollistaisi järjestelmien vertailun, joissa ominaisuudet olisivat painotettu tärkeyden mukaan. Tämän työn laajuuden puitteissa käytetty QFD ei anna suoraan mahdollisuuksia kustannusvertailuun, mutta tuloksia voisi käyttää apuna esimerkiksi vertaamalla kustannuksia järjestelmän ominaisuuksien pisteisiin.

5.2 Jatkotutkimusmahdollisuudet

QFD-menetelmän hyödyntämistä Fenniassa voisi tutkia laajemminkin. Fenniassa on vahvat systeemyöperinne ja -toimintamallit. Keskimäärin isojenkin tietojärjestelmäprojektien toteutus onnistuu hyvin niin omilla kuin ulkopuolisillakin resursseilla tehtynä. Näin ollen itse systeemyömallit eivät vaadi suurta kehittämistä, mutta vaatimusten ja tarpeiden priorisointi on asia, josta käydään säännöllistä keskustelua. QFD:n käytöllä voitaisiin saada tarpeet asetettua tärkeysjärjestykseen ja toteuttavalle taholle selkeä käsitys siitä, mitkä asiat ovat onnistumisen kannalta kaikista tärkeimpiä. Tarpeiden tärkeyden perusteella johdettu tärkeysjärjestys vaatimuksille tai ominaisuuksille olisi monessa projektissa hedeelmällistä. Jos osa vaatimuksista on sellaisia, että ne auttavat saavuttamaan useamman tarpeen niin nämä vaatimukset olisi arvokasta löytää ja priorisoida. Toisaalta osa vaatimuksista voi olla sellaisia, että ne eivät edistä tarpeiden saavuttamista, joten niiden toteuttaminen ei edistä sitä, mitä tavoitellaan.

Myös Analyyttisen hierarkia prosessin hyödyntämismahdollisuuksien tutkiminen voisi olla mielenkiintoinen jatkoselvityksen kohde. Parivertailu on yksinkertainen, mutta rationaalinen tapa priorisointiin ja AHP on paljon laajempi kokonaisuus kuin sen tässä käsitelty yksittäinen osa.

5.3 Opinnäytetyön prosessi ja oma oppiminen

Lopputyön tekeminen ei sinänsä ollut suuri oppimiskokemus. Vastaavia, mutta suppeampia töitä olen tehnyt aikaisemmin jo useita sekä Haaga-Heliassa että Teknillisessä korkeakoulussa. Ehkä tärkein oma oppimiskokemus oli työn laajuus ja lopullisen muodon kirkastuminen työn aikana. Vaikka opinnäytetyö on kirjoitettu perinteisessä muodossa teoriaosa ensin, oli työskentelymetodi kuitenkin ehkä enemmän vetoketjumallin mukainen ja teoriaosuus täydentyi soveltavan osan teossa. Ajan järjestäminen opinnäytetyön tekemiseksi lapsiperheen arjen ja työssä käynnin lomassa oli ajoittain haastavaa, mutta onnistui.

Lähteet

- Alrabghi, L. 2013. QFD in software engineering. Luettavissa: https://etd.ohiolink.edu/!etd.send_file?accession=kent1385046526. Luettu: 6.3.2016.
- Fennia 2015. Toimintakertomus ja tilinpäätös 2014. Luettavissa: http://vuosikertomus.fennia.fi/2014/filebank/307-Fennia_Toimintakertomus_ja_tilinja_tilinpaaatos_2014.pdf. Luettu: 16.4.2016
- Herzwurm, G., Schockert, S. 2003. The leading edge in QFD for software and electronic business. International Journal of Quality & Reliability Management 20. vuosikerta nro 1 sivut 36-55. Luettavissa: https://www.bwi.uni-stuttgart.de/abt8/.content/documentcenter/publikationen/Publikationen_Herzwurm/leading_edge_in_qfd.pdf. Luettu: 14.3.2016.
- Hämäläinen, O., Jalarvo, E. 2008. Quality Function Deployment asiakaslähtöisen tuotekehityksen menetelmänä. Luettavissa: <http://urn.fi/URN:NBN:fi-fe200805061322>. Luettu: 6.3.2016.
- Kivinen, T. 2008. Applying QFD to improve the requirements and project management in small-scale project. Luettavissa: <https://tampub.uta.fi/handle/10024/79769>. Luettu: 6.3.2016.
- Koski, J. 2003. Quality Function Deploymentin Requirements Engineering: A Review and Case Studies. Luettavissa: <http://www.soberit.hut.fi/core/reports/mba-jouko-koski.pdf>. Luettu: 6.3.2016.
- Krogstie, J. 1999. Using Quality Function Deployment in Software Requirement Specification. Luettavissa: <http://www.idi.ntnu.no/~krogstie/publications/1999/REFSQ/fulltext.pdf>. Luettu: 6.3.2016.
- Lillrank, P. 1990. Laatunjohtaminen: Johdatus Japanin talouselämään laatujohtamisen näkökulmasta. Gaudeamus.
- Villanen, H. 2013. Päätöksenteko ja analyttinen hierarkiaprosessi, AHP. Luettavissa: http://www.prosessitaito.fi/Paatoksenteko_AHPn_avulla.pdf. Luettu 5.4.2016.

Liitteet

Liite 1. Liiketoimintavaatimusten korrelaatio järjestelmän osiin

Liiketoimintatarve	Liiketoimintavaatimus	Tuoterakenne	Tuotehallinta	Sopimusrakenne	Sopimushallinta	Maksunlaskenta	veloitukset	Tarjouksen teko	Sopimuksen voimaansaatto	Muutos kesken vakuutuskauden	Uudistus	Sopimuksen päättäminen	Rajapinnat ja palvelut	Integraatiot	Käyttöliittymä
Automaation lisäys	LV01			3	9										
	LV07							3							9
	LV09								3	3					9
	LV10									3					9
	LV12			9								3			9
	LV15								3						9
	LV73												9		
Tärkeät toiminnot	LV11														9
	LV16								3	3					9
	LV05						9	9							
	LV08					3	9		9						
	LV14						9				9				
	LV18	9	3												
	LV21			9	9										
	LV23														9
	LV24			9		9									
	LV28					3	9		9						
	LV04							3					9		
	LV41								3				9		
	LV44								3				9		
	LV63						3						9		
Tuotekehityksen sujuvuus	LV02			9	9										
	LV03					3	9			9	3				
	LV17			3		3								9	
	LV52					9									
	LV80			9		3									
	LV19				9										
	LV20	9	9												
	LV22	9	9												
LV27	9			9											

27 30 39 48 45 45 0 18 30 18 15 45 72 9

Liite 2. Tarpeista ja vaatimuksista johdettu osien tärkeys

Liiketoimintatarve	Tarpeen painokerroin	Liiketoimintavaatimus	Tuoterakenne	Tuotehallinta	Sopimusrakenne	Sopimushallinta	Maksunlaskenta	veloitukset	Tarjouksen teko	Sopimuksen voimaansaatto	Muutos kesken vakuutuskauden	Uudistus	Sopimuksen päättäminen	Rajapinnat ja palvelut	Integraatiot	Käyttöliittymä
Automaation lisäys	0,073	LV01				0	1									
	0,073	LV07								0						1
	0,073	LV09									0	0				1
	0,073	LV10										0				1
	0,073	LV12				1							0			1
	0,073	LV15									0					1
	0,073	LV73												1		
Tärkeät toiminnot	0,673	LV11														6
	0,673	LV16									2	2				6
	0,673	LV05						6		6						
	0,673	LV08					2	6			6					
	0,673	LV14						6					6			
	0,673	LV18	6	2												
	0,673	LV21			6	6										
	0,673	LV23														6
	0,673	LV24			6		6									
	0,673	LV02			6	6										
0,673	LV28					2	6			6						
Rajapintojen tarjoaminen	0,064	LV04								0						1
	0,064	LV41									0					1
	0,064	LV44								0						1
	0,064	LV63					0									1
Tuotekehityksen sujuvuus	0,189	LV03				1	2					2	1			
	0,189	LV17			1	1										2
	0,189	LV52				2										
	0,189	LV80			2	1										
	0,189	LV19				2										
	0,189	LV20	2	2												
	0,189	LV22	2	2												
0,189	LV27		2	2												
		Yhteensä	9	7	20	16	14	26	0	7	15	4	7	3	17	6